API Reference

Amazon DynamoDB

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon DynamoDB: API Reference

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Welcome **1**
Actions **2**
Amazon DynamoDB 5
BatchExecuteStatement 8
BatchGetItem 14
BatchWriteItem 27
CreateBackup 41
CreateGlobalTable 46
CreateTable 52
DeleteBackup 70
DeleteItem 75
DeleteResourcePolicy 89
DeleteTable 95
DescribeBackup 104
DescribeContinuousBackups 108
DescribeContributorInsights 111
DescribeEndpoints 115
DescribeExport 117
DescribeGlobalTable 121
DescribeGlobalTableSettings 125
DescribeImport 130
DescribeKinesisStreamingDestination 134
DescribeLimits 137
DescribeTable 142
DescribeTableReplicaAutoScaling 151
DescribeTimeToLive 155
DisableKinesisStreamingDestination 158
EnableKinesisStreamingDestination 163
ExecuteStatement 168
ExecuteTransaction 176
ExportTableToPointInTime 186
GetItem 193
GetResourcePolicy 202
ImportTable 207

ListBackups 215
ListContributorInsights 220
ListExports 223
ListGlobalTables 227
ListImports 231
ListTables 235
ListTagsOfResource 239
PutItem 242
PutResourcePolicy 257
Query 263
RestoreTableFromBackup 283
RestoreTableToPointInTime 293
Scan 305
TagResource 325
TransactGetItems 329
TransactWriteItems 338
UntagResource 353
UpdateContinuousBackups 357
UpdateContributorInsights 360
UpdateGlobalTable 364
UpdateGlobalTableSettings 369
UpdateItem 378
UpdateKinesisStreamingDestination 396
UpdateTable 402
UpdateTableReplicaAutoScaling 419
UpdateTimeToLive 427
DynamoDB Accelerator 431
CreateCluster 432
CreateParameterGroup 441
CreateSubnetGroup 444
DecreaseReplicationFactor 448
DeleteCluster 453
DeleteParameterGroup 457
DeleteSubnetGroup 460
DescribeClusters 463
DescribeDefaultParameters 468

DescribeEvents ... 472
DescribeParameterGroups ... 476
DescribeParameters ... 479
DescribeSubnetGroups ... 483
IncreaseReplicationFactor ... 486
ListTags ... 491
RebootNode ... 494
TagResource ... 498
UntagResource ... 501
UpdateCluster ... 504
UpdateParameterGroup ... 510
UpdateSubnetGroup ... 513
Amazon DynamoDB Streams ... 516
DescribeStream ... 517
GetRecords ... 523
GetShardIterator ... 531
ListStreams ... 536
Data Types ... 541
Amazon DynamoDB ... 546
ArchivalSummary ... 550
AttributeDefinition ... 552
AttributeValue ... 554
AttributeValueUpdate ... 558
AutoScalingPolicyDescription ... 561
AutoScalingPolicyUpdate ... 563
AutoScalingSettingsDescription ... 565
AutoScalingSettingsUpdate ... 567
AutoScalingTargetTrackingScalingPolicyConfigurationDescription ... 569
AutoScalingTargetTrackingScalingPolicyConfigurationUpdate ... 571
BackupDescription ... 573
BackupDetails ... 575
BackupSummary ... 578
BatchStatementError ... 581
BatchStatementRequest ... 583
BatchStatementResponse ... 585
BillingModeSummary ... 587

CancellationReason 589
Capacity 591
Condition 593
ConditionCheck 598
ConsumedCapacity 600
ContinuousBackupsDescription 603
ContributorInsightsSummary 604
CreateGlobalSecondaryIndexAction 606
CreateGlobalTableWitnessGroupMemberAction 608
CreateReplicaAction 609
CreateReplicationGroupMemberAction 610
CsvOptions 612
Delete 614
DeleteGlobalSecondaryIndexAction 616
DeleteGlobalTableWitnessGroupMemberAction 617
DeleteReplicaAction 618
DeleteReplicationGroupMemberAction 619
DeleteRequest 620
EnableKinesisStreamingConfiguration 621
Endpoint 622
ExpectedAttributeValue 623
ExportDescription 629
ExportSummary 634
FailureException 636
Get 637
GlobalSecondaryIndex 639
GlobalSecondaryIndexAutoScalingUpdate 642
GlobalSecondaryIndexDescription 644
GlobalSecondaryIndexInfo 648
GlobalSecondaryIndexUpdate 650
GlobalSecondaryIndexWarmThroughputDescription 652
GlobalTable 654
GlobalTableDescription 656
GlobalTableGlobalSecondaryIndexSettingsUpdate 658
GlobalTableWitnessDescription 660
GlobalTableWitnessGroupUpdate 661

ImportSummary 663
ImportTableDescription 666
IncrementalExportSpecification 671
InputFormatOptions 673
ItemCollectionMetrics 674
ItemResponse 676
KeysAndAttributes 677
KeySchemaElement 680
KinesisDataStreamDestination 682
LocalSecondaryIndex 684
LocalSecondaryIndexDescription 686
LocalSecondaryIndexInfo 689
OnDemandThroughput 691
OnDemandThroughputOverride 693
ParameterizedStatement 694
PointInTimeRecoveryDescription 696
PointInTimeRecoverySpecification 698
Projection 700
ProvisionedThroughput 702
ProvisionedThroughputDescription 704
ProvisionedThroughputOverride 706
Put 707
PutRequest 709
Replica 710
ReplicaAutoScalingDescription 711
ReplicaAutoScalingUpdate 713
ReplicaDescription 715
ReplicaGlobalSecondaryIndex 719
ReplicaGlobalSecondaryIndexAutoScalingDescription 721
ReplicaGlobalSecondaryIndexAutoScalingUpdate 723
ReplicaGlobalSecondaryIndexDescription 725
ReplicaGlobalSecondaryIndexSettingsDescription 727
ReplicaGlobalSecondaryIndexSettingsUpdate 729
ReplicaSettingsDescription 731
ReplicaSettingsUpdate 734
ReplicationGroupUpdate 736

ReplicaUpdate 738
RestoreSummary 740
S3BucketSource 742
SourceTableDetails 744
SourceTableFeatureDetails 747
SSEDescription 749
SSESpecification 751
StreamSpecification 753
TableAutoScalingDescription 755
TableClassSummary 757
TableCreationParameters 758
TableDescription 761
TableWarmThroughputDescription 770
Tag 772
ThrottlingReason 774
TimeToLiveDescription 776
TimeToLiveSpecification 778
TransactGetItem 779
TransactWriteItem 780
Update 782
UpdateGlobalSecondaryIndexAction 785
UpdateKinesisStreamingConfiguration 787
UpdateReplicationGroupMemberAction 788
WarmThroughput 790
WriteRequest 791
DynamoDB Accelerator 792
Cluster 793
Endpoint 797
Event 799
Node 801
NodeTypeSpecificValue 803
NotificationConfiguration 804
Parameter 805
ParameterGroup 808
ParameterGroupStatus 809
ParameterNameValue 811

SecurityGroupMembership 812
SSEDescription 813
SSESpecification 814
Subnet 815
SubnetGroup 816
Tag 818
Amazon DynamoDB Streams 819
AttributeValue 820
Identity 824
KeySchemaElement 825
Record 827
SequenceNumberRange 830
Shard 832
ShardFilter 834
Stream 836
StreamDescription 838
StreamRecord 842
Common Errors 845

Welcome

Amazon DynamoDB provides low-level API actions for managing database tables and indexes, and for creating, reading, updating and deleting data. DynamoDB also provides API actions for accessing and processing stream records.

> **Note**
>
> This API Reference describes the low-level API for Amazon DynamoDB. Instead of making requests to the low-level API directly from your application, we recommend that you use one of the AWS Software Development Kits (SDKs) for your programming language. The AWS SDKs take care of request authentication, serialization, and connection management. For more information, see Overview of AWS SDK Support for DynamoDB in the Amazon DynamoDB Developer Guide.

At the end of each API action description there are links to the equivalent CLI command and programming-specific language method. Similarly, at the end of each API datatype description, there are links to the equivalent programming-specific language type.

Actions

The following actions are supported by Amazon DynamoDB:

- BatchExecuteStatement
- BatchGetItem
- BatchWriteItem
- CreateBackup
- CreateGlobalTable
- CreateTable
- DeleteBackup
- DeleteItem
- DeleteResourcePolicy
- DeleteTable
- DescribeBackup
- DescribeContinuousBackups
- DescribeContributorInsights
- DescribeEndpoints
- DescribeExport
- DescribeGlobalTable
- DescribeGlobalTableSettings
- DescribeImport
- DescribeKinesisStreamingDestination
- DescribeLimits
- DescribeTable
- DescribeTableReplicaAutoScaling
- DescribeTimeToLive
- DisableKinesisStreamingDestination
- EnableKinesisStreamingDestination
- ExecuteStatement

- ExecuteTransaction
- ExportTableToPointInTime
- GetItem
- GetResourcePolicy
- ImportTable
- ListBackups
- ListContributorInsights
- ListExports
- ListGlobalTables
- ListImports
- ListTables
- ListTagsOfResource
- PutItem
- PutResourcePolicy
- Query
- RestoreTableFromBackup
- RestoreTableToPointInTime
- Scan
- TagResource
- TransactGetItems
- TransactWriteItems
- UntagResource
- UpdateContinuousBackups
- UpdateContributorInsights
- UpdateGlobalTable
- UpdateGlobalTableSettings
- UpdateItem
- UpdateKinesisStreamingDestination
- UpdateTable
- UpdateTableReplicaAutoScaling

- UpdateTimeToLive

The following actions are supported by DynamoDB Accelerator:

- CreateCluster
- CreateParameterGroup
- CreateSubnetGroup
- DecreaseReplicationFactor
- DeleteCluster
- DeleteParameterGroup
- DeleteSubnetGroup
- DescribeClusters
- DescribeDefaultParameters
- DescribeEvents
- DescribeParameterGroups
- DescribeParameters
- DescribeSubnetGroups
- IncreaseReplicationFactor
- ListTags
- RebootNode
- TagResource
- UntagResource
- UpdateCluster
- UpdateParameterGroup
- UpdateSubnetGroup

The following actions are supported by Amazon DynamoDB Streams:

- DescribeStream
- GetRecords
- GetShardIterator
- ListStreams

Amazon DynamoDB

The following actions are supported by Amazon DynamoDB:

- BatchExecuteStatement
- BatchGetItem
- BatchWriteItem
- CreateBackup
- CreateGlobalTable
- CreateTable
- DeleteBackup
- DeleteItem
- DeleteResourcePolicy
- DeleteTable
- DescribeBackup
- DescribeContinuousBackups
- DescribeContributorInsights
- DescribeEndpoints
- DescribeExport
- DescribeGlobalTable
- DescribeGlobalTableSettings
- DescribeImport
- DescribeKinesisStreamingDestination
- DescribeLimits
- DescribeTable
- DescribeTableReplicaAutoScaling
- DescribeTimeToLive
- DisableKinesisStreamingDestination
- EnableKinesisStreamingDestination
- ExecuteStatement
- ExecuteTransaction

- ExportTableToPointInTime
- GetItem
- GetResourcePolicy
- ImportTable
- ListBackups
- ListContributorInsights
- ListExports
- ListGlobalTables
- ListImports
- ListTables
- ListTagsOfResource
- PutItem
- PutResourcePolicy
- Query
- RestoreTableFromBackup
- RestoreTableToPointInTime
- Scan
- TagResource
- TransactGetItems
- TransactWriteItems
- UntagResource
- UpdateContinuousBackups
- UpdateContributorInsights
- UpdateGlobalTable
- UpdateGlobalTableSettings
- UpdateItem
- UpdateKinesisStreamingDestination
- UpdateTable
- UpdateTableReplicaAutoScaling
- UpdateTimeToLive

BatchExecuteStatement

Service: Amazon DynamoDB

This operation allows you to perform batch reads or writes on data stored in DynamoDB, using PartiQL. Each read statement in a `BatchExecuteStatement` must specify an equality condition on all key attributes. This enforces that each `SELECT` statement in a batch returns at most a single item. For more information, see Running batch operations with PartiQL for DynamoDB .

> **Note**
>
> The entire batch must consist of either read statements or write statements, you cannot mix both in one batch.

> **Important**
>
> A HTTP 200 response does not mean that all statements in the BatchExecuteStatement succeeded. Error details for individual statements can be found under the Error field of the `BatchStatementResponse` for each statement.

Request Syntax

```
{
   "ReturnConsumedCapacity": "string",
   "Statements": [
      {
         "ConsistentRead": boolean,
         "Parameters": [
            {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
                  "AttributeValue"
               ],
               "M": {
                  "string" : "AttributeValue"
               },
               "N": "string",
```

```
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
            }
        ],
        "ReturnValuesOnConditionCheckFailure": "string",
        "Statement": "string"
    }
  ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

Statements

The list of PartiQL statements representing the batch to run.

Type: Array of BatchStatementRequest objects

Array Members: Minimum number of 1 item. Maximum number of 25 items.

Required: Yes

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.

- NONE - No ConsumedCapacity details are included in the response.

Type: String

Valid Values: INDEXES | TOTAL | NONE

Required: No

Response Syntax

```
{
   "ConsumedCapacity": [
      {
         "CapacityUnits": number,
         "GlobalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "LocalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "ReadCapacityUnits": number,
         "Table": {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         },
         "TableName": "string",
         "WriteCapacityUnits": number
      }
   ],
   "Responses": [
      {
         "Error": {
            "Code": "string",
            "Item": {
```

```
                "string" : {
                   "B": blob,
                   "BOOL": boolean,
                   "BS": [ blob ],
                   "L": [
                      "AttributeValue"
                   ],
                   "M": {
                      "string" : "AttributeValue"
                   },
                   "N": "string",
                   "NS": [ "string" ],
                   "NULL": boolean,
                   "S": "string",
                   "SS": [ "string" ]
                }
             },
             "Message": "string"
          },
          "Item": {
             "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "TableName": "string"
       }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The capacity units consumed by the entire operation. The values of the list are ordered according to the ordering of the statements.

Type: Array of ConsumedCapacity objects

Responses

The response to each PartiQL statement in the batch. The values of the list are ordered according to the ordering of the request statements.

Type: Array of BatchStatementResponse objects

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

BatchGetItem

Service: Amazon DynamoDB

The `BatchGetItem` operation returns the attributes of one or more items from one or more tables. You identify requested items by primary key.

A single operation can retrieve up to 16 MB of data, which can contain as many as 100 items. `BatchGetItem` returns a partial result if the response size limit is exceeded, the table's provisioned throughput is exceeded, more than 1MB per partition is requested, or an internal processing failure occurs. If a partial result is returned, the operation returns a value for `UnprocessedKeys`. You can use this value to retry the operation starting with the next item to get.

> **Important**
>
> If you request more than 100 items, `BatchGetItem` returns a `ValidationException` with the message "Too many items requested for the BatchGetItem call."

For example, if you ask to retrieve 100 items, but each individual item is 300 KB in size, the system returns 52 items (so as not to exceed the 16 MB limit). It also returns an appropriate `UnprocessedKeys` value so you can get the next page of results. If desired, your application can include its own logic to assemble the pages of results into one dataset.

If *none* of the items can be processed due to insufficient provisioned throughput on all of the tables in the request, then `BatchGetItem` returns a `ProvisionedThroughputExceededException`. If *at least one* of the items is successfully processed, then `BatchGetItem` completes successfully, while returning the keys of the unread items in `UnprocessedKeys`.

> **Important**
>
> If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, *we strongly recommend that you use an exponential backoff algorithm*. If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed. For more information, see Batch Operations and Error Handling in the *Amazon DynamoDB Developer Guide*.

By default, `BatchGetItem` performs eventually consistent reads on every table in the request. If you want strongly consistent reads instead, you can set `ConsistentRead` to `true` for any or all tables.

In order to minimize response latency, `BatchGetItem` may retrieve items in parallel.

When designing your application, keep in mind that DynamoDB does not return items in any particular order. To help parse the response by item, include the primary key values for the items in your request in the `ProjectionExpression` parameter.

If a requested item does not exist, it is not returned in the result. Requests for nonexistent items consume the minimum read capacity units according to the type of read. For more information, see Working with Tables in the *Amazon DynamoDB Developer Guide*.

> **Note**
>
> `BatchGetItem` will result in a `ValidationException` if the same key is specified multiple times.

Request Syntax

```
{
   "RequestItems": {
      "string" : {
         "AttributesToGet": [ "string" ],
         "ConsistentRead": boolean,
         "ExpressionAttributeNames": {
            "string" : "string"
         },
         "Keys": [
            {
               "string" : {
                  "B": blob,
                  "BOOL": boolean,
                  "BS": [ blob ],
                  "L": [
                     "AttributeValue"
                  ],
                  "M": {
                     "string" : "AttributeValue"
                  },
```

```
                    "N": "string",
                    "NS": [ "string" ],
                    "NULL": boolean,
                    "S": "string",
                    "SS": [ "string" ]
                }
            }
        ],
        "ProjectionExpression": "string"
    }
  },
  "ReturnConsumedCapacity": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

RequestItems

A map of one or more table names or table ARNs and, for each table, a map that describes one or more items to retrieve from that table. Each table name or ARN can be used only once per `BatchGetItem` request.

Each element in the map of items to retrieve consists of the following:

- `ConsistentRead` - If `true`, a strongly consistent read is used; if `false` (the default), an eventually consistent read is used.
- `ExpressionAttributeNames` - One or more substitution tokens for attribute names in the `ProjectionExpression` parameter. The following are some use cases for using `ExpressionAttributeNames`:
  - To access an attribute whose name conflicts with a DynamoDB reserved word.
  - To create a placeholder for repeating occurrences of an attribute name in an expression.
  - To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the **#** character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for `ExpressionAttributeNames`:

- `{"#P":"Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

> **Note**
>
> Tokens that begin with the **:** character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information about expression attribute names, see Accessing Item Attributes in the *Amazon DynamoDB Developer Guide*.

- `Keys` - An array of primary key attribute values that define specific items in the table. For each primary key, you must provide *all* of the key attributes. For example, with a simple primary key, you only need to provide the partition key value. For a composite key, you must provide *both* the partition key value and the sort key value.
- `ProjectionExpression` - A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

  If no attribute names are specified, then all attributes are returned. If any of the requested attributes are not found, they do not appear in the result.

  For more information, see Accessing Item Attributes in the *Amazon DynamoDB Developer Guide*.
- `AttributesToGet` - This is a legacy parameter. Use `ProjectionExpression` instead. For more information, see AttributesToGet in the *Amazon DynamoDB Developer Guide*.

Type: String to KeysAndAttributes object map

Map Entries: Maximum number of 100 items.

Key Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

Response Syntax

```
{
   "ConsumedCapacity": [
      {
         "CapacityUnits": number,
         "GlobalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "LocalSecondaryIndexes": {
            "string" : {
```

```
                "CapacityUnits": number,
                "ReadCapacityUnits": number,
                "WriteCapacityUnits": number
            }
        },
        "ReadCapacityUnits": number,
        "Table": {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
        },
        "TableName": "string",
        "WriteCapacityUnits": number
    }
],
"Responses": {
    "string" : [
        {
            "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
                "L": [
                    "AttributeValue"
                ],
                "M": {
                    "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
            }
        }
    ]
},
"UnprocessedKeys": {
    "string" : {
        "AttributesToGet": [ "string" ],
        "ConsistentRead": boolean,
        "ExpressionAttributeNames": {
            "string" : "string"
        },
```

```
        "Keys": [
           {
              "string" : {
                 "B": blob,
                 "BOOL": boolean,
                 "BS": [ blob ],
                 "L": [
                    "AttributeValue"
                 ],
                 "M": {
                    "string" : "AttributeValue"
                 },
                 "N": "string",
                 "NS": [ "string" ],
                 "NULL": boolean,
                 "S": "string",
                 "SS": [ "string" ]
              }
           }
        ],
        "ProjectionExpression": "string"
     }
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The read capacity units consumed by the entire `BatchGetItem` operation.

Each element consists of:

- `TableName` - The table that consumed the provisioned throughput.
- `CapacityUnits` - The total number of capacity units consumed.

Type: Array of ConsumedCapacity objects

Responses

A map of table name or table ARN to a list of items. Each object in `Responses` consists of a table name or ARN, along with a map of attribute data consisting of the data type and attribute value.

Type: String to array of string to AttributeValue object maps map

Key Length Constraints: Minimum length of 1. Maximum length of 1024.

Key Length Constraints: Maximum length of 65535.

UnprocessedKeys

A map of tables and their respective keys that were not processed with the current response. The `UnprocessedKeys` value is in the same form as `RequestItems`, so the value can be provided directly to a subsequent `BatchGetItem` operation. For more information, see `RequestItems` in the Request Parameters section.

Each element consists of:

- `Keys` - An array of primary key attribute values that define specific items in the table.
- `ProjectionExpression` - One or more attributes to be retrieved from the table or index. By default, all attributes are returned. If a requested attribute is not found, it does not appear in the result.
- `ConsistentRead` - The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

If there are no unprocessed keys remaining, the response contains an empty `UnprocessedKeys` map.

Type: String to KeysAndAttributes object map

Map Entries: Maximum number of 100 items.

Key Length Constraints: Minimum length of 1. Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

Examples

Retrieve Items from Multiple Tables

The following example requests attributes from two different tables.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.BatchGetItem

{
    "RequestItems": {
        "Forum": {
            "Keys": [
                {
                    "Name":{"S":"Amazon DynamoDB"}
                },
                {
                    "Name":{"S":"Amazon RDS"}
                },
                {
                    "Name":{"S":"Amazon Redshift"}
                }
            ],
            "ProjectionExpression":"Name, Threads, Messages, Views"
        },
        "Thread": {
            "Keys": [
                {
                    "ForumName":{"S":"Amazon DynamoDB"},
                    "Subject":{"S":"Concurrent reads"}
                }
            ],
            "ProjectionExpression":"Tags, Message"
        }
    },
    "ReturnConsumedCapacity": "TOTAL"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "Responses": {
        "Forum": [
            {
                "Name":{
                    "S":"Amazon DynamoDB"
                },
                "Threads":{
                    "N":"5"
                },
                "Messages":{
                    "N":"19"
                },
                "Views":{
                    "N":"35"
                }
            },
            {
                "Name":{
                    "S":"Amazon RDS"
                },
                "Threads":{
                    "N":"8"
                },
                "Messages":{
                    "N":"32"
                },
                "Views":{
                    "N":"38"
                }
            },
            {
                "Name":{
                    "S":"Amazon Redshift"
                },
                "Threads":{
```

```
                    "N":"12"
                },
                "Messages":{
                    "N":"55"
                },
                "Views":{
                    "N":"47"
                }
            }
        ]
        "Thread": [
            {
                "Tags":{
                    "SS":["Reads","MultipleUsers"]
                },
                "Message":{
                    "S":"How many users can read a single data item at a time? Are
 there any limits?"
                }
            }
        ]
    },
    "UnprocessedKeys": {
    },
    "ConsumedCapacity": [
        {
            "TableName": "Forum",
            "CapacityUnits": 3
        },
        {
            "TableName": "Thread",
            "CapacityUnits": 1
        }
    ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

BatchWriteItem

Service: Amazon DynamoDB

The `BatchWriteItem` operation puts or deletes multiple items in one or more tables. A single call to `BatchWriteItem` can transmit up to 16MB of data over the network, consisting of up to 25 item put or delete operations. While individual items can be up to 400 KB once stored, it's important to note that an item's representation might be greater than 400KB while being sent in DynamoDB's JSON format for the API call. For more details on this distinction, see Naming Rules and Data Types.

> **Note**
>
> `BatchWriteItem` cannot update items. If you perform a `BatchWriteItem` operation on an existing item, that item's values will be overwritten by the operation and it will appear like it was updated. To update items, we recommend you use the `UpdateItem` action.

The individual `PutItem` and `DeleteItem` operations specified in `BatchWriteItem` are atomic; however `BatchWriteItem` as a whole is not. If any requested operations fail because the table's provisioned throughput is exceeded or an internal processing failure occurs, the failed operations are returned in the `UnprocessedItems` response parameter. You can investigate and optionally resend the requests. Typically, you would call `BatchWriteItem` in a loop. Each iteration would check for unprocessed items and submit a new `BatchWriteItem` request with those unprocessed items until all items have been processed.

For tables and indexes with provisioned capacity, if none of the items can be processed due to insufficient provisioned throughput on all of the tables in the request, then `BatchWriteItem` returns a `ProvisionedThroughputExceededException`. For all tables and indexes, if none of the items can be processed due to other throttling scenarios (such as exceeding partition level limits), then `BatchWriteItem` returns a `ThrottlingException`.

> **Important**
>
> If DynamoDB returns any unprocessed items, you should retry the batch operation on those items. However, *we strongly recommend that you use an exponential backoff algorithm*. If you retry the batch operation immediately, the underlying read or write requests can still fail due to throttling on the individual tables. If you delay the batch operation using exponential backoff, the individual requests in the batch are much more likely to succeed.

For more information, see Batch Operations and Error Handling in the *Amazon DynamoDB Developer Guide*.

With `BatchWriteItem`, you can efficiently write or delete large amounts of data, such as from Amazon EMR, or copy data from another database into DynamoDB. In order to improve performance with these large-scale operations, `BatchWriteItem` does not behave in the same way as individual `PutItem` and `DeleteItem` calls would. For example, you cannot specify conditions on individual put and delete requests, and `BatchWriteItem` does not return deleted items in the response.

If you use a programming language that supports concurrency, you can use threads to write items in parallel. Your application must include the necessary logic to manage the threads. With languages that don't support threading, you must update or delete the specified items one at a time. In both situations, `BatchWriteItem` performs the specified put and delete operations in parallel, giving you the power of the thread pool approach without having to introduce complexity into your application.

Parallel processing reduces latency, but each specified put and delete request consumes the same number of write capacity units whether it is processed in parallel or not. Delete operations on nonexistent items consume one write capacity unit.

If one or more of the following is true, DynamoDB rejects the entire batch write operation:

- One or more tables specified in the `BatchWriteItem` request does not exist.
- Primary key attributes specified on an item in the request do not match those in the corresponding table's primary key schema.
- You try to perform multiple operations on the same item in the same `BatchWriteItem` request. For example, you cannot put and delete the same item in the same `BatchWriteItem` request.
- Your request contains at least two items with identical hash and range keys (which essentially is two put operations).
- There are more than 25 requests in the batch.
- Any individual item in a batch exceeds 400 KB.
- The total request size exceeds 16 MB.
- Any individual items with keys exceeding the key length limits. For a partition key, the limit is 2048 bytes and for a sort key, the limit is 1024 bytes.

Request Syntax

```
{
   "RequestItems": {
      "string" : [
         {
            "DeleteRequest": {
               "Key": {
                  "string" : {
                     "B": blob,
                     "BOOL": boolean,
                     "BS": [ blob ],
                     "L": [
                        "AttributeValue"
                     ],
                     "M": {
                        "string" : "AttributeValue"
                     },
                     "N": "string",
                     "NS": [ "string" ],
                     "NULL": boolean,
                     "S": "string",
                     "SS": [ "string" ]
                  }
               }
            },
            "PutRequest": {
               "Item": {
                  "string" : {
                     "B": blob,
                     "BOOL": boolean,
                     "BS": [ blob ],
                     "L": [
                        "AttributeValue"
                     ],
                     "M": {
                        "string" : "AttributeValue"
                     },
                     "N": "string",
                     "NS": [ "string" ],
                     "NULL": boolean,
                     "S": "string",
                     "SS": [ "string" ]
```

```
                    }
                }
            }
        }
    ]
  },
  "ReturnConsumedCapacity": "string",
  "ReturnItemCollectionMetrics": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

RequestItems

A map of one or more table names or table ARNs and, for each table, a list of operations to be performed (`DeleteRequest` or `PutRequest`). Each element in the map consists of the following:

- `DeleteRequest` - Perform a `DeleteItem` operation on the specified item. The item to be deleted is identified by a `Key` subelement:
  - `Key` - A map of primary key attribute values that uniquely identify the item. Each entry in this map consists of an attribute name and an attribute value. For each primary key, you must provide *all* of the key attributes. For example, with a simple primary key, you only need to provide a value for the partition key. For a composite primary key, you must provide values for *both* the partition key and the sort key.
- `PutRequest` - Perform a `PutItem` operation on the specified item. The item to be put is identified by an `Item` subelement:
  - `Item` - A map of attributes and their values. Each entry in this map consists of an attribute name and an attribute value. Attribute values must not be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests that contain empty values are rejected with a `ValidationException` exception.

If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

Type: String to array of WriteRequest objects map

Map Entries: Maximum number of 25 items.

Key Length Constraints: Minimum length of 1. Maximum length of 1024.

Array Members: Minimum number of 1 item. Maximum number of 25 items.

Required: Yes

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

ReturnItemCollectionMetrics

Determines whether item collection metrics are returned. If set to `SIZE`, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to `NONE` (the default), no statistics are returned.

Type: String

Valid Values: `SIZE | NONE`

Required: No

Response Syntax

```
{
   "ConsumedCapacity": [
      {
         "CapacityUnits": number,
         "GlobalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "LocalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "ReadCapacityUnits": number,
         "Table": {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         },
         "TableName": "string",
         "WriteCapacityUnits": number
      }
   ],
   "ItemCollectionMetrics": {
      "string" : [
         {
            "ItemCollectionKey": {
               "string" : {
                  "B": blob,
                  "BOOL": boolean,
                  "BS": [ blob ],
                  "L": [
                     "AttributeValue"
                  ],
                  "M": {
                     "string" : "AttributeValue"
```

```
                    },
                    "N": "string",
                    "NS": [ "string" ],
                    "NULL": boolean,
                    "S": "string",
                    "SS": [ "string" ]
                }
            },
            "SizeEstimateRangeGB": [ number ]
        }
    ]
},
"UnprocessedItems": {
    "string" : [
        {
            "DeleteRequest": {
                "Key": {
                    "string" : {
                        "B": blob,
                        "BOOL": boolean,
                        "BS": [ blob ],
                        "L": [
                            "AttributeValue"
                        ],
                        "M": {
                            "string" : "AttributeValue"
                        },
                        "N": "string",
                        "NS": [ "string" ],
                        "NULL": boolean,
                        "S": "string",
                        "SS": [ "string" ]
                    }
                }
            },
            "PutRequest": {
                "Item": {
                    "string" : {
                        "B": blob,
                        "BOOL": boolean,
                        "BS": [ blob ],
                        "L": [
                            "AttributeValue"
                        ],
```

```
                        "M": {
                           "string" : "AttributeValue"
                        },
                        "N": "string",
                        "NS": [ "string" ],
                        "NULL": boolean,
                        "S": "string",
                        "SS": [ "string" ]
                     }
                  }
               }
            }
         ]
      }
   }
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The capacity units consumed by the entire `BatchWriteItem` operation.

Each element consists of:

- `TableName` - The table that consumed the provisioned throughput.
- `CapacityUnits` - The total number of capacity units consumed.

Type: Array of ConsumedCapacity objects

ItemCollectionMetrics

A list of tables that were processed by `BatchWriteItem` and, for each table, information about any item collections that were affected by individual `DeleteItem` or `PutItem` operations.

Each entry consists of the following subelements:

- `ItemCollectionKey` - The partition key value of the item collection. This is the same as the partition key value of the item.
- `SizeEstimateRangeGB` - An estimate of item collection size, expressed in GB. This is a two-element array containing a lower bound and an upper bound for the estimate. The estimate

includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on the table. Use this estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

Type: String to array of ItemCollectionMetrics objects map

Key Length Constraints: Minimum length of 1. Maximum length of 1024.

UnprocessedItems

A map of tables and requests against those tables that were not processed. The `UnprocessedItems` value is in the same form as `RequestItems`, so you can provide this value directly to a subsequent `BatchWriteItem` operation. For more information, see `RequestItems` in the Request Parameters section.

Each `UnprocessedItems` entry consists of a table name or table ARN and, for that table, a list of operations to perform (`DeleteRequest` or `PutRequest`).

- `DeleteRequest` - Perform a `DeleteItem` operation on the specified item. The item to be deleted is identified by a `Key` subelement:
  - `Key` - A map of primary key attribute values that uniquely identify the item. Each entry in this map consists of an attribute name and an attribute value.
- `PutRequest` - Perform a `PutItem` operation on the specified item. The item to be put is identified by an `Item` subelement:
  - `Item` - A map of attributes and their values. Each entry in this map consists of an attribute name and an attribute value. Attribute values must not be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests that contain empty values will be rejected with a `ValidationException` exception.

    If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

If there are no unprocessed items remaining, the response contains an empty `UnprocessedItems` map.

Type: String to array of WriteRequest objects map

Map Entries: Maximum number of 25 items.

Key Length Constraints: Minimum length of 1. Maximum length of 1024.

Array Members: Minimum number of 1 item. Maximum number of 25 items.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ItemCollectionSizeLimitExceededException

An item collection is too large. This exception is only returned for tables that have one or more local secondary indexes.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

ReplicatedWriteConflictException

The request was rejected because one or more items in the request are being modified by a request in another Region.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

Examples

Multiple Operations on One Table

This example writes several items to the `Forum` table. The response shows that the final put operation failed, possibly because the application exceeded the provisioned throughput on the table. The `UnprocessedItems` object shows the unsuccessful put request. The application can call `BatchWriteItem` again to address such unprocessed requests.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
```

```
X-Amz-Target: DynamoDB_20120810.BatchWriteItem

{
    "RequestItems": {
        "Forum": [
            {
                "PutRequest": {
                    "Item": {
                        "Name": {
                            "S": "Amazon DynamoDB"
                        },
                        "Category": {
                            "S": "Amazon Web Services"
                        }
                    }
                }
            },
            {
                "PutRequest": {
                    "Item": {
                        "Name": {
                            "S": "Amazon RDS"
                        },
                        "Category": {
                            "S": "Amazon Web Services"
                        }
                    }
                }
            },
            {
                "PutRequest": {
                    "Item": {
                        "Name": {
                            "S": "Amazon Redshift"
                        },
                        "Category": {
                            "S": "Amazon Web Services"
                        }
                    }
                }
            },
            {
                "PutRequest": {
                    "Item": {
```

```
                        "Name": {
                            "S": "Amazon ElastiCache"
                        },
                        "Category": {
                            "S": "Amazon Web Services"
                        }
                    }
                }
            }
        ]
    },
    "ReturnConsumedCapacity": "TOTAL"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "UnprocessedItems": {
        "Forum": [
            {
                "PutRequest": {
                    "Item": {
                        "Name": {
                            "S": "Amazon ElastiCache"
                        },
                        "Category": {
                            "S": "Amazon Web Services"
                        }
                    }
                }
            }
        ]
    },
    "ConsumedCapacity": [
        {
            "TableName": "Forum",
            "CapacityUnits": 3
```

```
        }
    ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateBackup

Service: Amazon DynamoDB

Creates a backup for an existing table.

Each time you create an on-demand backup, the entire table data is backed up. There is no limit to the number of on-demand backups that can be taken.

When you create an on-demand backup, a time marker of the request is cataloged, and the backup is created asynchronously, by applying all changes until the time of the request to the last full table snapshot. Backup requests are processed instantaneously and become available for restore within minutes.

You can call `CreateBackup` at a maximum rate of 50 times per second.

All backups in DynamoDB work without consuming any provisioned throughput on the table.

If you submit a backup request on 2018-12-14 at 14:25:00, the backup is guaranteed to contain all data committed to the table up to 14:24:00, and data committed after 14:26:00 will not be. The backup might contain data modifications made between 14:24:00 and 14:26:00. On-demand backup does not support causal consistency.

Along with data, the following are also included on the backups:

- Global secondary indexes (GSIs)
- Local secondary indexes (LSIs)
- Streams
- Provisioned read and write capacity

Request Syntax

```
{
   "BackupName": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

BackupName

Specified name for the backup.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

TableName

The name of the table. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "BackupDetails": {
      "BackupArn": "string",
      "BackupCreationDateTime": number,
      "BackupExpiryDateTime": number,
      "BackupName": "string",
      "BackupSizeBytes": number,
      "BackupStatus": "string",
      "BackupType": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BackupDetails

Contains the details of the backup created for the table.

Type: BackupDetails object

Errors

For information about the errors that are common to all actions, see Common Errors.

BackupInUseException

There is another ongoing conflicting backup control plane operation on the table. The backup is either being created, deleted or restored to a table.

HTTP Status Code: 400

ContinuousBackupsUnavailableException

Backups have not yet been enabled for this table.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

TableInUseException

A target table with the specified name is either being created or deleted.

HTTP Status Code: 400

TableNotFoundException

A source table with the name `TableName` does not currently exist within the subscriber's account or the subscriber is operating in the wrong AWS Region.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3

- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateGlobalTable

Service: Amazon DynamoDB

Creates a global table from an existing table. A global table creates a replication relationship between two or more DynamoDB tables with the same table name in the provided Regions.

> **Important**
>
> This documentation is for version 2017.11.29 (Legacy) of global tables, which should be avoided for new global tables. Customers should use Global Tables version 2019.11.21 (Current) when possible, because it provides greater flexibility, higher efficiency, and consumes less write capacity than 2017.11.29 (Legacy).
> To determine which version you're using, see Determining the global table version you are using. To update existing global tables from version 2017.11.29 (Legacy) to version 2019.11.21 (Current), see Upgrading global tables.

If you want to add a new replica table to a global table, each of the following conditions must be true:

- The table must have the same primary key as all of the other replicas.
- The table must have the same name as all of the other replicas.
- The table must have DynamoDB Streams enabled, with the stream containing both the new and the old images of the item.
- None of the replica tables in the global table can contain any data.

If global secondary indexes are specified, then the following conditions must also be met:

- The global secondary indexes must have the same name.
- The global secondary indexes must have the same hash key and sort key (if present).

If local secondary indexes are specified, then the following conditions must also be met:

- The local secondary indexes must have the same name.
- The local secondary indexes must have the same hash key and sort key (if present).

> **Important**
>
> Write capacity settings should be set consistently across your replica tables and secondary indexes. DynamoDB strongly recommends enabling auto scaling to manage the write capacity settings for all of your global tables replicas and indexes.
> If you prefer to manage write capacity settings manually, you should provision equal replicated write capacity units to your replica tables. You should also provision equal replicated write capacity units to matching secondary indexes across your global table.

Request Syntax

```
{
   "GlobalTableName": "string",
   "ReplicationGroup": [
      {
         "RegionName": "string"
      }
   ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

GlobalTableName

The global table name.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

ReplicationGroup

The Regions where the global table needs to be created.

Type: Array of Replica objects

Required: Yes

Response Syntax

```
{
   "GlobalTableDescription": {
      "CreationDateTime": number,
      "GlobalTableArn": "string",
      "GlobalTableName": "string",
      "GlobalTableStatus": "string",
      "ReplicationGroup": [
         {
            "GlobalSecondaryIndexes": [
               {
                  "IndexName": "string",
                  "OnDemandThroughputOverride": {
                     "MaxReadRequestUnits": number
                  },
                  "ProvisionedThroughputOverride": {
                     "ReadCapacityUnits": number
                  },
                  "WarmThroughput": {
                     "ReadUnitsPerSecond": number,
                     "Status": "string",
                     "WriteUnitsPerSecond": number
                  }
               }
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
               "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
               "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "ReplicaInaccessibleDateTime": number,
```

```
            "ReplicaStatus": "string",
            "ReplicaStatusDescription": "string",
            "ReplicaStatusPercentProgress": "string",
            "ReplicaTableClassSummary": {
               "LastUpdateDateTime": number,
               "TableClass": "string"
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ]
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GlobalTableDescription

Contains the details of the global table.

Type: GlobalTableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

GlobalTableAlreadyExistsException

The specified global table already exists.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

TableNotFoundException

A source table with the name `TableName` does not currently exist within the subscriber's account or the subscriber is operating in the wrong AWS Region.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2

- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateTable

Service: Amazon DynamoDB

The `CreateTable` operation adds a new table to your account. In an AWS account, table names must be unique within each Region. That is, you can have two tables with same name if you create the tables in different Regions.

`CreateTable` is an asynchronous operation. Upon receiving a `CreateTable` request, DynamoDB immediately returns a response with a `TableStatus` of `CREATING`. After the table is created, DynamoDB sets the `TableStatus` to `ACTIVE`. You can perform read and write operations only on an `ACTIVE` table.

You can optionally define secondary indexes on the new table, as part of the `CreateTable` operation. If you want to create multiple tables with secondary indexes on them, you must create the tables sequentially. Only one table with secondary indexes can be in the `CREATING` state at any given time.

You can use the `DescribeTable` action to check the table status.

Request Syntax

```
{
   "AttributeDefinitions": [
      {
         "AttributeName": "string",
         "AttributeType": "string"
      }
   ],
   "BillingMode": "string",
   "DeletionProtectionEnabled": boolean,
   "GlobalSecondaryIndexes": [
      {
         "IndexName": "string",
         "KeySchema": [
            {
               "AttributeName": "string",
               "KeyType": "string"
            }
         ],
         "OnDemandThroughput": {
            "MaxReadRequestUnits": number,
            "MaxWriteRequestUnits": number
```

```
            },
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            },
            "ProvisionedThroughput": {
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "KeySchema": [
         {
            "AttributeName": "string",
            "KeyType": "string"
         }
      ],
      "LocalSecondaryIndexes": [
         {
            "IndexName": "string",
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            }
         }
      ],
      "OnDemandThroughput": {
         "MaxReadRequestUnits": number,
         "MaxWriteRequestUnits": number
      },
      "ProvisionedThroughput": {
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
```

```
    "ResourcePolicy": "string",
    "SSESpecification": {
       "Enabled": boolean,
       "KMSMasterKeyId": "string",
       "SSEType": "string"
    },
    "StreamSpecification": {
       "StreamEnabled": boolean,
       "StreamViewType": "string"
    },
    "TableClass": "string",
    "TableName": "string",
    "Tags": [
       {
          "Key": "string",
          "Value": "string"
       }
    ],
    "WarmThroughput": {
       "ReadUnitsPerSecond": number,
       "WriteUnitsPerSecond": number
    }
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

AttributeDefinitions

An array of attributes that describe the key schema for the table and indexes.

Type: Array of AttributeDefinition objects

Required: Yes

KeySchema

Specifies the attributes that make up the primary key for a table or an index. The attributes in `KeySchema` must also be defined in the `AttributeDefinitions` array. For more information, see Data Model in the *Amazon DynamoDB Developer Guide*.

Each `KeySchemaElement` in the array is composed of:

- `AttributeName` - The name of this key attribute.
- `KeyType` - The role that the key attribute will assume:
  - `HASH` - partition key
  - `RANGE` - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from the DynamoDB usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.
> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

For a simple primary key (partition key), you must provide exactly one element with a `KeyType` of `HASH`.

For a composite primary key (partition key and sort key), you must provide exactly two elements, in this order: The first element must have a `KeyType` of `HASH`, and the second element must have a `KeyType` of `RANGE`.

For more information, see Working with Tables in the *Amazon DynamoDB Developer Guide*.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: Yes

TableName

The name of the table to create. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

BillingMode

Controls how you are charged for read and write throughput and how you manage capacity. This setting can be changed later.

- `PAY_PER_REQUEST` - We recommend using `PAY_PER_REQUEST` for most DynamoDB workloads. `PAY_PER_REQUEST` sets the billing mode to On-demand capacity mode.
- `PROVISIONED` - We recommend using `PROVISIONED` for steady workloads with predictable growth where capacity requirements can be reliably forecasted. `PROVISIONED` sets the billing mode to Provisioned capacity mode.

Type: String

Valid Values: `PROVISIONED | PAY_PER_REQUEST`

Required: No

DeletionProtectionEnabled

Indicates whether deletion protection is to be enabled (true) or disabled (false) on the table.

Type: Boolean

Required: No

GlobalSecondaryIndexes

One or more global secondary indexes (the maximum is 20) to be created on the table. Each global secondary index in the array includes the following:

- `IndexName` - The name of the global secondary index. Must be unique only for this table.
- `KeySchema` - Specifies the key schema for the global secondary index.
- `Projection` - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
  - `ProjectionType` - One of the following:

  - `KEYS_ONLY` - Only the index and primary keys are projected into the index.
  - `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes is in `NonKeyAttributes`.
  - `ALL` - All of the table attributes are projected into the index.
- `NonKeyAttributes` - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in `NonKeyAttributes`, summed across all of the secondary indexes, must not exceed 100. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total. This limit only applies when you specify the ProjectionType of `INCLUDE`. You still can specify the ProjectionType of `ALL` to project all attributes from the source table, even if the table has more than 100 attributes.
- `ProvisionedThroughput` - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units.

Type: Array of GlobalSecondaryIndex objects

Required: No

LocalSecondaryIndexes

One or more local secondary indexes (the maximum is 5) to be created on the table. Each index is scoped to a given partition key value. There is a 10 GB size limit per partition key value; otherwise, the size of a local secondary index is unconstrained.

Each local secondary index in the array includes the following:

- `IndexName` - The name of the local secondary index. Must be unique only for this table.
- `KeySchema` - Specifies the key schema for the local secondary index. The key schema must begin with the same partition key as the table.
- `Projection` - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
  - `ProjectionType` - One of the following:
    - `KEYS_ONLY` - Only the index and primary keys are projected into the index.
    - `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes is in `NonKeyAttributes`.
    - `ALL` - All of the table attributes are projected into the index.

- `NonKeyAttributes` - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in `NonKeyAttributes`, summed across all of the secondary indexes, must not exceed 100. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total. This limit only applies when you specify the ProjectionType of `INCLUDE`. You still can specify the ProjectionType of `ALL` to project all attributes from the source table, even if the table has more than 100 attributes.

Type: Array of LocalSecondaryIndex objects

Required: No

OnDemandThroughput

Sets the maximum number of read and write units for the specified table in on-demand capacity mode. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughput

Represents the provisioned throughput settings for a specified table or index. The settings can be modified using the `UpdateTable` operation.

If you set BillingMode as `PROVISIONED`, you must specify this property. If you set BillingMode as `PAY_PER_REQUEST`, you cannot specify this property.

For current minimum and maximum provisioned throughput values, see Service, Account, and Table Quotas in the *Amazon DynamoDB Developer Guide*.

Type: ProvisionedThroughput object

Required: No

ResourcePolicy

An AWS resource-based policy document in JSON format that will be attached to the table.

When you attach a resource-based policy while creating a table, the policy application is *strongly consistent*.

The maximum size supported for a resource-based policy document is 20 KB. DynamoDB counts whitespaces when calculating the size of a policy against this limit. For a full list of all considerations that apply for resource-based policies, see Resource-based policy considerations.

> **Note**
>
> You need to specify the `CreateTable` and `PutResourcePolicy` IAM actions for authorizing a user to create a table with a resource-based policy.

Type: String

Required: No

SSESpecification

Represents the settings used to enable server-side encryption.

Type: SSESpecification object

Required: No

StreamSpecification

The settings for DynamoDB Streams on the table. These settings consist of:

- `StreamEnabled` - Indicates whether DynamoDB Streams is to be enabled (true) or disabled (false).
- `StreamViewType` - When an item in the table is modified, `StreamViewType` determines what information is written to the table's stream. Valid values for `StreamViewType` are:
  - `KEYS_ONLY` - Only the key attributes of the modified item are written to the stream.
  - `NEW_IMAGE` - The entire item, as it appears after it was modified, is written to the stream.
  - `OLD_IMAGE` - The entire item, as it appeared before it was modified, is written to the stream.
  - `NEW_AND_OLD_IMAGES` - Both the new and the old item images of the item are written to the stream.

Type: StreamSpecification object

Required: No

TableClass

The table class of the new table. Valid values are STANDARD and STANDARD_INFREQUENT_ACCESS.

Type: String

Valid Values: STANDARD | STANDARD_INFREQUENT_ACCESS

Required: No

Tags

A list of key-value pairs to label the table. For more information, see Tagging for DynamoDB.

Type: Array of Tag objects

Required: No

WarmThroughput

Represents the warm throughput (in read units per second and write units per second) for creating a table.

Type: WarmThroughput object

Required: No

Response Syntax

```
{
   "TableDescription": {
      "ArchivalSummary": {
         "ArchivalBackupArn": "string",
         "ArchivalDateTime": number,
         "ArchivalReason": "string"
      },
      "AttributeDefinitions": [
         {
            "AttributeName": "string",
            "AttributeType": "string"
         }
      ],
      "BillingModeSummary": {
         "BillingMode": "string",
```

```
            "LastUpdateToPayPerRequestDateTime": number
        },
        "CreationDateTime": number,
        "DeletionProtectionEnabled": boolean,
        "GlobalSecondaryIndexes": [
            {
                "Backfilling": boolean,
                "IndexArn": "string",
                "IndexName": "string",
                "IndexSizeBytes": number,
                "IndexStatus": "string",
                "ItemCount": number,
                "KeySchema": [
                    {
                        "AttributeName": "string",
                        "KeyType": "string"
                    }
                ],
                "OnDemandThroughput": {
                    "MaxReadRequestUnits": number,
                    "MaxWriteRequestUnits": number
                },
                "Projection": {
                    "NonKeyAttributes": [ "string" ],
                    "ProjectionType": "string"
                },
                "ProvisionedThroughput": {
                    "LastDecreaseDateTime": number,
                    "LastIncreaseDateTime": number,
                    "NumberOfDecreasesToday": number,
                    "ReadCapacityUnits": number,
                    "WriteCapacityUnits": number
                },
                "WarmThroughput": {
                    "ReadUnitsPerSecond": number,
                    "Status": "string",
                    "WriteUnitsPerSecond": number
                }
            }
        ],
        "GlobalTableVersion": "string",
        "GlobalTableWitnesses": [
            {
                "RegionName": "string",
```

```
            "WitnessStatus": "string"
         }
      ],
      "ItemCount": number,
      "KeySchema": [
         {
            "AttributeName": "string",
            "KeyType": "string"
         }
      ],
      "LatestStreamArn": "string",
      "LatestStreamLabel": "string",
      "LocalSecondaryIndexes": [
         {
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            }
         }
      ],
      "MultiRegionConsistency": "string",
      "OnDemandThroughput": {
         "MaxReadRequestUnits": number,
         "MaxWriteRequestUnits": number
      },
      "ProvisionedThroughput": {
         "LastDecreaseDateTime": number,
         "LastIncreaseDateTime": number,
         "NumberOfDecreasesToday": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
      "Replicas": [
         {
```

```
            "GlobalSecondaryIndexes": [
               {
                  "IndexName": "string",
                  "OnDemandThroughputOverride": {
                     "MaxReadRequestUnits": number
                  },
                  "ProvisionedThroughputOverride": {
                     "ReadCapacityUnits": number
                  },
                  "WarmThroughput": {
                     "ReadUnitsPerSecond": number,
                     "Status": "string",
                     "WriteUnitsPerSecond": number
                  }
               }
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
               "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
               "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "ReplicaInaccessibleDateTime": number,
            "ReplicaStatus": "string",
            "ReplicaStatusDescription": "string",
            "ReplicaStatusPercentProgress": "string",
            "ReplicaTableClassSummary": {
               "LastUpdateDateTime": number,
               "TableClass": "string"
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "RestoreSummary": {
         "RestoreDateTime": number,
         "RestoreInProgress": boolean,
         "SourceBackupArn": "string",
         "SourceTableArn": "string"
```

```
        },
        "SSEDescription": {
           "InaccessibleEncryptionDateTime": number,
           "KMSMasterKeyArn": "string",
           "SSEType": "string",
           "Status": "string"
        },
        "StreamSpecification": {
           "StreamEnabled": boolean,
           "StreamViewType": "string"
        },
        "TableArn": "string",
        "TableClassSummary": {
           "LastUpdateDateTime": number,
           "TableClass": "string"
        },
        "TableId": "string",
        "TableName": "string",
        "TableSizeBytes": number,
        "TableStatus": "string",
        "WarmThroughput": {
           "ReadUnitsPerSecond": number,
           "Status": "string",
           "WriteUnitsPerSecond": number
        }
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TableDescription

Represents the properties of the table.

Type: TableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

Examples

Create a Table

This example creates a table named `Thread`. The table primary key consists of `ForumName` (partition key) and `Subject` (sort key). A local secondary index is also created; its key consists of `ForumName` (partition key) and `LastPostDateTime` (sort key).

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.CreateTable

{
    "AttributeDefinitions": [
        {
            "AttributeName": "ForumName",
            "AttributeType": "S"
        },
        {
            "AttributeName": "Subject",
            "AttributeType": "S"
        },
        {
            "AttributeName": "LastPostDateTime",
            "AttributeType": "S"
        }
    ],
    "TableName": "Thread",
    "KeySchema": [
        {
            "AttributeName": "ForumName",
            "KeyType": "HASH"
        },
        {
```

```
            "AttributeName": "Subject",
            "KeyType": "RANGE"
        }
    ],
    "LocalSecondaryIndexes": [
        {
            "IndexName": "LastPostIndex",
            "KeySchema": [
                {
                    "AttributeName": "ForumName",
                    "KeyType": "HASH"
                },
                {
                    "AttributeName": "LastPostDateTime",
                    "KeyType": "RANGE"
                }
            ],
            "Projection": {
                "ProjectionType": "KEYS_ONLY"
            }
        }
    ],
    "BillingMode": "PAY_PER_REQUEST",
    "Tags": [
      {
         "Key": "Owner",
         "Value": "BlueTeam"
      }
   ]
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "TableDescription": {
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Thread",
        "AttributeDefinitions": [
```

```
            {
                "AttributeName": "ForumName",
                "AttributeType": "S"
            },
            {
                "AttributeName": "LastPostDateTime",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Subject",
                "AttributeType": "S"
            }
        ],
        "CreationDateTime": 1.36372808007E9,
        "ItemCount": 0,
        "KeySchema": [
            {
                "AttributeName": "ForumName",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "Subject",
                "KeyType": "RANGE"
            }
        ],
        "LocalSecondaryIndexes": [
            {
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Thread/
index/LastPostIndex",
                "IndexName": "LastPostIndex",
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "KeySchema": [
                    {
                        "AttributeName": "ForumName",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "LastPostDateTime",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "KEYS_ONLY"
```

```
                }
            }
        ],
        "TableName": "Thread",
        "TableSizeBytes": 0,
        "TableStatus": "CREATING"
    }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteBackup

Service: Amazon DynamoDB

Deletes an existing backup of a table.

You can call `DeleteBackup` at a maximum rate of 10 times per second.

Request Syntax

```
{
   "BackupArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

BackupArn

The ARN associated with the backup.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "BackupDescription": {
      "BackupDetails": {
         "BackupArn": "string",
         "BackupCreationDateTime": number,
         "BackupExpiryDateTime": number,
         "BackupName": "string",
         "BackupSizeBytes": number,
```

```
            "BackupStatus": "string",
            "BackupType": "string"
        },
        "SourceTableDetails": {
            "BillingMode": "string",
            "ItemCount": number,
            "KeySchema": [
                {
                    "AttributeName": "string",
                    "KeyType": "string"
                }
            ],
            "OnDemandThroughput": {
                "MaxReadRequestUnits": number,
                "MaxWriteRequestUnits": number
            },
            "ProvisionedThroughput": {
                "ReadCapacityUnits": number,
                "WriteCapacityUnits": number
            },
            "TableArn": "string",
            "TableCreationDateTime": number,
            "TableId": "string",
            "TableName": "string",
            "TableSizeBytes": number
        },
        "SourceTableFeatureDetails": {
            "GlobalSecondaryIndexes": [
                {
                    "IndexName": "string",
                    "KeySchema": [
                        {
                            "AttributeName": "string",
                            "KeyType": "string"
                        }
                    ],
                    "OnDemandThroughput": {
                        "MaxReadRequestUnits": number,
                        "MaxWriteRequestUnits": number
                    },
                    "Projection": {
                        "NonKeyAttributes": [ "string" ],
                        "ProjectionType": "string"
                    },
```

```
                "ProvisionedThroughput": {
                   "ReadCapacityUnits": number,
                   "WriteCapacityUnits": number
                }
             }
          ],
          "LocalSecondaryIndexes": [
             {
                "IndexName": "string",
                "KeySchema": [
                   {
                      "AttributeName": "string",
                      "KeyType": "string"
                   }
                ],
                "Projection": {
                   "NonKeyAttributes": [ "string" ],
                   "ProjectionType": "string"
                }
             }
          ],
          "SSEDescription": {
             "InaccessibleEncryptionDateTime": number,
             "KMSMasterKeyArn": "string",
             "SSEType": "string",
             "Status": "string"
          },
          "StreamDescription": {
             "StreamEnabled": boolean,
             "StreamViewType": "string"
          },
          "TimeToLiveDescription": {
             "AttributeName": "string",
             "TimeToLiveStatus": "string"
          }
       }
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BackupDescription

Contains the description of the backup created for the table.

Type: BackupDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

BackupInUseException

There is another ongoing conflicting backup control plane operation on the table. The backup is either being created, deleted or restored to a table.

HTTP Status Code: 400

BackupNotFoundException

Backup not found for the given BackupARN.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteItem

Service: Amazon DynamoDB

Deletes a single item in a table by primary key. You can perform a conditional delete operation that deletes the item if it exists, or if it has an expected attribute value.

In addition to deleting an item, you can also return the item's attribute values in the same operation, using the `ReturnValues` parameter.

Unless you specify conditions, the `DeleteItem` is an idempotent operation; running it multiple times on the same item or attribute does *not* result in an error response.

Conditional deletes are useful for deleting items only if specific conditions are met. If those conditions are met, DynamoDB performs the delete. Otherwise, the item is not deleted.

Request Syntax

```
{
   "ConditionalOperator": "string",
   "ConditionExpression": "string",
   "Expected": {
      "string" : {
         "AttributeValueList": [
            {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
                  "AttributeValue"
               ],
               "M": {
                  "string" : "AttributeValue"
               },
               "N": "string",
               "NS": [ "string" ],
               "NULL": boolean,
               "S": "string",
               "SS": [ "string" ]
            }
         ],
         "ComparisonOperator": "string",
         "Exists": boolean,
         "Value": {
```

```
            "B": blob,
            "BOOL": boolean,
            "BS": [ blob ],
            "L": [
               "AttributeValue"
            ],
            "M": {
               "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
         }
      }
   },
   "ExpressionAttributeNames": {
      "string" : "string"
   },
   "ExpressionAttributeValues": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "Key": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
```

```
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ReturnConsumedCapacity": "string",
   "ReturnItemCollectionMetrics": "string",
   "ReturnValues": "string",
   "ReturnValuesOnConditionCheckFailure": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

Key

A map of attribute names to `AttributeValue` objects, representing the primary key of the item to delete.

For the primary key, you must provide all of the key attributes. For example, with a simple primary key, you only need to provide a value for the partition key. For a composite primary key, you must provide values for both the partition key and the sort key.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

The name of the table from which to delete the item. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ConditionalOperator

This is a legacy parameter. Use `ConditionExpression` instead. For more information, see ConditionalOperator in the *Amazon DynamoDB Developer Guide*.

Type: String

Valid Values: `AND | OR`

Required: No

ConditionExpression

A condition that must be satisfied in order for a conditional `DeleteItem` to succeed.

An expression can contain any of the following:

- Functions: `attribute_exists | attribute_not_exists | attribute_type | contains | begins_with | size`

  These function names are case-sensitive.
- Comparison operators: `= | <> | < | > | <= | >= | BETWEEN | IN`
- Logical operators: `AND | OR | NOT`

For more information about condition expressions, see Condition Expressions in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

Expected

This is a legacy parameter. Use `ConditionExpression` instead. For more information, see Expected in the *Amazon DynamoDB Developer Guide*.

Type: String to ExpectedAttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression. The following are some use cases for using `ExpressionAttributeNames`:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the **#** character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for `ExpressionAttributeNames`:

- `{"#P":"Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

> **Note**
>
> Tokens that begin with the **:** character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see Specifying Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression.

Use the **:** (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify `ExpressionAttributeValues` as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"}, ":disc":
{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see Condition Expressions in the *Amazon DynamoDB Developer Guide*.

Type: String to AttributeValue object map

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

ReturnItemCollectionMetrics

Determines whether item collection metrics are returned. If set to `SIZE`, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to `NONE` (the default), no statistics are returned.

Type: String

Valid Values: `SIZE` | `NONE`

Required: No

ReturnValues

Use `ReturnValues` if you want to get the item attributes as they appeared before they were deleted. For `DeleteItem`, the valid values are:

- `NONE` - If `ReturnValues` is not specified, or if its value is `NONE`, then nothing is returned. (This setting is the default for `ReturnValues`.)
- `ALL_OLD` - The content of the old item is returned.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

> **Note**
>
> The `ReturnValues` parameter is used by several DynamoDB operations; however, `DeleteItem` does not recognize any values other than `NONE` or `ALL_OLD`.

Type: String

Valid Values: `NONE` | `ALL_OLD` | `UPDATED_OLD` | `ALL_NEW` | `UPDATED_NEW`

Required: No

ReturnValuesOnConditionCheckFailure

An optional parameter that returns the item attributes for a `DeleteItem` operation that failed a condition check.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

Type: String

Valid Values: `ALL_OLD | NONE`

Required: No

Response Syntax

```
{
   "Attributes": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ConsumedCapacity": {
      "CapacityUnits": number,
      "GlobalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "LocalSecondaryIndexes": {
         "string" : {
```

```
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "ReadCapacityUnits": number,
      "Table": {
         "CapacityUnits": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
      "TableName": "string",
      "WriteCapacityUnits": number
   },
   "ItemCollectionMetrics": {
      "ItemCollectionKey": {
         "string" : {
            "B": blob,
            "BOOL": boolean,
            "BS": [ blob ],
            "L": [
               "AttributeValue"
            ],
            "M": {
               "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
         }
      },
      "SizeEstimateRangeGB": [ number ]
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Attributes

A map of attribute names to `AttributeValue` objects, representing the item as it appeared before the `DeleteItem` operation. This map appears in the response only if `ReturnValues` was specified as `ALL_OLD` in the request.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

ConsumedCapacity

The capacity units consumed by the `DeleteItem` operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. `ConsumedCapacity` is only returned if the `ReturnConsumedCapacity` parameter was specified. For more information, see Provisioned capacity mode in the *Amazon DynamoDB Developer Guide*.

Type: ConsumedCapacity object

ItemCollectionMetrics

Information about item collections, if any, that were affected by the `DeleteItem` operation. `ItemCollectionMetrics` is only returned if the `ReturnItemCollectionMetrics` parameter was specified. If the table does not have any local secondary indexes, this information is not returned in the response.

Each `ItemCollectionMetrics` element consists of:

- `ItemCollectionKey` - The partition key value of the item collection. This is the same as the partition key value of the item itself.
- `SizeEstimateRangeGB` - An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

  The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

Type: ItemCollectionMetrics object

Errors

For information about the errors that are common to all actions, see Common Errors.

ConditionalCheckFailedException

A condition specified in the operation failed to be evaluated.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ItemCollectionSizeLimitExceededException

An item collection is too large. This exception is only returned for tables that have one or more local secondary indexes.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

ReplicatedWriteConflictException

The request was rejected because one or more items in the request are being modified by a request in another Region.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

TransactionConflictException

Operation was rejected because there is an ongoing transaction for the item.

HTTP Status Code: 400

Examples

Delete an Item

The following example deletes an item from the `Thread` table, but only if that item does not already have an attribute named `Replies`. Because `ReturnValues` is set to `ALL_OLD`, the response contains the item as it appeared before the delete.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.DeleteItem
```

```
{
    "TableName": "Thread",
    "Key": {
        "ForumName": {
            "S": "Amazon DynamoDB"
        },
        "Subject": {
            "S": "How do I update multiple items?"
        }
    },
    "ConditionExpression": "attribute_not_exists(Replies)",
    "ReturnValues": "ALL_OLD"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "Attributes": {
        "LastPostedBy": {
            "S": "fred@example.com"
        },
        "ForumName": {
            "S": "Amazon DynamoDB"
        },
        "LastPostDateTime": {
            "S": "201303201023"
        },
        "Tags": {
            "SS": ["Update","Multiple Items","HelpMe"]
        },
        "Subject": {
            "S": "How do I update multiple items?"
        },
        "Message": {
            "S": "I want to update multiple items in a single call. What's the best way
 to do that?"
        }
```

```
    }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteResourcePolicy

Service: Amazon DynamoDB

Deletes the resource-based policy attached to the resource, which can be a table or stream.

`DeleteResourcePolicy` is an idempotent operation; running it multiple times on the same resource *doesn't* result in an error response, unless you specify an `ExpectedRevisionId`, which will then return a `PolicyNotFoundException`.

> **Important**
>
> To make sure that you don't inadvertently lock yourself out of your own resources, the root principal in your AWS account can perform `DeleteResourcePolicy` requests, even if your resource-based policy explicitly denies the root principal's access.

> **Note**
>
> `DeleteResourcePolicy` is an asynchronous operation. If you issue a `GetResourcePolicy` request immediately after running the `DeleteResourcePolicy` request, DynamoDB might still return the deleted policy. This is because the policy for your resource might not have been deleted yet. Wait for a few seconds, and then try the `GetResourcePolicy` request again.

Request Syntax

```
{
   "ExpectedRevisionId": "string",
   "ResourceArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ResourceArn

The Amazon Resource Name (ARN) of the DynamoDB resource from which the policy will be removed. The resources you can specify include tables and streams. If you remove the policy of a table, it will also remove the permissions for the table's indexes defined in that policy document. This is because index permissions are defined in the table's policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1283.

Required: Yes

ExpectedRevisionId

A string value that you can use to conditionally delete your policy. When you provide an expected revision ID, if the revision ID of the existing policy on the resource doesn't match or if there's no policy attached to the resource, the request will fail and return a `PolicyNotFoundException`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

Response Syntax

```
{
   "RevisionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

RevisionId

A unique string that represents the revision ID of the policy. If you're comparing revision IDs, make sure to always use string comparison logic.

This value will be empty if you make a request against a resource without a policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

PolicyNotFoundException

The operation tried to access a nonexistent resource-based policy.

If you specified an `ExpectedRevisionId`, it's possible that a policy is present for the resource but its revision ID didn't match the expected value.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Delete the resource-based policy of a table

The following example deletes the resource-based policy of a table named `Thread`.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
```

```
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.DeleteResourcePolicy
{
    "ResourceArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Thread"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "RevisionId": "1683717331354"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteTable

Service: Amazon DynamoDB

The `DeleteTable` operation deletes a table and all of its items. After a `DeleteTable` request, the specified table is in the `DELETING` state until DynamoDB completes the deletion. If the table is in the `ACTIVE` state, you can delete it. If a table is in `CREATING` or `UPDATING` states, then DynamoDB returns a `ResourceInUseException`. If the specified table does not exist, DynamoDB returns a `ResourceNotFoundException`. If table is already in the `DELETING` state, no error is returned.

> **Note**
>
> DynamoDB might continue to accept data read and write operations, such as `GetItem` and `PutItem`, on a table in the `DELETING` state until the table deletion is complete. For the full list of table states, see TableStatus.

When you delete a table, any indexes on that table are also deleted.

If you have DynamoDB Streams enabled on the table, then the corresponding stream on that table goes into the `DISABLED` state, and the stream is automatically deleted after 24 hours.

Use the `DescribeTable` action to check the status of the table.

Request Syntax

```
{
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table to delete. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "TableDescription": {
      "ArchivalSummary": {
         "ArchivalBackupArn": "string",
         "ArchivalDateTime": number,
         "ArchivalReason": "string"
      },
      "AttributeDefinitions": [
         {
            "AttributeName": "string",
            "AttributeType": "string"
         }
      ],
      "BillingModeSummary": {
         "BillingMode": "string",
         "LastUpdateToPayPerRequestDateTime": number
      },
      "CreationDateTime": number,
      "DeletionProtectionEnabled": boolean,
      "GlobalSecondaryIndexes": [
         {
            "Backfilling": boolean,
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "IndexStatus": "string",
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
```

```
                "KeyType": "string"
            }
        ],
        "OnDemandThroughput": {
            "MaxReadRequestUnits": number,
            "MaxWriteRequestUnits": number
        },
        "Projection": {
            "NonKeyAttributes": [ "string" ],
            "ProjectionType": "string"
        },
        "ProvisionedThroughput": {
            "LastDecreaseDateTime": number,
            "LastIncreaseDateTime": number,
            "NumberOfDecreasesToday": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
        },
        "WarmThroughput": {
            "ReadUnitsPerSecond": number,
            "Status": "string",
            "WriteUnitsPerSecond": number
        }
    }
],
"GlobalTableVersion": "string",
"GlobalTableWitnesses": [
    {
        "RegionName": "string",
        "WitnessStatus": "string"
    }
],
"ItemCount": number,
"KeySchema": [
    {
        "AttributeName": "string",
        "KeyType": "string"
    }
],
"LatestStreamArn": "string",
"LatestStreamLabel": "string",
"LocalSecondaryIndexes": [
    {
        "IndexArn": "string",
```

```
            "IndexName": "string",
            "IndexSizeBytes": number,
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            }
         }
      ],
      "MultiRegionConsistency": "string",
      "OnDemandThroughput": {
         "MaxReadRequestUnits": number,
         "MaxWriteRequestUnits": number
      },
      "ProvisionedThroughput": {
         "LastDecreaseDateTime": number,
         "LastIncreaseDateTime": number,
         "NumberOfDecreasesToday": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
      "Replicas": [
         {
            "GlobalSecondaryIndexes": [
               {
                  "IndexName": "string",
                  "OnDemandThroughputOverride": {
                     "MaxReadRequestUnits": number
                  },
                  "ProvisionedThroughputOverride": {
                     "ReadCapacityUnits": number
                  },
                  "WarmThroughput": {
                     "ReadUnitsPerSecond": number,
                     "Status": "string",
                     "WriteUnitsPerSecond": number
                  }
               }
```

```
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
               "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
               "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "ReplicaInaccessibleDateTime": number,
            "ReplicaStatus": "string",
            "ReplicaStatusDescription": "string",
            "ReplicaStatusPercentProgress": "string",
            "ReplicaTableClassSummary": {
               "LastUpdateDateTime": number,
               "TableClass": "string"
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "RestoreSummary": {
         "RestoreDateTime": number,
         "RestoreInProgress": boolean,
         "SourceBackupArn": "string",
         "SourceTableArn": "string"
      },
      "SSEDescription": {
         "InaccessibleEncryptionDateTime": number,
         "KMSMasterKeyArn": "string",
         "SSEType": "string",
         "Status": "string"
      },
      "StreamSpecification": {
         "StreamEnabled": boolean,
         "StreamViewType": "string"
      },
      "TableArn": "string",
      "TableClassSummary": {
         "LastUpdateDateTime": number,
         "TableClass": "string"
```

```
        },
        "TableId": "string",
        "TableName": "string",
        "TableSizeBytes": number,
        "TableStatus": "string",
        "WarmThroughput": {
           "ReadUnitsPerSecond": number,
           "Status": "string",
           "WriteUnitsPerSecond": number
        }
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TableDescription

Represents the properties of a table.

Type: TableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Delete a Table

This example deletes the Reply table.

Sample Request

```
POST / HTTP/1.1
```

```
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.DeleteTable

{
    "TableName": "Reply"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "TableDescription": {
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Reply",
        "ItemCount": 0,
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableName": "Reply",
        "TableSizeBytes": 0,
        "TableStatus": "DELETING"
    }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeBackup

Service: Amazon DynamoDB

Describes an existing backup of a table.

You can call `DescribeBackup` at a maximum rate of 10 times per second.

Request Syntax

```
{
   "BackupArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

BackupArn

The Amazon Resource Name (ARN) associated with the backup.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "BackupDescription": {
      "BackupDetails": {
         "BackupArn": "string",
         "BackupCreationDateTime": number,
         "BackupExpiryDateTime": number,
         "BackupName": "string",
         "BackupSizeBytes": number,
```

```
            "BackupStatus": "string",
            "BackupType": "string"
        },
        "SourceTableDetails": {
            "BillingMode": "string",
            "ItemCount": number,
            "KeySchema": [
                {
                    "AttributeName": "string",
                    "KeyType": "string"
                }
            ],
            "OnDemandThroughput": {
                "MaxReadRequestUnits": number,
                "MaxWriteRequestUnits": number
            },
            "ProvisionedThroughput": {
                "ReadCapacityUnits": number,
                "WriteCapacityUnits": number
            },
            "TableArn": "string",
            "TableCreationDateTime": number,
            "TableId": "string",
            "TableName": "string",
            "TableSizeBytes": number
        },
        "SourceTableFeatureDetails": {
            "GlobalSecondaryIndexes": [
                {
                    "IndexName": "string",
                    "KeySchema": [
                        {
                            "AttributeName": "string",
                            "KeyType": "string"
                        }
                    ],
                    "OnDemandThroughput": {
                        "MaxReadRequestUnits": number,
                        "MaxWriteRequestUnits": number
                    },
                    "Projection": {
                        "NonKeyAttributes": [ "string" ],
                        "ProjectionType": "string"
                    },
```

```
                "ProvisionedThroughput": {
                   "ReadCapacityUnits": number,
                   "WriteCapacityUnits": number
                }
             }
          ],
          "LocalSecondaryIndexes": [
             {
                "IndexName": "string",
                "KeySchema": [
                   {
                      "AttributeName": "string",
                      "KeyType": "string"
                   }
                ],
                "Projection": {
                   "NonKeyAttributes": [ "string" ],
                   "ProjectionType": "string"
                }
             }
          ],
          "SSEDescription": {
             "InaccessibleEncryptionDateTime": number,
             "KMSMasterKeyArn": "string",
             "SSEType": "string",
             "Status": "string"
          },
          "StreamDescription": {
             "StreamEnabled": boolean,
             "StreamViewType": "string"
          },
          "TimeToLiveDescription": {
             "AttributeName": "string",
             "TimeToLiveStatus": "string"
          }
       }
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BackupDescription

Contains the description of the backup created for the table.

Type: BackupDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

BackupNotFoundException

Backup not found for the given BackupARN.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeContinuousBackups

Service: Amazon DynamoDB

Checks the status of continuous backups and point in time recovery on the specified table. Continuous backups are `ENABLED` on all tables at table creation. If point in time recovery is enabled, `PointInTimeRecoveryStatus` will be set to ENABLED.

After continuous backups and point in time recovery are enabled, you can restore to any point in time within `EarliestRestorableDateTime` and `LatestRestorableDateTime`.

`LatestRestorableDateTime` is typically 5 minutes before the current time. You can restore your table to any point in time in the last 35 days. You can set the recovery period to any value between 1 and 35 days.

You can call `DescribeContinuousBackups` at a maximum rate of 10 times per second.

Request Syntax

```
{
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

Name of the table for which the customer wants to check the continuous backups and point in time recovery settings.

You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "ContinuousBackupsDescription": {
      "ContinuousBackupsStatus": "string",
      "PointInTimeRecoveryDescription": {
         "EarliestRestorableDateTime": number,
         "LatestRestorableDateTime": number,
         "PointInTimeRecoveryStatus": "string",
         "RecoveryPeriodInDays": number
      }
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ContinuousBackupsDescription

Represents the continuous backups and point in time recovery settings on the table.

Type: ContinuousBackupsDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

TableNotFoundException

A source table with the name `TableName` does not currently exist within the subscriber's account or the subscriber is operating in the wrong AWS Region.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeContributorInsights

Service: Amazon DynamoDB

Returns information about contributor insights for a given table or global secondary index.

Request Syntax

```
{
   "IndexName": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table to describe. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

IndexName

The name of the global secondary index to describe, if applicable.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Response Syntax

```
{
   "ContributorInsightsMode": "string",
   "ContributorInsightsRuleList": [ "string" ],
   "ContributorInsightsStatus": "string",
   "FailureException": {
      "ExceptionDescription": "string",
      "ExceptionName": "string"
   },
   "IndexName": "string",
   "LastUpdateDateTime": number,
   "TableName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ContributorInsightsMode

The mode of CloudWatch Contributor Insights for DynamoDB that determines which events are emitted. Can be set to track all access and throttled events or throttled events only.

Type: String

Valid Values: `ACCESSED_AND_THROTTLED_KEYS | THROTTLED_KEYS`

ContributorInsightsRuleList

List of names of the associated contributor insights rules.

Type: Array of strings

Pattern: `[A-Za-z0-9][A-Za-z0-9\-\_\.]{0,126}[A-Za-z0-9]`

ContributorInsightsStatus

Current status of contributor insights.

Type: String

Valid Values: `ENABLING | ENABLED | DISABLING | DISABLED | FAILED`

FailureException

Returns information about the last failure that was encountered.

The most common exceptions for a FAILED status are:

- LimitExceededException - Per-account Amazon CloudWatch Contributor Insights rule limit reached. Please disable Contributor Insights for other tables/indexes OR disable Contributor Insights rules before retrying.
- AccessDeniedException - Amazon CloudWatch Contributor Insights rules cannot be modified due to insufficient permissions.
- AccessDeniedException - Failed to create service-linked role for Contributor Insights due to insufficient permissions.
- InternalServerError - Failed to create Amazon CloudWatch Contributor Insights rules. Please retry request.

Type: FailureException object

IndexName

The name of the global secondary index being described.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

LastUpdateDateTime

Timestamp of the last time the status was changed.

Type: Timestamp

TableName

The name of the table being described.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeEndpoints

Service: Amazon DynamoDB

Returns the regional endpoint information. For more information on policy permissions, please see Internetwork traffic privacy.

Response Syntax

```
{
   "Endpoints": [
      {
         "Address": "string",
         "CachePeriodInMinutes": number
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Endpoints

List of endpoints.

Type: Array of Endpoint objects

Errors

For information about the errors that are common to all actions, see Common Errors.

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeExport

Service: Amazon DynamoDB

Describes an existing table export.

Request Syntax

```
{
   "ExportArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ExportArn

The Amazon Resource Name (ARN) associated with the export.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "ExportDescription": {
      "BilledSizeBytes": number,
      "ClientToken": "string",
      "EndTime": number,
      "ExportArn": "string",
      "ExportFormat": "string",
      "ExportManifest": "string",
```

```
        "ExportStatus": "string",
        "ExportTime": number,
        "ExportType": "string",
        "FailureCode": "string",
        "FailureMessage": "string",
        "IncrementalExportSpecification": {
           "ExportFromTime": number,
           "ExportToTime": number,
           "ExportViewType": "string"
        },
        "ItemCount": number,
        "S3Bucket": "string",
        "S3BucketOwner": "string",
        "S3Prefix": "string",
        "S3SseAlgorithm": "string",
        "S3SseKmsKeyId": "string",
        "StartTime": number,
        "TableArn": "string",
        "TableId": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ExportDescription

Represents the properties of the export.

Type: ExportDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

ExportNotFoundException

The specified export was not found.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeGlobalTable

Service: Amazon DynamoDB

Returns information about the specified global table.

> **Important**
>
> This documentation is for version 2017.11.29 (Legacy) of global tables, which should be avoided for new global tables. Customers should use Global Tables version 2019.11.21 (Current) when possible, because it provides greater flexibility, higher efficiency, and consumes less write capacity than 2017.11.29 (Legacy).
> To determine which version you're using, see Determining the global table version you are using. To update existing global tables from version 2017.11.29 (Legacy) to version 2019.11.21 (Current), see Upgrading global tables.

Request Syntax

```
{
   "GlobalTableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

GlobalTableName

The name of the global table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
   "GlobalTableDescription": {
      "CreationDateTime": number,
      "GlobalTableArn": "string",
      "GlobalTableName": "string",
      "GlobalTableStatus": "string",
      "ReplicationGroup": [
         {
            "GlobalSecondaryIndexes": [
               {
                  "IndexName": "string",
                  "OnDemandThroughputOverride": {
                     "MaxReadRequestUnits": number
                  },
                  "ProvisionedThroughputOverride": {
                     "ReadCapacityUnits": number
                  },
                  "WarmThroughput": {
                     "ReadUnitsPerSecond": number,
                     "Status": "string",
                     "WriteUnitsPerSecond": number
                  }
               }
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
               "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
               "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "ReplicaInaccessibleDateTime": number,
            "ReplicaStatus": "string",
            "ReplicaStatusDescription": "string",
            "ReplicaStatusPercentProgress": "string",
            "ReplicaTableClassSummary": {
               "LastUpdateDateTime": number,
               "TableClass": "string"
```

```
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ]
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GlobalTableDescription

Contains the details of the global table.

Type: GlobalTableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

GlobalTableNotFoundException

The specified global table does not exist.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeGlobalTableSettings

Service: Amazon DynamoDB

Describes Region-specific settings for a global table.

> **Important**
>
> This documentation is for version 2017.11.29 (Legacy) of global tables, which should be avoided for new global tables. Customers should use Global Tables version 2019.11.21 (Current) when possible, because it provides greater flexibility, higher efficiency, and consumes less write capacity than 2017.11.29 (Legacy).
> To determine which version you're using, see Determining the global table version you are using. To update existing global tables from version 2017.11.29 (Legacy) to version 2019.11.21 (Current), see Upgrading global tables.

Request Syntax

```
{
   "GlobalTableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

GlobalTableName

The name of the global table to describe.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

Response Syntax

```
{
   "GlobalTableName": "string",
   "ReplicaSettings": [
      {
         "RegionName": "string",
         "ReplicaBillingModeSummary": {
            "BillingMode": "string",
            "LastUpdateToPayPerRequestDateTime": number
         },
         "ReplicaGlobalSecondaryIndexSettings": [
            {
               "IndexName": "string",
               "IndexStatus": "string",
               "ProvisionedReadCapacityAutoScalingSettings": {
                  "AutoScalingDisabled": boolean,
                  "AutoScalingRoleArn": "string",
                  "MaximumUnits": number,
                  "MinimumUnits": number,
                  "ScalingPolicies": [
                     {
                        "PolicyName": "string",
                        "TargetTrackingScalingPolicyConfiguration": {
                           "DisableScaleIn": boolean,
                           "ScaleInCooldown": number,
                           "ScaleOutCooldown": number,
                           "TargetValue": number
                        }
                     }
                  ]
               },
               "ProvisionedReadCapacityUnits": number,
               "ProvisionedWriteCapacityAutoScalingSettings": {
                  "AutoScalingDisabled": boolean,
                  "AutoScalingRoleArn": "string",
                  "MaximumUnits": number,
                  "MinimumUnits": number,
                  "ScalingPolicies": [
                     {
                        "PolicyName": "string",
```

```
                    "TargetTrackingScalingPolicyConfiguration": {
                       "DisableScaleIn": boolean,
                       "ScaleInCooldown": number,
                       "ScaleOutCooldown": number,
                       "TargetValue": number
                    }
                 }
              ]
           },
           "ProvisionedWriteCapacityUnits": number
        }
     ],
     "ReplicaProvisionedReadCapacityAutoScalingSettings": {
        "AutoScalingDisabled": boolean,
        "AutoScalingRoleArn": "string",
        "MaximumUnits": number,
        "MinimumUnits": number,
        "ScalingPolicies": [
           {
              "PolicyName": "string",
              "TargetTrackingScalingPolicyConfiguration": {
                 "DisableScaleIn": boolean,
                 "ScaleInCooldown": number,
                 "ScaleOutCooldown": number,
                 "TargetValue": number
              }
           }
        ]
     },
     "ReplicaProvisionedReadCapacityUnits": number,
     "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
        "AutoScalingDisabled": boolean,
        "AutoScalingRoleArn": "string",
        "MaximumUnits": number,
        "MinimumUnits": number,
        "ScalingPolicies": [
           {
              "PolicyName": "string",
              "TargetTrackingScalingPolicyConfiguration": {
                 "DisableScaleIn": boolean,
                 "ScaleInCooldown": number,
                 "ScaleOutCooldown": number,
                 "TargetValue": number
              }
```

```
                }
            ]
        },
        "ReplicaProvisionedWriteCapacityUnits": number,
        "ReplicaStatus": "string",
        "ReplicaTableClassSummary": {
            "LastUpdateDateTime": number,
            "TableClass": "string"
        }
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GlobalTableName

The name of the global table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

ReplicaSettings

The Region-specific settings for the global table.

Type: Array of ReplicaSettingsDescription objects

Errors

For information about the errors that are common to all actions, see Common Errors.

GlobalTableNotFoundException

The specified global table does not exist.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeImport

Service: Amazon DynamoDB

Represents the properties of the import.

Request Syntax

```
{
   "ImportArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ImportArn

The Amazon Resource Name (ARN) associated with the table you're importing to.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "ImportTableDescription": {
      "ClientToken": "string",
      "CloudWatchLogGroupArn": "string",
      "EndTime": number,
      "ErrorCount": number,
      "FailureCode": "string",
      "FailureMessage": "string",
      "ImportArn": "string",
      "ImportedItemCount": number,
```

```
      "ImportStatus": "string",
      "InputCompressionType": "string",
      "InputFormat": "string",
      "InputFormatOptions": {
         "Csv": {
            "Delimiter": "string",
            "HeaderList": [ "string" ]
         }
      },
      "ProcessedItemCount": number,
      "ProcessedSizeBytes": number,
      "S3BucketSource": {
         "S3Bucket": "string",
         "S3BucketOwner": "string",
         "S3KeyPrefix": "string"
      },
      "StartTime": number,
      "TableArn": "string",
      "TableCreationParameters": {
         "AttributeDefinitions": [
            {
               "AttributeName": "string",
               "AttributeType": "string"
            }
         ],
         "BillingMode": "string",
         "GlobalSecondaryIndexes": [
            {
               "IndexName": "string",
               "KeySchema": [
                  {
                     "AttributeName": "string",
                     "KeyType": "string"
                  }
               ],
               "OnDemandThroughput": {
                  "MaxReadRequestUnits": number,
                  "MaxWriteRequestUnits": number
               },
               "Projection": {
                  "NonKeyAttributes": [ "string" ],
                  "ProjectionType": "string"
               },
               "ProvisionedThroughput": {
```

```
                "ReadCapacityUnits": number,
                "WriteCapacityUnits": number
            },
            "WarmThroughput": {
                "ReadUnitsPerSecond": number,
                "WriteUnitsPerSecond": number
            }
        }
    ],
    "KeySchema": [
        {
            "AttributeName": "string",
            "KeyType": "string"
        }
    ],
    "OnDemandThroughput": {
        "MaxReadRequestUnits": number,
        "MaxWriteRequestUnits": number
    },
    "ProvisionedThroughput": {
        "ReadCapacityUnits": number,
        "WriteCapacityUnits": number
    },
    "SSESpecification": {
        "Enabled": boolean,
        "KMSMasterKeyId": "string",
        "SSEType": "string"
    },
    "TableName": "string"
  },
  "TableId": "string"
}
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ImportTableDescription

Represents the properties of the table created for the import, and parameters of the import. The import parameters include import status, how many items were processed, and how many errors were encountered.

Type: ImportTableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

ImportNotFoundException

The specified import was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeKinesisStreamingDestination

Service: Amazon DynamoDB

Returns information about the status of Kinesis streaming.

Request Syntax

```
{
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table being described. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "KinesisDataStreamDestinations": [
      {
         "ApproximateCreationDateTimePrecision": "string",
         "DestinationStatus": "string",
         "DestinationStatusDescription": "string",
         "StreamArn": "string"
```

```
        }
    ],
    "TableName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

KinesisDataStreamDestinations

The list of replica structures for the table being described.

Type: Array of KinesisDataStreamDestination objects

TableName

The name of the table being described.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeLimits

Service: Amazon DynamoDB

Returns the current provisioned-capacity quotas for your AWS account in a Region, both for the Region as a whole and for any one DynamoDB table that you create there.

When you establish an AWS account, the account has initial quotas on the maximum read capacity units and write capacity units that you can provision across all of your DynamoDB tables in a given Region. Also, there are per-table quotas that apply when you create a table there. For more information, see Service, Account, and Table Quotas page in the *Amazon DynamoDB Developer Guide*.

Although you can increase these quotas by filing a case at AWS Support Center, obtaining the increase is not instantaneous. The `DescribeLimits` action lets you write code to compare the capacity you are currently using to those quotas imposed by your account so that you have enough time to apply for an increase before you hit a quota.

For example, you could use one of the AWS SDKs to do the following:

1. Call `DescribeLimits` for a particular Region to obtain your current account quotas on provisioned capacity there.
2. Create a variable to hold the aggregate read capacity units provisioned for all your tables in that Region, and one to hold the aggregate write capacity units. Zero them both.
3. Call `ListTables` to obtain a list of all your DynamoDB tables.
4. For each table name listed by `ListTables`, do the following:
   - Call `DescribeTable` with the table name.
   - Use the data returned by `DescribeTable` to add the read capacity units and write capacity units provisioned for the table itself to your variables.
   - If the table has one or more global secondary indexes (GSIs), loop over these GSIs and add their provisioned capacity values to your variables as well.
5. Report the account quotas for that Region returned by `DescribeLimits`, along with the total current provisioned capacity levels you have calculated.

This will let you see whether you are getting close to your account-level quotas.

The per-table quotas apply only when you are creating a new table. They restrict the sum of the provisioned capacity of the new table itself and all its global secondary indexes.

For existing tables and their GSIs, DynamoDB doesn't let you increase provisioned capacity extremely rapidly, but the only quota that applies is that the aggregate provisioned capacity over all your tables and GSIs cannot exceed either of the per-account quotas.

> **Note**
>
> `DescribeLimits` should only be called periodically. You can expect throttling errors if you call it more than once in a minute.

The `DescribeLimits` Request element has no content.

Response Syntax

```
{
   "AccountMaxReadCapacityUnits": number,
   "AccountMaxWriteCapacityUnits": number,
   "TableMaxReadCapacityUnits": number,
   "TableMaxWriteCapacityUnits": number
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

AccountMaxReadCapacityUnits

The maximum total read capacity units that your account allows you to provision across all of your tables in this Region.

Type: Long

Valid Range: Minimum value of 1.

AccountMaxWriteCapacityUnits

The maximum total write capacity units that your account allows you to provision across all of your tables in this Region.

Type: Long

Valid Range: Minimum value of 1.

TableMaxReadCapacityUnits

The maximum read capacity units that your account allows you to provision for a new table that you are creating in this Region, including the read capacity units provisioned for its global secondary indexes (GSIs).

Type: Long

Valid Range: Minimum value of 1.

TableMaxWriteCapacityUnits

The maximum write capacity units that your account allows you to provision for a new table that you are creating in this Region, including the write capacity units provisioned for its global secondary indexes (GSIs).

Type: Long

Valid Range: Minimum value of 1.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

Examples

DescribeLimits

This example illustrates one usage of DescribeLimits.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
```

```
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.DescribeLimits
{ }
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "AccountMaxReadCapacityUnits": 20000,
    "AccountMaxWriteCapacityUnits": 20000,
    "TableMaxReadCapacityUnits": 10000,
    "TableMaxWriteCapacityUnits": 10000
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

DescribeTable

Service: Amazon DynamoDB

Returns information about the table, including the current status of the table, when it was created, the primary key schema, and any indexes on the table.

> **Note**
>
> If you issue a `DescribeTable` request immediately after a `CreateTable` request, DynamoDB might return a `ResourceNotFoundException`. This is because `DescribeTable` uses an eventually consistent query, and the metadata for your table might not be available at that moment. Wait for a few seconds, and then try the `DescribeTable` request again.

Request Syntax

```
{
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table to describe. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "Table": {
      "ArchivalSummary": {
         "ArchivalBackupArn": "string",
         "ArchivalDateTime": number,
         "ArchivalReason": "string"
      },
      "AttributeDefinitions": [
         {
            "AttributeName": "string",
            "AttributeType": "string"
         }
      ],
      "BillingModeSummary": {
         "BillingMode": "string",
         "LastUpdateToPayPerRequestDateTime": number
      },
      "CreationDateTime": number,
      "DeletionProtectionEnabled": boolean,
      "GlobalSecondaryIndexes": [
         {
            "Backfilling": boolean,
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "IndexStatus": "string",
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "OnDemandThroughput": {
               "MaxReadRequestUnits": number,
               "MaxWriteRequestUnits": number
            },
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            },
```

```
            "ProvisionedThroughput": {
               "LastDecreaseDateTime": number,
               "LastIncreaseDateTime": number,
               "NumberOfDecreasesToday": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "GlobalTableVersion": "string",
      "GlobalTableWitnesses": [
         {
            "RegionName": "string",
            "WitnessStatus": "string"
         }
      ],
      "ItemCount": number,
      "KeySchema": [
         {
            "AttributeName": "string",
            "KeyType": "string"
         }
      ],
      "LatestStreamArn": "string",
      "LatestStreamLabel": "string",
      "LocalSecondaryIndexes": [
         {
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "Projection": {
               "NonKeyAttributes": [ "string" ],
```

```
                "ProjectionType": "string"
            }
        }
    ],
    "MultiRegionConsistency": "string",
    "OnDemandThroughput": {
        "MaxReadRequestUnits": number,
        "MaxWriteRequestUnits": number
    },
    "ProvisionedThroughput": {
        "LastDecreaseDateTime": number,
        "LastIncreaseDateTime": number,
        "NumberOfDecreasesToday": number,
        "ReadCapacityUnits": number,
        "WriteCapacityUnits": number
    },
    "Replicas": [
        {
            "GlobalSecondaryIndexes": [
                {
                    "IndexName": "string",
                    "OnDemandThroughputOverride": {
                        "MaxReadRequestUnits": number
                    },
                    "ProvisionedThroughputOverride": {
                        "ReadCapacityUnits": number
                    },
                    "WarmThroughput": {
                        "ReadUnitsPerSecond": number,
                        "Status": "string",
                        "WriteUnitsPerSecond": number
                    }
                }
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
                "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
                "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "ReplicaInaccessibleDateTime": number,
            "ReplicaStatus": "string",
```

```
            "ReplicaStatusDescription": "string",
            "ReplicaStatusPercentProgress": "string",
            "ReplicaTableClassSummary": {
               "LastUpdateDateTime": number,
               "TableClass": "string"
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "RestoreSummary": {
         "RestoreDateTime": number,
         "RestoreInProgress": boolean,
         "SourceBackupArn": "string",
         "SourceTableArn": "string"
      },
      "SSEDescription": {
         "InaccessibleEncryptionDateTime": number,
         "KMSMasterKeyArn": "string",
         "SSEType": "string",
         "Status": "string"
      },
      "StreamSpecification": {
         "StreamEnabled": boolean,
         "StreamViewType": "string"
      },
      "TableArn": "string",
      "TableClassSummary": {
         "LastUpdateDateTime": number,
         "TableClass": "string"
      },
      "TableId": "string",
      "TableName": "string",
      "TableSizeBytes": number,
      "TableStatus": "string",
      "WarmThroughput": {
         "ReadUnitsPerSecond": number,
         "Status": "string",
         "WriteUnitsPerSecond": number
      }
   }
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Table

The properties of the table.

Type: TableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Describe a Table

This example describes the `Thread` table.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
```

```
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.DescribeTable

{
    "TableName":"Thread"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "Table": {
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Thread",
        "AttributeDefinitions": [
            {
                "AttributeName": "ForumName",
                "AttributeType": "S"
            },
            {
                "AttributeName": "LastPostDateTime",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Subject",
                "AttributeType": "S"
            }
        ],
        "CreationDateTime": 1.363729002358E9,
        "ItemCount": 0,
        "KeySchema": [
            {
                "AttributeName": "ForumName",
                "KeyType": "HASH"
```

```
            },
            {
                "AttributeName": "Subject",
                "KeyType": "RANGE"
            }
        ],
        "LocalSecondaryIndexes": [
            {
                "IndexArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Thread/
index/LastPostIndex",
                "IndexName": "LastPostIndex",
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "KeySchema": [
                    {
                        "AttributeName": "ForumName",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "LastPostDateTime",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "KEYS_ONLY"
                }
            }
        ],
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableName": "Thread",
        "TableSizeBytes": 0,
        "TableStatus": "ACTIVE"
    }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeTableReplicaAutoScaling

Service: Amazon DynamoDB

Describes auto scaling settings across replicas of the global table at once.

Request Syntax

```
{
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "TableAutoScalingDescription": {
      "Replicas": [
         {
            "GlobalSecondaryIndexes": [
               {
                  "IndexName": "string",
                  "IndexStatus": "string",
```

```
                "ProvisionedReadCapacityAutoScalingSettings": {
                   "AutoScalingDisabled": boolean,
                   "AutoScalingRoleArn": "string",
                   "MaximumUnits": number,
                   "MinimumUnits": number,
                   "ScalingPolicies": [
                      {
                         "PolicyName": "string",
                         "TargetTrackingScalingPolicyConfiguration": {
                            "DisableScaleIn": boolean,
                            "ScaleInCooldown": number,
                            "ScaleOutCooldown": number,
                            "TargetValue": number
                         }
                      }
                   ]
                },
                "ProvisionedWriteCapacityAutoScalingSettings": {
                   "AutoScalingDisabled": boolean,
                   "AutoScalingRoleArn": "string",
                   "MaximumUnits": number,
                   "MinimumUnits": number,
                   "ScalingPolicies": [
                      {
                         "PolicyName": "string",
                         "TargetTrackingScalingPolicyConfiguration": {
                            "DisableScaleIn": boolean,
                            "ScaleInCooldown": number,
                            "ScaleOutCooldown": number,
                            "TargetValue": number
                         }
                      }
                   ]
                }
             }
          ],
          "RegionName": "string",
          "ReplicaProvisionedReadCapacityAutoScalingSettings": {
             "AutoScalingDisabled": boolean,
             "AutoScalingRoleArn": "string",
             "MaximumUnits": number,
             "MinimumUnits": number,
             "ScalingPolicies": [
                {
```

```
                    "PolicyName": "string",
                    "TargetTrackingScalingPolicyConfiguration": {
                       "DisableScaleIn": boolean,
                       "ScaleInCooldown": number,
                       "ScaleOutCooldown": number,
                       "TargetValue": number
                    }
                 }
              ]
           },
           "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
              "AutoScalingDisabled": boolean,
              "AutoScalingRoleArn": "string",
              "MaximumUnits": number,
              "MinimumUnits": number,
              "ScalingPolicies": [
                 {
                    "PolicyName": "string",
                    "TargetTrackingScalingPolicyConfiguration": {
                       "DisableScaleIn": boolean,
                       "ScaleInCooldown": number,
                       "ScaleOutCooldown": number,
                       "TargetValue": number
                    }
                 }
              ]
           },
           "ReplicaStatus": "string"
        }
     ],
     "TableName": "string",
     "TableStatus": "string"
  }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TableAutoScalingDescription

Represents the auto scaling properties of the table.

Type: TableAutoScalingDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeTimeToLive

Service: Amazon DynamoDB

Gives a description of the Time to Live (TTL) status on the specified table.

Request Syntax

```
{
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table to be described. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "TimeToLiveDescription": {
      "AttributeName": "string",
      "TimeToLiveStatus": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TimeToLiveDescription

Type: TimeToLiveDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DisableKinesisStreamingDestination

Service: Amazon DynamoDB

Stops replication from the DynamoDB table to the Kinesis data stream. This is done without deleting either of the resources.

Request Syntax

```
{
   "EnableKinesisStreamingConfiguration": {
      "ApproximateCreationDateTimePrecision": "string"
   },
   "StreamArn": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

StreamArn

The ARN for a Kinesis data stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

TableName

The name of the DynamoDB table. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

EnableKinesisStreamingConfiguration

The source for the Kinesis streaming information that is being enabled.

Type: EnableKinesisStreamingConfiguration object

Required: No

Response Syntax

```
{
   "DestinationStatus": "string",
   "EnableKinesisStreamingConfiguration": {
      "ApproximateCreationDateTimePrecision": "string"
   },
   "StreamArn": "string",
   "TableName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DestinationStatus

The current status of the replication.

Type: String

Valid Values: `ENABLING | ACTIVE | DISABLING | DISABLED | ENABLE_FAILED | UPDATING`

EnableKinesisStreamingConfiguration

The destination for the Kinesis streaming information that is being enabled.

Type: EnableKinesisStreamingConfiguration object

StreamArn

The ARN for the specific Kinesis data stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

TableName

The name of the table being modified.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

EnableKinesisStreamingDestination

Service: Amazon DynamoDB

Starts table data replication to the specified Kinesis data stream at a timestamp chosen during the enable workflow. If this operation doesn't return results immediately, use DescribeKinesisStreamingDestination to check if streaming to the Kinesis data stream is ACTIVE.

Request Syntax

```
{
   "EnableKinesisStreamingConfiguration": {
      "ApproximateCreationDateTimePrecision": "string"
   },
   "StreamArn": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

StreamArn

The ARN for a Kinesis data stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

TableName

The name of the DynamoDB table. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

EnableKinesisStreamingConfiguration

The source for the Kinesis streaming information that is being enabled.

Type: EnableKinesisStreamingConfiguration object

Required: No

Response Syntax

```
{
   "DestinationStatus": "string",
   "EnableKinesisStreamingConfiguration": {
      "ApproximateCreationDateTimePrecision": "string"
   },
   "StreamArn": "string",
   "TableName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DestinationStatus

The current status of the replication.

Type: String

Valid Values: `ENABLING | ACTIVE | DISABLING | DISABLED | ENABLE_FAILED | UPDATING`

EnableKinesisStreamingConfiguration

The destination for the Kinesis streaming information that is being enabled.

Type: EnableKinesisStreamingConfiguration object

StreamArn

The ARN for the specific Kinesis data stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

TableName

The name of the table being modified.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

ExecuteStatement

Service: Amazon DynamoDB

This operation allows you to perform reads and singleton writes on data stored in DynamoDB, using PartiQL.

For PartiQL reads (`SELECT` statement), if the total number of processed items exceeds the maximum dataset size limit of 1 MB, the read stops and results are returned to the user as a `LastEvaluatedKey` value to continue the read in a subsequent operation. If the filter criteria in `WHERE` clause does not match any data, the read will return an empty result set.

A single `SELECT` statement response can return up to the maximum number of items (if using the Limit parameter) or a maximum of 1 MB of data (and then apply any filtering to the results using `WHERE` clause). If `LastEvaluatedKey` is present in the response, you need to paginate the result set. If `NextToken` is present, you need to paginate the result set and include `NextToken`.

Request Syntax

```
{
   "ConsistentRead": boolean,
   "Limit": number,
   "NextToken": "string",
   "Parameters": [
      {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   ],
   "ReturnConsumedCapacity": "string",
   "ReturnValuesOnConditionCheckFailure": "string",
```

```
    "Statement": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

Statement

The PartiQL statement representing the operation to run.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: Yes

ConsistentRead

The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

Type: Boolean

Required: No

Limit

The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, along with a key in `LastEvaluatedKey` to apply in a subsequent operation so you can pick up where you left off. Also, if the processed dataset size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in `LastEvaluatedKey` to apply in a subsequent operation to continue the operation.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

NextToken

Set this value to get remaining results, if `NextToken` was returned in the statement response.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32768.

Required: No

Parameters

The parameters for the PartiQL statement, if any.

Type: Array of AttributeValue objects

Array Members: Minimum number of 1 item.

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

ReturnValuesOnConditionCheckFailure

An optional parameter that returns the item attributes for an `ExecuteStatement` operation that failed a condition check.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

Type: String

Valid Values: `ALL_OLD` | `NONE`

Required: No

Response Syntax

```
{
   "ConsumedCapacity": {
      "CapacityUnits": number,
      "GlobalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "LocalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "ReadCapacityUnits": number,
      "Table": {
         "CapacityUnits": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
      "TableName": "string",
      "WriteCapacityUnits": number
   },
   "Items": [
      {
         "string" : {
            "B": blob,
```

```
            "BOOL": boolean,
            "BS": [ blob ],
            "L": [
               "AttributeValue"
            ],
            "M": {
               "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
         }
      }
   ],
   "LastEvaluatedKey": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. `ConsumedCapacity` is only returned if the request asked for it. For more information, see Provisioned capacity mode in the *Amazon DynamoDB Developer Guide*.

Type: ConsumedCapacity object

Items

If a read operation was used, this property will contain the result of the read operation; a map of attribute names and their values. For the write operations this value will be empty.

Type: Array of string to AttributeValue object maps

Key Length Constraints: Maximum length of 65535.

LastEvaluatedKey

The primary key of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request. If `LastEvaluatedKey` is empty, then the "last page" of results has been processed and there is no more data to be retrieved. If `LastEvaluatedKey` is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when `LastEvaluatedKey` is empty.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

NextToken

If the response of a read request exceeds the response payload limit DynamoDB will set this value in the response. If set, you can use that this value in the subsequent request to get the remaining results.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 32768.

Errors

For information about the errors that are common to all actions, see Common Errors.

ConditionalCheckFailedException

A condition specified in the operation failed to be evaluated.

HTTP Status Code: 400

DuplicateItemException

There was an attempt to insert an item with the same primary key as an item that already exists in the DynamoDB table.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ItemCollectionSizeLimitExceededException

An item collection is too large. This exception is only returned for tables that have one or more local secondary indexes.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

TransactionConflictException

Operation was rejected because there is an ongoing transaction for the item.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExecuteTransaction

Service: Amazon DynamoDB

This operation allows you to perform transactional reads or writes on data stored in DynamoDB, using PartiQL.

> **Note**
>
> The entire transaction must consist of either read statements or write statements, you cannot mix both in one transaction. The EXISTS function is an exception and can be used to check the condition of specific attributes of the item in a similar manner to `ConditionCheck` in the TransactWriteItems API.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "ReturnConsumedCapacity": "string",
   "TransactStatements": [
      {
         "Parameters": [
            {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
                  "AttributeValue"
               ],
               "M": {
                  "string" : "AttributeValue"
               },
               "N": "string",
               "NS": [ "string" ],
               "NULL": boolean,
               "S": "string",
               "SS": [ "string" ]
            }
         ],
         "ReturnValuesOnConditionCheckFailure": "string",
         "Statement": "string"
```

```
        }
    ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TransactStatements

The list of PartiQL statements representing the transaction to run.

Type: Array of ParameterizedStatement objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: Yes

ClientRequestToken

Set this value to get remaining results, if `NextToken` was returned in the statement response.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 36.

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response. For more information, see TransactGetItems and TransactWriteItems.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

Response Syntax

```
{
   "ConsumedCapacity": [
      {
         "CapacityUnits": number,
         "GlobalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "LocalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "ReadCapacityUnits": number,
         "Table": {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         },
         "TableName": "string",
         "WriteCapacityUnits": number
      }
   ],
   "Responses": [
      {
         "Item": {
            "string" : {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
                  "AttributeValue"
               ],
               "M": {
```

```
                    "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
            }
        }
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The capacity units consumed by the entire operation. The values of the list are ordered according to the ordering of the statements.

Type: Array of ConsumedCapacity objects

Responses

The response to a PartiQL transaction.

Type: Array of ItemResponse objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Errors

For information about the errors that are common to all actions, see Common Errors.

IdempotentParameterMismatchException

DynamoDB rejected the request because you retried a request with a different payload but with an idempotent token that was already used.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

TransactionCanceledException

The entire transaction request was canceled.

DynamoDB cancels a `TransactWriteItems` request under the following circumstances:

- A condition in one of the condition expressions is not met.

- A table in the `TransactWriteItems` request is in a different account or region.
- More than one action in the `TransactWriteItems` operation targets the same item.
- There is insufficient provisioned capacity for the transaction to be completed.
- An item size becomes too large (larger than 400 KB), or a local secondary index (LSI) becomes too large, or a similar validation error occurs because of changes made by the transaction.
- There is a user error, such as an invalid data format.
- There is an ongoing `TransactWriteItems` operation that conflicts with a concurrent `TransactWriteItems` request. In this case the `TransactWriteItems` operation fails with a `TransactionCanceledException`.

DynamoDB cancels a `TransactGetItems` request under the following circumstances:

- There is an ongoing `TransactGetItems` operation that conflicts with a concurrent `PutItem`, `UpdateItem`, `DeleteItem` or `TransactWriteItems` request. In this case the `TransactGetItems` operation fails with a `TransactionCanceledException`.
- A table in the `TransactGetItems` request is in a different account or region.
- There is insufficient provisioned capacity for the transaction to be completed.
- There is a user error, such as an invalid data format.

> **Note**
>
> If using Java, DynamoDB lists the cancellation reasons on the `CancellationReasons` property. This property is not set for other languages. Transaction cancellation reasons are ordered in the order of requested items, if an item has no error it will have `None` code and `Null` message.

Cancellation reason codes and possible error messages:

- No Errors:
  - Code: `None`
  - Message: `null`
- Conditional Check Failed:
  - Code: `ConditionalCheckFailed`
  - Message: The conditional request failed.
- Item Collection Size Limit Exceeded:

  - Code: `ItemCollectionSizeLimitExceeded`
  - Message: Collection size exceeded.
- Transaction Conflict:
  - Code: `TransactionConflict`
  - Message: Transaction is ongoing for the item.
- Provisioned Throughput Exceeded:
  - Code: `ProvisionedThroughputExceeded`
  - Messages:
    - The level of configured provisioned throughput for the table was exceeded. Consider increasing your provisioning level with the UpdateTable API.

      > **Note**
      >
      > This Message is received when provisioned throughput is exceeded is on a provisioned DynamoDB table.

    - The level of configured provisioned throughput for one or more global secondary indexes of the table was exceeded. Consider increasing your provisioning level for the under-provisioned global secondary indexes with the UpdateTable API.

      > **Note**
      >
      > This message is returned when provisioned throughput is exceeded is on a provisioned GSI.

- Throttling Error:
  - Code: `ThrottlingError`
  - Messages:
    - Throughput exceeds the current capacity of your table or index. DynamoDB is automatically scaling your table or index so please try again shortly. If exceptions persist, check if you have a hot key: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html.

> **Note**
>
> This message is returned when writes get throttled on an On-Demand table as DynamoDB is automatically scaling the table.

- Throughput exceeds the current capacity for one or more global secondary indexes. DynamoDB is automatically scaling your index so please try again shortly.

> **Note**
>
> This message is returned when writes get throttled on an On-Demand GSI as DynamoDB is automatically scaling the GSI.

- Validation Error:
  - Code: `ValidationError`
  - Messages:
    - One or more parameter values were invalid.
    - The update expression attempted to update the secondary index key beyond allowed size limits.
    - The update expression attempted to update the secondary index key to unsupported type.
    - An operand in the update expression has an incorrect data type.
    - Item size to update has exceeded the maximum allowed size.
    - Number overflow. Attempting to store a number with magnitude larger than supported range.
    - Type mismatch for attribute to update.
    - Nesting Levels have exceeded supported limits.
    - The document path provided in the update expression is invalid for update.
    - The provided expression refers to an attribute that does not exist in the item.

HTTP Status Code: 400

TransactionInProgressException

The transaction with the given request token is already in progress.

Recommended Settings

> **Note**
>
> This is a general recommendation for handling the `TransactionInProgressException`. These settings help ensure that the client retries will trigger completion of the ongoing `TransactWriteItems` request.

- Set `clientExecutionTimeout` to a value that allows at least one retry to be processed after 5 seconds have elapsed since the first attempt for the `TransactWriteItems` operation.
- Set `socketTimeout` to a value a little lower than the `requestTimeout` setting.
- `requestTimeout` should be set based on the time taken for the individual retries of a single HTTP request for your use case, but setting it to 1 second or higher should work well to reduce chances of retries and `TransactionInProgressException` errors.
- Use exponential backoff when retrying and tune backoff if needed.

Assuming default retry policy, example timeout settings based on the guidelines above are as follows:

Example timeline:

- 0-1000 first attempt
- 1000-1500 first sleep/delay (default retry policy uses 500 ms as base delay for 4xx errors)
- 1500-2500 second attempt
- 2500-3500 second sleep/delay (500 * 2, exponential backoff)
- 3500-4500 third attempt
- 4500-6500 third sleep/delay (500 * 2^2)
- 6500-7500 fourth attempt (this can trigger inline recovery since 5 seconds have elapsed since the first attempt reached TC)

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ExportTableToPointInTime

Service: Amazon DynamoDB

Exports table data to an S3 bucket. The table must have point in time recovery enabled, and you can export data from any time within the point in time recovery window.

Request Syntax

```
{
   "ClientToken": "string",
   "ExportFormat": "string",
   "ExportTime": number,
   "ExportType": "string",
   "IncrementalExportSpecification": {
      "ExportFromTime": number,
      "ExportToTime": number,
      "ExportViewType": "string"
   },
   "S3Bucket": "string",
   "S3BucketOwner": "string",
   "S3Prefix": "string",
   "S3SseAlgorithm": "string",
   "S3SseKmsKeyId": "string",
   "TableArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

S3Bucket

The name of the Amazon S3 bucket to export the snapshot to.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `^[a-z0-9A-Z]+[\.\-\w]*[a-z0-9A-Z]+$`

Required: Yes

TableArn

The Amazon Resource Name (ARN) associated with the table to export.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ClientToken

Providing a `ClientToken` makes the call to `ExportTableToPointInTimeInput` idempotent, meaning that multiple identical calls have the same effect as one single call.

A client token is valid for 8 hours after the first request that uses it is completed. After 8 hours, any request with the same client token is treated as a new request. Do not resubmit the same request with the same client token for more than 8 hours, or the result might not be idempotent.

If you submit a request with the same client token but a change in other parameters within the 8-hour idempotency window, DynamoDB returns an `ImportConflictException`.

Type: String

Pattern: `^[^\$]+$`

Required: No

ExportFormat

The format for the exported data. Valid values for `ExportFormat` are `DYNAMODB_JSON` or `ION`.

Type: String

Valid Values: `DYNAMODB_JSON | ION`

Required: No

ExportTime

Time in the past from which to export table data, counted in seconds from the start of the Unix epoch. The table export will be a snapshot of the table's state at this point in time.

Type: Timestamp

Required: No

ExportType

Choice of whether to execute as a full export or incremental export. Valid values are FULL_EXPORT or INCREMENTAL_EXPORT. The default value is FULL_EXPORT. If INCREMENTAL_EXPORT is provided, the IncrementalExportSpecification must also be used.

Type: String

Valid Values: `FULL_EXPORT | INCREMENTAL_EXPORT`

Required: No

IncrementalExportSpecification

Optional object containing the parameters specific to an incremental export.

Type: IncrementalExportSpecification object

Required: No

S3BucketOwner

The ID of the AWS account that owns the bucket the export will be stored in.

> **Note**
>
> S3BucketOwner is a required parameter when exporting to a S3 bucket in another account.

Type: String

Pattern: `[0-9]{12}`

Required: No

S3Prefix

The Amazon S3 bucket prefix to use as the file name and path of the exported snapshot.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

S3SseAlgorithm

Type of encryption used on the bucket where export data will be stored. Valid values for `S3SseAlgorithm` are:

- `AES256` - server-side encryption with Amazon S3 managed keys
- `KMS` - server-side encryption with AWS KMS managed keys

Type: String

Valid Values: `AES256 | KMS`

Required: No

S3SseKmsKeyId

The ID of the AWS KMS managed key used to encrypt the S3 bucket where export data will be stored (if applicable).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

Response Syntax

```
{
   "ExportDescription": {
      "BilledSizeBytes": number,
      "ClientToken": "string",
      "EndTime": number,
      "ExportArn": "string",
      "ExportFormat": "string",
      "ExportManifest": "string",
      "ExportStatus": "string",
      "ExportTime": number,
      "ExportType": "string",
      "FailureCode": "string",
```

```
        "FailureMessage": "string",
        "IncrementalExportSpecification": {
           "ExportFromTime": number,
           "ExportToTime": number,
           "ExportViewType": "string"
        },
        "ItemCount": number,
        "S3Bucket": "string",
        "S3BucketOwner": "string",
        "S3Prefix": "string",
        "S3SseAlgorithm": "string",
        "S3SseKmsKeyId": "string",
        "StartTime": number,
        "TableArn": "string",
        "TableId": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ExportDescription

Contains a description of the table export.

Type: ExportDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

ExportConflictException

There was a conflict when writing to the specified S3 bucket.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

InvalidExportTimeException

The specified `ExportTime` is outside of the point in time recovery window.

HTTP Status Code: 400

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

PointInTimeRecoveryUnavailableException

Point in time recovery has not yet been enabled for this source table.

HTTP Status Code: 400

TableNotFoundException

A source table with the name `TableName` does not currently exist within the subscriber's account or the subscriber is operating in the wrong AWS Region.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetItem

Service: Amazon DynamoDB

The `GetItem` operation returns a set of attributes for the item with the given primary key. If there is no matching item, `GetItem` does not return any data and there will be no `Item` element in the response.

`GetItem` provides an eventually consistent read by default. If your application requires a strongly consistent read, set `ConsistentRead` to `true`. Although a strongly consistent read might take more time than an eventually consistent read, it always returns the last updated value.

Request Syntax

```
{
   "AttributesToGet": [ "string" ],
   "ConsistentRead": boolean,
   "ExpressionAttributeNames": {
      "string" : "string"
   },
   "Key": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ProjectionExpression": "string",
   "ReturnConsumedCapacity": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

Key

A map of attribute names to `AttributeValue` objects, representing the primary key of the item to retrieve.

For the primary key, you must provide all of the attributes. For example, with a simple primary key, you only need to provide a value for the partition key. For a composite primary key, you must provide values for both the partition key and the sort key.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

The name of the table containing the requested item. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

AttributesToGet

This is a legacy parameter. Use `ProjectionExpression` instead. For more information, see AttributesToGet in the *Amazon DynamoDB Developer Guide*.

Type: Array of strings

Array Members: Minimum number of 1 item.

Length Constraints: Maximum length of 65535.

Required: No

ConsistentRead

Determines the read consistency model: If set to `true`, then the operation uses strongly consistent reads; otherwise, the operation uses eventually consistent reads.

Type: Boolean

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression. The following are some use cases for using `ExpressionAttributeNames`:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the **#** character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for `ExpressionAttributeNames`:

- `{"#P":"Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

> **Note**
>
> Tokens that begin with the **:** character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see Specifying Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ProjectionExpression

A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes are returned. If any of the requested attributes are not found, they do not appear in the result.

For more information, see Specifying Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

Response Syntax

```
{
   "ConsumedCapacity": {
      "CapacityUnits": number,
      "GlobalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "LocalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "ReadCapacityUnits": number,
      "Table": {
         "CapacityUnits": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
      "TableName": "string",
      "WriteCapacityUnits": number
   },
   "Item": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
```

```
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
        }
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The capacity units consumed by the `GetItem` operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. `ConsumedCapacity` is only returned if the `ReturnConsumedCapacity` parameter was specified. For more information, see Capacity unit consumption for read operations in the *Amazon DynamoDB Developer Guide*.

Type: ConsumedCapacity object

Item

A map of attribute names to `AttributeValue` objects, as specified by `ProjectionExpression`.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

Examples

Retrieve Item Attributes

The following example retrieves three attributes from the `Thread` table. In the response, the `ConsumedCapacityUnits` value is 1, because `ConsistentRead` is set to true. If `ConsistentRead` had been set to false (or not specified) for the same request, an eventually consistent read would have been used and `ConsumedCapacityUnits` would have been 0.5.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.GetItem

{
    "TableName": "Thread",
    "Key": {
        "ForumName": {
            "S": "Amazon DynamoDB"
        },
        "Subject": {
            "S": "How do I update multiple items?"
        }
    },
    "ProjectionExpression":"LastPostDateTime, Message, Tags",
    "ConsistentRead": true,
    "ReturnConsumedCapacity": "TOTAL"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "ConsumedCapacity": {
        "CapacityUnits": 1,
        "TableName": "Thread"
    },
    "Item": {
        "Tags": {
```

```
            "SS": ["Update","Multiple Items","HelpMe"]
        },
        "LastPostDateTime": {
            "S": "201303190436"
        },
        "Message": {
            "S": "I want to update multiple items in a single call. What's the best way
 to do that?"
        }
    }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetResourcePolicy

Service: Amazon DynamoDB

Returns the resource-based policy document attached to the resource, which can be a table or stream, in JSON format.

`GetResourcePolicy` follows an *eventually consistent* model. The following list describes the outcomes when you issue the `GetResourcePolicy` request immediately after issuing another request:

- If you issue a `GetResourcePolicy` request immediately after a `PutResourcePolicy` request, DynamoDB might return a `PolicyNotFoundException`.
- If you issue a `GetResourcePolicy`request immediately after a `DeleteResourcePolicy` request, DynamoDB might return the policy that was present before the deletion request.
- If you issue a `GetResourcePolicy` request immediately after a `CreateTable` request, which includes a resource-based policy, DynamoDB might return a `ResourceNotFoundException` or a `PolicyNotFoundException`.

Because `GetResourcePolicy` uses an *eventually consistent* query, the metadata for your policy or table might not be available at that moment. Wait for a few seconds, and then retry the `GetResourcePolicy` request.

After a `GetResourcePolicy` request returns a policy created using the `PutResourcePolicy` request, the policy will be applied in the authorization of requests to the resource. Because this process is eventually consistent, it will take some time to apply the policy to all requests to a resource. Policies that you attach while creating a table using the `CreateTable` request will always be applied to all requests for that table.

Request Syntax

```
{
   "ResourceArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ResourceArn

The Amazon Resource Name (ARN) of the DynamoDB resource to which the policy is attached. The resources you can specify include tables and streams.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1283.

Required: Yes

Response Syntax

```
{
   "Policy": "string",
   "RevisionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Policy

The resource-based policy document attached to the resource, which can be a table or stream, in JSON format.

Type: String

RevisionId

A unique string that represents the revision ID of the policy. If you're comparing revision IDs, make sure to always use string comparison logic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

PolicyNotFoundException

The operation tried to access a nonexistent resource-based policy.

If you specified an `ExpectedRevisionId`, it's possible that a policy is present for the resource but its revision ID didn't match the expected value.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Get the resource-based policy of a table

The following example retrieves the resource-based policy of a table named `Thread`.

Sample Request

```
GET / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
```

```
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.GetResourcePolicy
{
    "ResourceArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Thread"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\",
\"arn:aws:iam::444455556666:root\"]},\"Action\":[\"dynamodb:GetItem\"],\"Resource\":
\"arn:aws:dynamodb:us-west-2:123456789012:table/Thread\"}}",
    "RevisionId": "1683717331354"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ImportTable

Service: Amazon DynamoDB

Imports table data from an S3 bucket.

Request Syntax

```
{
   "ClientToken": "string",
   "InputCompressionType": "string",
   "InputFormat": "string",
   "InputFormatOptions": {
      "Csv": {
         "Delimiter": "string",
         "HeaderList": [ "string" ]
      }
   },
   "S3BucketSource": {
      "S3Bucket": "string",
      "S3BucketOwner": "string",
      "S3KeyPrefix": "string"
   },
   "TableCreationParameters": {
      "AttributeDefinitions": [
         {
            "AttributeName": "string",
            "AttributeType": "string"
         }
      ],
      "BillingMode": "string",
      "GlobalSecondaryIndexes": [
         {
            "IndexName": "string",
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "OnDemandThroughput": {
               "MaxReadRequestUnits": number,
               "MaxWriteRequestUnits": number
            },
```

```
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            },
            "ProvisionedThroughput": {
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "KeySchema": [
         {
            "AttributeName": "string",
            "KeyType": "string"
         }
      ],
      "OnDemandThroughput": {
         "MaxReadRequestUnits": number,
         "MaxWriteRequestUnits": number
      },
      "ProvisionedThroughput": {
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
      "SSESpecification": {
         "Enabled": boolean,
         "KMSMasterKeyId": "string",
         "SSEType": "string"
      },
      "TableName": "string"
   }
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

InputFormat

The format of the source data. Valid values for `ImportFormat` are `CSV`, `DYNAMODB_JSON` or `ION`.

Type: String

Valid Values: `DYNAMODB_JSON | ION | CSV`

Required: Yes

S3BucketSource

The S3 bucket that provides the source for the import.

Type: S3BucketSource object

Required: Yes

TableCreationParameters

Parameters for the table to import the data into.

Type: TableCreationParameters object

Required: Yes

ClientToken

Providing a `ClientToken` makes the call to `ImportTableInput` idempotent, meaning that multiple identical calls have the same effect as one single call.

A client token is valid for 8 hours after the first request that uses it is completed. After 8 hours, any request with the same client token is treated as a new request. Do not resubmit the same request with the same client token for more than 8 hours, or the result might not be idempotent.

If you submit a request with the same client token but a change in other parameters within the 8-hour idempotency window, DynamoDB returns an `IdempotentParameterMismatch` exception.

Type: String

Pattern: `^[^\$]+$`

Required: No

InputCompressionType

Type of compression to be used on the input coming from the imported table.

Type: String

Valid Values: `GZIP | ZSTD | NONE`

Required: No

InputFormatOptions

Additional properties that specify how the input is formatted,

Type: InputFormatOptions object

Required: No

Response Syntax

```
{
   "ImportTableDescription": {
      "ClientToken": "string",
      "CloudWatchLogGroupArn": "string",
      "EndTime": number,
      "ErrorCount": number,
      "FailureCode": "string",
      "FailureMessage": "string",
      "ImportArn": "string",
      "ImportedItemCount": number,
      "ImportStatus": "string",
      "InputCompressionType": "string",
      "InputFormat": "string",
      "InputFormatOptions": {
         "Csv": {
            "Delimiter": "string",
            "HeaderList": [ "string" ]
         }
      },
```

```
      "ProcessedItemCount": number,
      "ProcessedSizeBytes": number,
      "S3BucketSource": {
         "S3Bucket": "string",
         "S3BucketOwner": "string",
         "S3KeyPrefix": "string"
      },
      "StartTime": number,
      "TableArn": "string",
      "TableCreationParameters": {
         "AttributeDefinitions": [
            {
               "AttributeName": "string",
               "AttributeType": "string"
            }
         ],
         "BillingMode": "string",
         "GlobalSecondaryIndexes": [
            {
               "IndexName": "string",
               "KeySchema": [
                  {
                     "AttributeName": "string",
                     "KeyType": "string"
                  }
               ],
               "OnDemandThroughput": {
                  "MaxReadRequestUnits": number,
                  "MaxWriteRequestUnits": number
               },
               "Projection": {
                  "NonKeyAttributes": [ "string" ],
                  "ProjectionType": "string"
               },
               "ProvisionedThroughput": {
                  "ReadCapacityUnits": number,
                  "WriteCapacityUnits": number
               },
               "WarmThroughput": {
                  "ReadUnitsPerSecond": number,
                  "WriteUnitsPerSecond": number
               }
            }
         ],
```

```
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "OnDemandThroughput": {
               "MaxReadRequestUnits": number,
               "MaxWriteRequestUnits": number
            },
            "ProvisionedThroughput": {
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "SSESpecification": {
               "Enabled": boolean,
               "KMSMasterKeyId": "string",
               "SSEType": "string"
            },
            "TableName": "string"
         },
         "TableId": "string"
      }
   }
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ImportTableDescription

Represents the properties of the table created for the import, and parameters of the import. The import parameters include import status, how many items were processed, and how many errors were encountered.

Type: ImportTableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

ImportConflictException

There was a conflict when importing from the specified S3 source. This can occur when the current import conflicts with a previous import request that had the same client token.

HTTP Status Code: 400

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListBackups

Service: Amazon DynamoDB

List DynamoDB backups that are associated with an AWS account and weren't made with AWS Backup. To list these backups for a given table, specify `TableName`. `ListBackups` returns a paginated list of results with at most 1 MB worth of items in a page. You can also specify a maximum number of entries to be returned in a page.

In the request, start time is inclusive, but end time is exclusive. Note that these boundaries are for the time at which the original backup was requested.

You can call `ListBackups` a maximum of five times per second.

If you want to retrieve the complete list of backups made with AWS Backup, use the AWS Backup list API.

Request Syntax

```
{
   "BackupType": "string",
   "ExclusiveStartBackupArn": "string",
   "Limit": number,
   "TableName": "string",
   "TimeRangeLowerBound": number,
   "TimeRangeUpperBound": number
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

BackupType

The backups from the table specified by `BackupType` are listed.

Where `BackupType` can be:

- `USER` - On-demand backup created by you. (The default setting if no other backup types are specified.)
- `SYSTEM` - On-demand backup automatically created by DynamoDB.
- `ALL` - All types of on-demand backups (USER and SYSTEM).

Type: String

Valid Values: `USER | SYSTEM | AWS_BACKUP | ALL`

Required: No

ExclusiveStartBackupArn

`LastEvaluatedBackupArn` is the Amazon Resource Name (ARN) of the backup last evaluated when the current page of results was returned, inclusive of the current page of results. This value may be specified as the `ExclusiveStartBackupArn` of a new `ListBackups` operation in order to fetch the next page of results.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

Limit

Maximum number of backups to return at once.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

TableName

Lists the backups from the table specified in `TableName`. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

TimeRangeLowerBound

Only backups created after this time are listed. `TimeRangeLowerBound` is inclusive.

Type: Timestamp

Required: No

TimeRangeUpperBound

Only backups created before this time are listed. `TimeRangeUpperBound` is exclusive.

Type: Timestamp

Required: No

Response Syntax

```
{
   "BackupSummaries": [
      {
         "BackupArn": "string",
         "BackupCreationDateTime": number,
         "BackupExpiryDateTime": number,
         "BackupName": "string",
         "BackupSizeBytes": number,
         "BackupStatus": "string",
         "BackupType": "string",
         "TableArn": "string",
         "TableId": "string",
         "TableName": "string"
      }
   ],
   "LastEvaluatedBackupArn": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

BackupSummaries

List of `BackupSummary` objects.

Type: Array of BackupSummary objects

LastEvaluatedBackupArn

The ARN of the backup last evaluated when the current page of results was returned, inclusive of the current page of results. This value may be specified as the `ExclusiveStartBackupArn` of a new `ListBackups` operation in order to fetch the next page of results.

If `LastEvaluatedBackupArn` is empty, then the last page of results has been processed and there are no more results to be retrieved.

If `LastEvaluatedBackupArn` is not empty, this may or may not indicate that there is more data to be returned. All results are guaranteed to have been returned if and only if no value for `LastEvaluatedBackupArn` is returned.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListContributorInsights

Service: Amazon DynamoDB

Returns a list of ContributorInsightsSummary for a table and all its global secondary indexes.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

MaxResults

Maximum number of results to return per page.

Type: Integer

Valid Range: Maximum value of 100.

Required: No

NextToken

A token to for the desired page, if there is one.

Type: String

Required: No

TableName

The name of the table. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

Response Syntax

```
{
   "ContributorInsightsSummaries": [
      {
         "ContributorInsightsMode": "string",
         "ContributorInsightsStatus": "string",
         "IndexName": "string",
         "TableName": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ContributorInsightsSummaries

A list of ContributorInsightsSummary.

Type: Array of ContributorInsightsSummary objects

NextToken

A token to go to the next page if there is one.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListExports

Service: Amazon DynamoDB

Lists completed exports within the past 90 days.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "TableArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

MaxResults

Maximum number of results to return per page.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 25.

Required: No

NextToken

An optional string that, if supplied, must be copied from the output of a previous call to `ListExports`. When provided in this manner, the API fetches the next page of results.

Type: String

Required: No

TableArn

The Amazon Resource Name (ARN) associated with the exported table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

Response Syntax

```
{
   "ExportSummaries": [
      {
         "ExportArn": "string",
         "ExportStatus": "string",
         "ExportType": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ExportSummaries

A list of `ExportSummary` objects.

Type: Array of ExportSummary objects

NextToken

If this value is returned, there are additional results to be displayed. To retrieve them, call `ListExports` again, with `NextToken` set to this value.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListGlobalTables

Service: Amazon DynamoDB

Lists all global tables that have a replica in the specified Region.

> **Important**
>
> This documentation is for version 2017.11.29 (Legacy) of global tables, which should be avoided for new global tables. Customers should use Global Tables version 2019.11.21 (Current) when possible, because it provides greater flexibility, higher efficiency, and consumes less write capacity than 2017.11.29 (Legacy).
> To determine which version you're using, see Determining the global table version you are using. To update existing global tables from version 2017.11.29 (Legacy) to version 2019.11.21 (Current), see Upgrading global tables.

Request Syntax

```
{
   "ExclusiveStartGlobalTableName": "string",
   "Limit": number,
   "RegionName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ExclusiveStartGlobalTableName

The first global table name that this operation will evaluate.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Limit

The maximum number of table names to return, if the parameter is not specified DynamoDB defaults to 100.

If the number of global tables DynamoDB finds reaches this limit, it stops the operation and returns the table names collected up to that point, with a table name in the `LastEvaluatedGlobalTableName` to apply in a subsequent operation to the `ExclusiveStartGlobalTableName` parameter.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

RegionName

Lists the global tables in a specific Region.

Type: String

Required: No

Response Syntax

```
{
   "GlobalTables": [
      {
         "GlobalTableName": "string",
         "ReplicationGroup": [
            {
               "RegionName": "string"
            }
         ]
      }
   ],
   "LastEvaluatedGlobalTableName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GlobalTables

List of global table names.

Type: Array of GlobalTable objects

LastEvaluatedGlobalTableName

Last evaluated global table name.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2

- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListImports

Service: Amazon DynamoDB

Lists completed imports within the past 90 days.

Request Syntax

```
{
   "NextToken": "string",
   "PageSize": number,
   "TableArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

NextToken

An optional string that, if supplied, must be copied from the output of a previous call to `ListImports`. When provided in this manner, the API fetches the next page of results.

Type: String

Length Constraints: Minimum length of 112. Maximum length of 1024.

Pattern: `([0-9a-f]{16})+`

Required: No

PageSize

The number of `ImportSummary` objects returned in a single page.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 25.

Required: No

TableArn

The Amazon Resource Name (ARN) associated with the table that was imported to.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

Response Syntax

```
{
   "ImportSummaryList": [
      {
         "CloudWatchLogGroupArn": "string",
         "EndTime": number,
         "ImportArn": "string",
         "ImportStatus": "string",
         "InputFormat": "string",
         "S3BucketSource": {
            "S3Bucket": "string",
            "S3BucketOwner": "string",
            "S3KeyPrefix": "string"
         },
         "StartTime": number,
         "TableArn": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ImportSummaryList

A list of `ImportSummary` objects.

Type: Array of ImportSummary objects

NextToken

If this value is returned, there are additional results to be displayed. To retrieve them, call `ListImports` again, with `NextToken` set to this value.

Type: String

Length Constraints: Minimum length of 112. Maximum length of 1024.

Pattern: `([0-9a-f]{16})+`

Errors

For information about the errors that are common to all actions, see Common Errors.

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTables

Service: Amazon DynamoDB

Returns an array of table names associated with the current account and endpoint. The output from `ListTables` is paginated, with each page returning a maximum of 100 table names.

Request Syntax

```
{
   "ExclusiveStartTableName": "string",
   "Limit": number
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ExclusiveStartTableName

The first table name that this operation will evaluate. Use the value that was returned for `LastEvaluatedTableName` in a previous operation, so that you can obtain the next page of results.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Limit

A maximum number of table names to return. If this parameter is not specified, the limit is 100.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 100.

Required: No

Response Syntax

```
{
   "LastEvaluatedTableName": "string",
   "TableNames": [ "string" ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LastEvaluatedTableName

The name of the last table in the current page of results. Use this value as the `ExclusiveStartTableName` in a new request to obtain the next page of results, until all the table names are returned.

If you do not receive a `LastEvaluatedTableName` value in the response, this means that there are no more table names to be retrieved.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

TableNames

The names of the tables associated with the current account at the current endpoint. The maximum size of this array is 100.

If `LastEvaluatedTableName` also appears in the output, you can use this value as the `ExclusiveStartTableName` parameter in a subsequent `ListTables` request and obtain the next page of results.

Type: Array of strings

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

Examples

List Tables

This example requests a list of tables, starting with a table named `Forum` and ending after three table names have been returned.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.ListTables

{
    "ExclusiveStartTableName": "Forum",
    "Limit": 3
}
```

Sample Response

```
HTTP/1.1 200 OK
```

```
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "LastEvaluatedTableName": "Thread",
    "TableNames": ["Forum","Reply","Thread"]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsOfResource

Service: Amazon DynamoDB

List all tags on an Amazon DynamoDB resource. You can call ListTagsOfResource up to 10 times per second, per account.

For an overview on tagging DynamoDB resources, see Tagging for DynamoDB in the *Amazon DynamoDB Developer Guide*.

Request Syntax

```
{
   "NextToken": "string",
   "ResourceArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ResourceArn

The Amazon DynamoDB resource with tags to be listed. This value is an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1283.

Required: Yes

NextToken

An optional string that, if supplied, must be copied from the output of a previous call to ListTagOfResource. When provided in this manner, this API fetches the next page of results.

Type: String

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

If this value is returned, there are additional results to be displayed. To retrieve them, call ListTagsOfResource again, with NextToken set to this value.

Type: String

Tags

The tags currently associated with the Amazon DynamoDB resource.

Type: Array of Tag objects

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutItem

Service: Amazon DynamoDB

Creates a new item, or replaces an old item with a new item. If an item that has the same primary key as the new item already exists in the specified table, the new item completely replaces the existing item. You can perform a conditional put operation (add a new item if one with the specified primary key doesn't exist), or replace an existing item if it has certain attribute values. You can return the item's attribute values in the same operation, using the `ReturnValues` parameter.

When you add an item, the primary key attributes are the only required attributes.

Empty String and Binary attribute values are allowed. Attribute values of type String and Binary must have a length greater than zero if the attribute is used as a key attribute for a table or index. Set type attributes cannot be empty.

Invalid Requests with empty values will be rejected with a `ValidationException` exception.

> **Note**
>
> To prevent a new item from replacing an existing item, use a conditional expression that contains the `attribute_not_exists` function with the name of the attribute being used as the partition key for the table. Since every record must contain that attribute, the `attribute_not_exists` function will only succeed if no matching item exists.

For more information about `PutItem`, see Working with Items in the *Amazon DynamoDB Developer Guide*.

Request Syntax

```
{
   "ConditionalOperator": "string",
   "ConditionExpression": "string",
   "Expected": {
      "string" : {
         "AttributeValueList": [
            {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
```

```
                "AttributeValue"
            ],
            "M": {
                "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
        }
    ],
    "ComparisonOperator": "string",
    "Exists": boolean,
    "Value": {
        "B": blob,
        "BOOL": boolean,
        "BS": [ blob ],
        "L": [
            "AttributeValue"
        ],
        "M": {
            "string" : "AttributeValue"
        },
        "N": "string",
        "NS": [ "string" ],
        "NULL": boolean,
        "S": "string",
        "SS": [ "string" ]
    }
}
},
"ExpressionAttributeNames": {
    "string" : "string"
},
"ExpressionAttributeValues": {
    "string" : {
        "B": blob,
        "BOOL": boolean,
        "BS": [ blob ],
        "L": [
            "AttributeValue"
        ],
        "M": {
```

```
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "Item": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ReturnConsumedCapacity": "string",
   "ReturnItemCollectionMetrics": "string",
   "ReturnValues": "string",
   "ReturnValuesOnConditionCheckFailure": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

Item

A map of attribute name/value pairs, one for each attribute. Only the primary key attributes are required; you can optionally provide other attribute name-value pairs for the item.

You must provide all of the attributes for the primary key. For example, with a simple primary key, you only need to provide a value for the partition key. For a composite primary key, you must provide both values for both the partition key and the sort key.

If you specify any attributes that are part of an index key, then the data types for those attributes must match those of the schema in the table's attribute definition.

Empty String and Binary attribute values are allowed. Attribute values of type String and Binary must have a length greater than zero if the attribute is used as a key attribute for a table or index.

For more information about primary keys, see Primary Key in the *Amazon DynamoDB Developer Guide*.

Each element in the `Item` map is an `AttributeValue` object.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

The name of the table to contain the item. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ConditionalOperator

This is a legacy parameter. Use `ConditionExpression` instead. For more information, see ConditionalOperator in the *Amazon DynamoDB Developer Guide*.

Type: String

Valid Values: `AND | OR`

Required: No

ConditionExpression

A condition that must be satisfied in order for a conditional `PutItem` operation to succeed.

An expression can contain any of the following:

- Functions: `attribute_exists | attribute_not_exists | attribute_type | contains | begins_with | size`

  These function names are case-sensitive.
- Comparison operators: `= | <> | < | > | <= | >= | BETWEEN | IN`
- Logical operators: `AND | OR | NOT`

For more information on condition expressions, see Condition Expressions in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

Expected

This is a legacy parameter. Use `ConditionExpression` instead. For more information, see Expected in the *Amazon DynamoDB Developer Guide*.

Type: String to ExpectedAttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression. The following are some use cases for using `ExpressionAttributeNames`:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.

- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the **#** character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for `ExpressionAttributeNames`:

- `{"#P":"Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

> **Note**
>
> Tokens that begin with the **:** character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see Specifying Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression.

Use the **:** (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify `ExpressionAttributeValues` as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"}, ":disc":
{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see Condition Expressions in the *Amazon DynamoDB Developer Guide*.

Type: String to AttributeValue object map

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

ReturnItemCollectionMetrics

Determines whether item collection metrics are returned. If set to `SIZE`, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to `NONE` (the default), no statistics are returned.

Type: String

Valid Values: `SIZE` | `NONE`

Required: No

ReturnValues

Use `ReturnValues` if you want to get the item attributes as they appeared before they were updated with the `PutItem` request. For `PutItem`, the valid values are:

- `NONE` - If `ReturnValues` is not specified, or if its value is `NONE`, then nothing is returned. (This setting is the default for `ReturnValues`.)
- `ALL_OLD` - If `PutItem` overwrote an attribute name-value pair, then the content of the old item is returned.

The values returned are strongly consistent.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

> **Note**
>
> The `ReturnValues` parameter is used by several DynamoDB operations; however, `PutItem` does not recognize any values other than `NONE` or `ALL_OLD`.

Type: String

Valid Values: `NONE` | `ALL_OLD` | `UPDATED_OLD` | `ALL_NEW` | `UPDATED_NEW`

Required: No

ReturnValuesOnConditionCheckFailure

An optional parameter that returns the item attributes for a `PutItem` operation that failed a condition check.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

Type: String

Valid Values: `ALL_OLD` | `NONE`

Required: No

Response Syntax

```
{
   "Attributes": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ConsumedCapacity": {
      "CapacityUnits": number,
      "GlobalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "LocalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "ReadCapacityUnits": number,
      "Table": {
```

```
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         },
         "TableName": "string",
         "WriteCapacityUnits": number
      },
      "ItemCollectionMetrics": {
         "ItemCollectionKey": {
            "string" : {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
                  "AttributeValue"
               ],
               "M": {
                  "string" : "AttributeValue"
               },
               "N": "string",
               "NS": [ "string" ],
               "NULL": boolean,
               "S": "string",
               "SS": [ "string" ]
            }
         },
         "SizeEstimateRangeGB": [ number ]
      }
   }
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Attributes

The attribute values as they appeared before the `PutItem` operation, but only if `ReturnValues` is specified as `ALL_OLD` in the request. Each element consists of an attribute name and an attribute value.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

ConsumedCapacity

The capacity units consumed by the `PutItem` operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. `ConsumedCapacity` is only returned if the `ReturnConsumedCapacity` parameter was specified. For more information, see Capacity unity consumption for write operations in the *Amazon DynamoDB Developer Guide*.

Type: ConsumedCapacity object

ItemCollectionMetrics

Information about item collections, if any, that were affected by the `PutItem` operation. `ItemCollectionMetrics` is only returned if the `ReturnItemCollectionMetrics` parameter was specified. If the table does not have any local secondary indexes, this information is not returned in the response.

Each `ItemCollectionMetrics` element consists of:

- `ItemCollectionKey` - The partition key value of the item collection. This is the same as the partition key value of the item itself.
- `SizeEstimateRangeGB` - An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

  The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

Type: ItemCollectionMetrics object

Errors

For information about the errors that are common to all actions, see Common Errors.

ConditionalCheckFailedException

A condition specified in the operation failed to be evaluated.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ItemCollectionSizeLimitExceededException

An item collection is too large. This exception is only returned for tables that have one or more local secondary indexes.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

ReplicatedWriteConflictException

The request was rejected because one or more items in the request are being modified by a request in another Region.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

TransactionConflictException

Operation was rejected because there is an ongoing transaction for the item.

HTTP Status Code: 400

Examples

Put an Item

The following example puts a new item into the `Thread` table, but only if there is not already an item in the table with the same key.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.PutItem

{
    "TableName": "Thread",
    "Item": {
        "LastPostDateTime": {
            "S": "201303190422"
        },
        "Tags": {
```

```
            "SS": ["Update","Multiple Items","HelpMe"]
        },
        "ForumName": {
            "S": "Amazon DynamoDB"
        },
        "Message": {
            "S": "I want to update multiple items in a single call. What's the best way
 to do that?"
        },
        "Subject": {
            "S": "How do I update multiple items?"
        },
        "LastPostedBy": {
            "S": "fred@example.com"
        }
    },
    "ConditionExpression": "ForumName <> :f and Subject <> :s",
    "ExpressionAttributeValues": {
        ":f": {"S": "Amazon DynamoDB"},
        ":s": {"S": "How do I update multiple items?"}
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

PutResourcePolicy

Service: Amazon DynamoDB

Attaches a resource-based policy document to the resource, which can be a table or stream. When you attach a resource-based policy using this API, the policy application is *eventually consistent*.

`PutResourcePolicy` is an idempotent operation; running it multiple times on the same resource using the same policy document will return the same revision ID. If you specify an `ExpectedRevisionId` that doesn't match the current policy's `RevisionId`, the `PolicyNotFoundException` will be returned.

> **Note**
>
> `PutResourcePolicy` is an asynchronous operation. If you issue a `GetResourcePolicy` request immediately after a `PutResourcePolicy` request, DynamoDB might return your previous policy, if there was one, or return the `PolicyNotFoundException`. This is because `GetResourcePolicy` uses an eventually consistent query, and the metadata for your policy or table might not be available at that moment. Wait for a few seconds, and then try the `GetResourcePolicy` request again.

Request Syntax

```
{
   "ConfirmRemoveSelfResourceAccess": boolean,
   "ExpectedRevisionId": "string",
   "Policy": "string",
   "ResourceArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

Policy

An AWS resource-based policy document in JSON format.

- The maximum size supported for a resource-based policy document is 20 KB. DynamoDB counts whitespaces when calculating the size of a policy against this limit.
- Within a resource-based policy, if the action for a DynamoDB service-linked role (SLR) to replicate data for a global table is denied, adding or deleting a replica will fail with an error.

For a full list of all considerations that apply while attaching a resource-based policy, see Resource-based policy considerations.

Type: String

Required: Yes

ResourceArn

The Amazon Resource Name (ARN) of the DynamoDB resource to which the policy will be attached. The resources you can specify include tables and streams.

You can control index permissions using the base table's policy. To specify the same permission level for your table and its indexes, you can provide both the table and index Amazon Resource Name (ARN)s in the `Resource` field of a given `Statement` in your policy document. Alternatively, to specify different permissions for your table, indexes, or both, you can define multiple `Statement` fields in your policy document.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1283.

Required: Yes

ConfirmRemoveSelfResourceAccess

Set this parameter to `true` to confirm that you want to remove your permissions to change the policy of this resource in the future.

Type: Boolean

Required: No

ExpectedRevisionId

A string value that you can use to conditionally update your policy. You can provide the revision ID of your existing policy to make mutating requests against that policy.

> **Note**
>
> When you provide an expected revision ID, if the revision ID of the existing policy on the resource doesn't match or if there's no policy attached to the resource, your request will be rejected with a `PolicyNotFoundException`.

To conditionally attach a policy when no policy exists for the resource, specify `NO_POLICY` for the revision ID.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

Response Syntax

```
{
   "RevisionId": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

RevisionId

A unique string that represents the revision ID of the policy. If you're comparing revision IDs, make sure to always use string comparison logic.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

PolicyNotFoundException

The operation tried to access a nonexistent resource-based policy.

If you specified an `ExpectedRevisionId`, it's possible that a policy is present for the resource but its revision ID didn't match the expected value.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Attach a resource-based policy to a table

The following example attaches a resource-based policy to a table named `Thread`.

To view more examples of resource-based policies, see Resource-based policy examples.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.PutResourcePolicy
{
    "ResourceArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Thread",
```

```
    "Policy": "{\"Version\":\"2012-10-17\",\"Statement\":{\"Effect\":
\"Allow\",\"Principal\":{\"AWS\":[\"arn:aws:iam::111122223333:root\",
\"arn:aws:iam::444455556666:root\"]},\"Action\":[\"dynamodb:GetItem\"],\"Resource\":
\"arn:aws:dynamodb:us-west-2:123456789012:table/Thread\"}}"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
{
    "RevisionId": "1683717331354"
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Query

Service: Amazon DynamoDB

You must provide the name of the partition key attribute and a single value for that attribute. `Query` returns all items with that partition key value. Optionally, you can provide a sort key attribute and use a comparison operator to refine the search results.

Use the `KeyConditionExpression` parameter to provide a specific value for the partition key. The `Query` operation will return all of the items from the table or index with that partition key value. You can optionally narrow the scope of the `Query` operation by specifying a sort key value and a comparison operator in `KeyConditionExpression`. To further refine the `Query` results, you can optionally provide a `FilterExpression`. A `FilterExpression` determines which items within the results should be returned to you. All of the other results are discarded.

A `Query` operation always returns a result set. If no matching items are found, the result set will be empty. Queries that do not return results consume the minimum number of read capacity units for that type of read operation.

> **Note**
>
> DynamoDB calculates the number of read capacity units consumed based on item size, not on the amount of data that is returned to an application. The number of capacity units consumed will be the same whether you request all of the attributes (the default behavior) or just some of them (using a projection expression). The number will also be the same whether or not you use a `FilterExpression`.

`Query` results are always sorted by the sort key value. If the data type of the sort key is Number, the results are returned in numeric order; otherwise, the results are returned in order of UTF-8 bytes. By default, the sort order is ascending. To reverse the order, set the `ScanIndexForward` parameter to false.

A single `Query` operation will read up to the maximum number of items set (if using the `Limit` parameter) or a maximum of 1 MB of data and then apply any filtering to the results using `FilterExpression`. If `LastEvaluatedKey` is present in the response, you will need to paginate the result set. For more information, see Paginating the Results in the *Amazon DynamoDB Developer Guide*.

`FilterExpression` is applied after a `Query` finishes, but before the results are returned. A `FilterExpression` cannot contain partition key or sort key attributes. You need to specify those attributes in the `KeyConditionExpression`.

> **Note**
>
> A `Query` operation can return an empty result set and a `LastEvaluatedKey` if all the items read for the page of results are filtered out.

You can query a table, a local secondary index, or a global secondary index. For a query on a table or on a local secondary index, you can set the `ConsistentRead` parameter to `true` and obtain a strongly consistent result. Global secondary indexes support eventually consistent reads only, so do not specify `ConsistentRead` when querying a global secondary index.

For additional information on how to use the Query API, please see Working with Queries.

Request Syntax

```
{
   "AttributesToGet": [ "string" ],
   "ConditionalOperator": "string",
   "ConsistentRead": boolean,
   "ExclusiveStartKey": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ExpressionAttributeNames": {
```

```
      "string" : "string"
   },
   "ExpressionAttributeValues": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "FilterExpression": "string",
   "IndexName": "string",
   "KeyConditionExpression": "string",
   "KeyConditions": {
      "string" : {
         "AttributeValueList": [
            {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
                  "AttributeValue"
               ],
               "M": {
                  "string" : "AttributeValue"
               },
               "N": "string",
               "NS": [ "string" ],
               "NULL": boolean,
               "S": "string",
               "SS": [ "string" ]
            }
         ],
         "ComparisonOperator": "string"
```

```
      }
   },
   "Limit": number,
   "ProjectionExpression": "string",
   "QueryFilter": {
      "string" : {
         "AttributeValueList": [
            {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
                  "AttributeValue"
               ],
               "M": {
                  "string" : "AttributeValue"
               },
               "N": "string",
               "NS": [ "string" ],
               "NULL": boolean,
               "S": "string",
               "SS": [ "string" ]
            }
         ],
         "ComparisonOperator": "string"
      }
   },
   "ReturnConsumedCapacity": "string",
   "ScanIndexForward": boolean,
   "Select": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table containing the requested items. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

AttributesToGet

This is a legacy parameter. Use `ProjectionExpression` instead. For more information, see AttributesToGet in the *Amazon DynamoDB Developer Guide*.

Type: Array of strings

Array Members: Minimum number of 1 item.

Length Constraints: Maximum length of 65535.

Required: No

ConditionalOperator

This is a legacy parameter. Use `FilterExpression` instead. For more information, see ConditionalOperator in the *Amazon DynamoDB Developer Guide*.

Type: String

Valid Values: `AND` | `OR`

Required: No

ConsistentRead

Determines the read consistency model: If set to `true`, then the operation uses strongly consistent reads; otherwise, the operation uses eventually consistent reads.

Strongly consistent reads are not supported on global secondary indexes. If you query a global secondary index with `ConsistentRead` set to `true`, you will receive a `ValidationException`.

Type: Boolean

Required: No

ExclusiveStartKey

The primary key of the first item that this operation will evaluate. Use the value that was returned for `LastEvaluatedKey` in the previous operation.

The data type for `ExclusiveStartKey` must be String, Number, or Binary. No set data types are allowed.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression. The following are some use cases for using `ExpressionAttributeNames`:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the **#** character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for `ExpressionAttributeNames`:

- `{"#P":"Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

> **Note**
>
> Tokens that begin with the **:** character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see Specifying Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression.

Use the **:** (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the *ProductStatus* attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify `ExpressionAttributeValues` as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"}, ":disc":
{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see Specifying Conditions in the *Amazon DynamoDB Developer Guide*.

Type: String to AttributeValue object map

Required: No

FilterExpression

A string that contains conditions that DynamoDB applies after the `Query` operation, but before the data is returned to you. Items that do not satisfy the `FilterExpression` criteria are not returned.

A `FilterExpression` does not allow key attributes. You cannot define a filter expression based on a partition key or a sort key.

> **Note**
>
> A `FilterExpression` is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

For more information, see Filter Expressions in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

IndexName

The name of an index to query. This index can be any local secondary index or global secondary index on the table. Note that if you use the `IndexName` parameter, you must also provide `TableName`.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

KeyConditionExpression

The condition that specifies the key values for items to be retrieved by the `Query` action.

The condition must perform an equality test on a single partition key value.

The condition can optionally perform one of several comparison tests on a single sort key value. This allows `Query` to retrieve one item with a given partition key value and sort key value, or several items that have the same partition key value but different sort key values.

The partition key equality test is required, and must be specified in the following format:

`partitionKeyName = :partitionkeyval`

If you also want to provide a condition for the sort key, it must be combined using `AND` with the condition for the sort key. Following is an example, using the **=** comparison operator for the sort key:

`partitionKeyName = :partitionkeyval AND sortKeyName = :sortkeyval`

Valid comparisons for the sort key condition are as follows:

- `sortKeyName = :sortkeyval` - true if the sort key value is equal to `:sortkeyval`.
- `sortKeyName < :sortkeyval` - true if the sort key value is less than `:sortkeyval`.
- `sortKeyName <= :sortkeyval` - true if the sort key value is less than or equal to `:sortkeyval`.
- `sortKeyName > :sortkeyval` - true if the sort key value is greater than `:sortkeyval`.
- `sortKeyName >= :sortkeyval` - true if the sort key value is greater than or equal to `:sortkeyval`.
- `sortKeyName BETWEEN :sortkeyval1 AND :sortkeyval2` - true if the sort key value is greater than or equal to `:sortkeyval1`, and less than or equal to `:sortkeyval2`.
- `begins_with ( sortKeyName, :sortkeyval )` - true if the sort key value begins with a particular operand. (You cannot use this function with a sort key that is of type Number.) Note that the function name `begins_with` is case-sensitive.

Use the `ExpressionAttributeValues` parameter to replace tokens such as `:partitionval` and `:sortval` with actual values at runtime.

You can optionally use the `ExpressionAttributeNames` parameter to replace the names of the partition key and sort key with placeholder tokens. This option might be necessary if an attribute name conflicts with a DynamoDB reserved word. For example, the following `KeyConditionExpression` parameter causes an error because *Size* is a reserved word:

- `Size = :myval`

To work around this, define a placeholder (such a `#S`) to represent the attribute name *Size*. `KeyConditionExpression` then is as follows:

- `#S = :myval`

For a list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*.

For more information on `ExpressionAttributeNames` and `ExpressionAttributeValues`, see Using Placeholders for Attribute Names and Values in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

KeyConditions

This is a legacy parameter. Use `KeyConditionExpression` instead. For more information, see KeyConditions in the *Amazon DynamoDB Developer Guide*.

Type: String to Condition object map

Key Length Constraints: Maximum length of 65535.

Required: No

Limit

The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, and a key in `LastEvaluatedKey` to apply in a subsequent operation, so that you can pick up where you left off. Also, if the processed dataset size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in `LastEvaluatedKey` to apply in a subsequent operation to continue the operation. For more information, see Query and Scan in the *Amazon DynamoDB Developer Guide*.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

ProjectionExpression

A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see Accessing Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

QueryFilter

This is a legacy parameter. Use `FilterExpression` instead. For more information, see QueryFilter in the *Amazon DynamoDB Developer Guide*.

Type: String to Condition object map

Key Length Constraints: Maximum length of 65535.

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

ScanIndexForward

Specifies the order for index traversal: If `true` (default), the traversal is performed in ascending order; if `false`, the traversal is performed in descending order.

Items with the same partition key value are stored in sorted order by sort key. If the sort key data type is Number, the results are stored in numeric order. For type String, the results are stored in order of UTF-8 bytes. For type Binary, DynamoDB treats each byte of the binary data as unsigned.

If `ScanIndexForward` is `true`, DynamoDB returns the results in the order in which they are stored (by sort key value). This is the default behavior. If `ScanIndexForward` is `false`,

DynamoDB reads the results in reverse order by sort key value, and then returns the results to the client.

Type: Boolean

Required: No

Select

The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, the count of matching items, or in the case of an index, some or all of the attributes projected into the index.

- `ALL_ATTRIBUTES` - Returns all of the item attributes from the specified table or index. If you query a local secondary index, then for each matching item in the index, DynamoDB fetches the entire item from the parent table. If the index is configured to project all item attributes, then all of the data can be obtained from the local secondary index, and no fetching is required.
- `ALL_PROJECTED_ATTRIBUTES` - Allowed only when querying an index. Retrieves all attributes that have been projected into the index. If the index is configured to project all attributes, this return value is equivalent to specifying `ALL_ATTRIBUTES`.
- `COUNT` - Returns the number of matching items, rather than the matching items themselves. Note that this uses the same quantity of read capacity units as getting the items, and is subject to the same item size calculations.
- `SPECIFIC_ATTRIBUTES` - Returns only the attributes listed in `ProjectionExpression`. This return value is equivalent to specifying `ProjectionExpression` without specifying any value for `Select`.

  If you query or scan a local secondary index and request only attributes that are projected into that index, the operation will read only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB fetches each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency.

  If you query or scan a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

If neither `Select` nor `ProjectionExpression` are specified, DynamoDB defaults to `ALL_ATTRIBUTES` when accessing a table, and `ALL_PROJECTED_ATTRIBUTES` when accessing

an index. You cannot use both `Select` and `ProjectionExpression` together in a single request, unless the value for `Select` is `SPECIFIC_ATTRIBUTES`. (This usage is equivalent to specifying `ProjectionExpression` without any value for `Select`.)

> **Note**
>
> If you use the `ProjectionExpression` parameter, then the value for `Select` can only be `SPECIFIC_ATTRIBUTES`. Any other value for `Select` will return an error.

Type: String

Valid Values: `ALL_ATTRIBUTES | ALL_PROJECTED_ATTRIBUTES | SPECIFIC_ATTRIBUTES | COUNT`

Required: No

Response Syntax

```
{
   "ConsumedCapacity": {
      "CapacityUnits": number,
      "GlobalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "LocalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "ReadCapacityUnits": number,
      "Table": {
         "CapacityUnits": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
```

```
        },
        "TableName": "string",
        "WriteCapacityUnits": number
    },
    "Count": number,
    "Items": [
        {
            "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
                "L": [
                    "AttributeValue"
                ],
                "M": {
                    "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
            }
        }
    ],
    "LastEvaluatedKey": {
        "string" : {
            "B": blob,
            "BOOL": boolean,
            "BS": [ blob ],
            "L": [
                "AttributeValue"
            ],
            "M": {
                "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
        }
    },
    "ScannedCount": number
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The capacity units consumed by the `Query` operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. `ConsumedCapacity` is only returned if the `ReturnConsumedCapacity` parameter was specified. For more information, see Capacity unit consumption for read operations in the *Amazon DynamoDB Developer Guide*.

Type: ConsumedCapacity object

Count

The number of items in the response.

If you used a `QueryFilter` in the request, then `Count` is the number of items returned after the filter was applied, and `ScannedCount` is the number of matching items before the filter was applied.

If you did not use a filter in the request, then `Count` and `ScannedCount` are the same.

Type: Integer

Items

An array of item attributes that match the query criteria. Each element in this array consists of an attribute name and the value for that attribute.

Type: Array of string to AttributeValue object maps

Key Length Constraints: Maximum length of 65535.

LastEvaluatedKey

The primary key of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If `LastEvaluatedKey` is empty, then the "last page" of results has been processed and there is no more data to be retrieved.

If `LastEvaluatedKey` is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when `LastEvaluatedKey` is empty.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

ScannedCount

The number of items evaluated, before any `QueryFilter` is applied. A high `ScannedCount` value with few, or no, `Count` results indicates an inefficient `Query` operation. For more information, see Count and ScannedCount in the *Amazon DynamoDB Developer Guide*.

If you did not use a filter in the request, then `ScannedCount` is the same as `Count`.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

Examples

Retrieve a Range of Items

The following example queries the `Reply` table for replies in a forum that were posted by particular users. There is a local secondary index on the *Reply* table, `PostedBy-Index`, to facilitate fast lookups on these attributes.

The `ProjectionExpression` parameter determines which attributes are returned.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
```

```
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.Query

{
    "TableName": "Reply",
    "IndexName": "PostedBy-Index",
    "Limit": 3,
    "ConsistentRead": true,
    "ProjectionExpression": "Id, PostedBy, ReplyDateTime",
    "KeyConditionExpression": "Id = :v1 AND PostedBy BETWEEN :v2a AND :v2b",
    "ExpressionAttributeValues": {
        ":v1": {"S": "Amazon DynamoDB#DynamoDB Thread 1"},
        ":v2a": {"S": "User A"},
        ":v2b": {"S": "User C"}
    },
    "ReturnConsumedCapacity": "TOTAL"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "ConsumedCapacity": {
        "CapacityUnits": 1,
        "TableName": "Reply"
    },
    "Count": 2,
    "Items": [
        {
            "ReplyDateTime": {"S": "2015-02-18T20:27:36.165Z"},
            "PostedBy": {"S": "User A"},
            "Id": {"S": "Amazon DynamoDB#DynamoDB Thread 1"}
        },
        {
            "ReplyDateTime": {"S": "2015-02-25T20:27:36.165Z"},
            "PostedBy": {"S": "User B"},
            "Id": {"S": "Amazon DynamoDB#DynamoDB Thread 1"}
        }
```

```
    ],
    "ScannedCount": 2
}
```

Count Items

The following example returns the number of items in the `Thread` table for a particular forum.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.Query

{
    "TableName": "Thread",
    "ConsistentRead": true,
    "KeyConditionExpression": "ForumName = :val",
    "ExpressionAttributeValues": {":val": {"S": "Amazon DynamoDB"}}
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
          "Count":2,
          "ScannedCount":2
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

RestoreTableFromBackup

Service: Amazon DynamoDB

Creates a new table from an existing backup. Any number of users can execute up to 50 concurrent restores (any type of restore) in a given account.

You can call `RestoreTableFromBackup` at a maximum rate of 10 times per second.

You must manually set up the following on the restored table:

- Auto scaling policies
- IAM policies
- Amazon CloudWatch metrics and alarms
- Tags
- Stream settings
- Time to Live (TTL) settings

Request Syntax

```
{
   "BackupArn": "string",
   "BillingModeOverride": "string",
   "GlobalSecondaryIndexOverride": [
      {
         "IndexName": "string",
         "KeySchema": [
            {
               "AttributeName": "string",
               "KeyType": "string"
            }
         ],
         "OnDemandThroughput": {
            "MaxReadRequestUnits": number,
            "MaxWriteRequestUnits": number
         },
         "Projection": {
            "NonKeyAttributes": [ "string" ],
            "ProjectionType": "string"
         },
         "ProvisionedThroughput": {
```

```
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         },
         "WarmThroughput": {
            "ReadUnitsPerSecond": number,
            "WriteUnitsPerSecond": number
         }
      }
   ],
   "LocalSecondaryIndexOverride": [
      {
         "IndexName": "string",
         "KeySchema": [
            {
               "AttributeName": "string",
               "KeyType": "string"
            }
         ],
         "Projection": {
            "NonKeyAttributes": [ "string" ],
            "ProjectionType": "string"
         }
      }
   ],
   "OnDemandThroughputOverride": {
      "MaxReadRequestUnits": number,
      "MaxWriteRequestUnits": number
   },
   "ProvisionedThroughputOverride": {
      "ReadCapacityUnits": number,
      "WriteCapacityUnits": number
   },
   "SSESpecificationOverride": {
      "Enabled": boolean,
      "KMSMasterKeyId": "string",
      "SSEType": "string"
   },
   "TargetTableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

BackupArn

The Amazon Resource Name (ARN) associated with the backup.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

TargetTableName

The name of the new table to which the backup must be restored.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

BillingModeOverride

The billing mode of the restored table.

Type: String

Valid Values: `PROVISIONED | PAY_PER_REQUEST`

Required: No

GlobalSecondaryIndexOverride

List of global secondary indexes for the restored table. The indexes provided should match existing secondary indexes. You can choose to exclude some or all of the indexes at the time of restore.

Type: Array of GlobalSecondaryIndex objects

Required: No

LocalSecondaryIndexOverride

List of local secondary indexes for the restored table. The indexes provided should match existing secondary indexes. You can choose to exclude some or all of the indexes at the time of restore.

Type: Array of LocalSecondaryIndex objects

Required: No

OnDemandThroughputOverride

Sets the maximum number of read and write units for the specified on-demand table. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughputOverride

Provisioned throughput settings for the restored table.

Type: ProvisionedThroughput object

Required: No

SSESpecificationOverride

The new server-side encryption settings for the restored table.

Type: SSESpecification object

Required: No

Response Syntax

```
{
   "TableDescription": {
      "ArchivalSummary": {
         "ArchivalBackupArn": "string",
         "ArchivalDateTime": number,
         "ArchivalReason": "string"
      },
      "AttributeDefinitions": [
         {
```

```
            "AttributeName": "string",
            "AttributeType": "string"
         }
      ],
      "BillingModeSummary": {
         "BillingMode": "string",
         "LastUpdateToPayPerRequestDateTime": number
      },
      "CreationDateTime": number,
      "DeletionProtectionEnabled": boolean,
      "GlobalSecondaryIndexes": [
         {
            "Backfilling": boolean,
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "IndexStatus": "string",
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "OnDemandThroughput": {
               "MaxReadRequestUnits": number,
               "MaxWriteRequestUnits": number
            },
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            },
            "ProvisionedThroughput": {
               "LastDecreaseDateTime": number,
               "LastIncreaseDateTime": number,
               "NumberOfDecreasesToday": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
```

```
            }
        ],
        "GlobalTableVersion": "string",
        "GlobalTableWitnesses": [
            {
                "RegionName": "string",
                "WitnessStatus": "string"
            }
        ],
        "ItemCount": number,
        "KeySchema": [
            {
                "AttributeName": "string",
                "KeyType": "string"
            }
        ],
        "LatestStreamArn": "string",
        "LatestStreamLabel": "string",
        "LocalSecondaryIndexes": [
            {
                "IndexArn": "string",
                "IndexName": "string",
                "IndexSizeBytes": number,
                "ItemCount": number,
                "KeySchema": [
                    {
                        "AttributeName": "string",
                        "KeyType": "string"
                    }
                ],
                "Projection": {
                    "NonKeyAttributes": [ "string" ],
                    "ProjectionType": "string"
                }
            }
        ],
        "MultiRegionConsistency": "string",
        "OnDemandThroughput": {
            "MaxReadRequestUnits": number,
            "MaxWriteRequestUnits": number
        },
        "ProvisionedThroughput": {
            "LastDecreaseDateTime": number,
            "LastIncreaseDateTime": number,
```

```
            "NumberOfDecreasesToday": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
        },
        "Replicas": [
            {
                "GlobalSecondaryIndexes": [
                    {
                        "IndexName": "string",
                        "OnDemandThroughputOverride": {
                            "MaxReadRequestUnits": number
                        },
                        "ProvisionedThroughputOverride": {
                            "ReadCapacityUnits": number
                        },
                        "WarmThroughput": {
                            "ReadUnitsPerSecond": number,
                            "Status": "string",
                            "WriteUnitsPerSecond": number
                        }
                    }
                ],
                "KMSMasterKeyId": "string",
                "OnDemandThroughputOverride": {
                    "MaxReadRequestUnits": number
                },
                "ProvisionedThroughputOverride": {
                    "ReadCapacityUnits": number
                },
                "RegionName": "string",
                "ReplicaInaccessibleDateTime": number,
                "ReplicaStatus": "string",
                "ReplicaStatusDescription": "string",
                "ReplicaStatusPercentProgress": "string",
                "ReplicaTableClassSummary": {
                    "LastUpdateDateTime": number,
                    "TableClass": "string"
                },
                "WarmThroughput": {
                    "ReadUnitsPerSecond": number,
                    "Status": "string",
                    "WriteUnitsPerSecond": number
                }
            }
```

```
        ],
        "RestoreSummary": {
           "RestoreDateTime": number,
           "RestoreInProgress": boolean,
           "SourceBackupArn": "string",
           "SourceTableArn": "string"
        },
        "SSEDescription": {
           "InaccessibleEncryptionDateTime": number,
           "KMSMasterKeyArn": "string",
           "SSEType": "string",
           "Status": "string"
        },
        "StreamSpecification": {
           "StreamEnabled": boolean,
           "StreamViewType": "string"
        },
        "TableArn": "string",
        "TableClassSummary": {
           "LastUpdateDateTime": number,
           "TableClass": "string"
        },
        "TableId": "string",
        "TableName": "string",
        "TableSizeBytes": number,
        "TableStatus": "string",
        "WarmThroughput": {
           "ReadUnitsPerSecond": number,
           "Status": "string",
           "WriteUnitsPerSecond": number
        }
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TableDescription

The description of the table created from an existing backup.

Type: TableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

BackupInUseException

There is another ongoing conflicting backup control plane operation on the table. The backup is either being created, deleted or restored to a table.

HTTP Status Code: 400

BackupNotFoundException

Backup not found for the given BackupARN.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

TableAlreadyExistsException

A target table with the specified name already exists.

HTTP Status Code: 400

TableInUseException

A target table with the specified name is either being created or deleted.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

RestoreTableToPointInTime

Service: Amazon DynamoDB

Restores the specified table to the specified point in time within `EarliestRestorableDateTime` and `LatestRestorableDateTime`. You can restore your table to any point in time in the last 35 days. You can set the recovery period to any value between 1 and 35 days. Any number of users can execute up to 50 concurrent restores (any type of restore) in a given account.

When you restore using point in time recovery, DynamoDB restores your table data to the state based on the selected date and time (day:hour:minute:second) to a new table.

Along with data, the following are also included on the new restored table using point in time recovery:

- Global secondary indexes (GSIs)
- Local secondary indexes (LSIs)
- Provisioned read and write capacity
- Encryption settings

> **Important**
>
> All these settings come from the current settings of the source table at the time of restore.

You must manually set up the following on the restored table:

- Auto scaling policies
- IAM policies
- Amazon CloudWatch metrics and alarms
- Tags
- Stream settings
- Time to Live (TTL) settings
- Point in time recovery settings

Request Syntax

```
{
   "BillingModeOverride": "string",
   "GlobalSecondaryIndexOverride": [
      {
         "IndexName": "string",
         "KeySchema": [
            {
               "AttributeName": "string",
               "KeyType": "string"
            }
         ],
         "OnDemandThroughput": {
            "MaxReadRequestUnits": number,
            "MaxWriteRequestUnits": number
         },
         "Projection": {
            "NonKeyAttributes": [ "string" ],
            "ProjectionType": "string"
         },
         "ProvisionedThroughput": {
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         },
         "WarmThroughput": {
            "ReadUnitsPerSecond": number,
            "WriteUnitsPerSecond": number
         }
      }
   ],
   "LocalSecondaryIndexOverride": [
      {
         "IndexName": "string",
         "KeySchema": [
            {
               "AttributeName": "string",
               "KeyType": "string"
            }
         ],
         "Projection": {
            "NonKeyAttributes": [ "string" ],
            "ProjectionType": "string"
```

```
            }
        }
    ],
    "OnDemandThroughputOverride": {
        "MaxReadRequestUnits": number,
        "MaxWriteRequestUnits": number
    },
    "ProvisionedThroughputOverride": {
        "ReadCapacityUnits": number,
        "WriteCapacityUnits": number
    },
    "RestoreDateTime": number,
    "SourceTableArn": "string",
    "SourceTableName": "string",
    "SSESpecificationOverride": {
        "Enabled": boolean,
        "KMSMasterKeyId": "string",
        "SSEType": "string"
    },
    "TargetTableName": "string",
    "UseLatestRestorableTime": boolean
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TargetTableName

The name of the new table to which it must be restored to.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

BillingModeOverride

The billing mode of the restored table.

Type: String

Valid Values: `PROVISIONED | PAY_PER_REQUEST`

Required: No

GlobalSecondaryIndexOverride

List of global secondary indexes for the restored table. The indexes provided should match existing secondary indexes. You can choose to exclude some or all of the indexes at the time of restore.

Type: Array of GlobalSecondaryIndex objects

Required: No

LocalSecondaryIndexOverride

List of local secondary indexes for the restored table. The indexes provided should match existing secondary indexes. You can choose to exclude some or all of the indexes at the time of restore.

Type: Array of LocalSecondaryIndex objects

Required: No

OnDemandThroughputOverride

Sets the maximum number of read and write units for the specified on-demand table. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughputOverride

Provisioned throughput settings for the restored table.

Type: ProvisionedThroughput object

Required: No

RestoreDateTime

Time in the past to restore the table to.

Type: Timestamp

Required: No

SourceTableArn

The DynamoDB table that will be restored. This value is an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

SourceTableName

Name of the source table that is being restored.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

SSESpecificationOverride

The new server-side encryption settings for the restored table.

Type: SSESpecification object

Required: No

UseLatestRestorableTime

Restore the table to the latest possible time. `LatestRestorableDateTime` is typically 5 minutes before the current time.

Type: Boolean

Required: No

Response Syntax

```
{
   "TableDescription": {
      "ArchivalSummary": {
         "ArchivalBackupArn": "string",
         "ArchivalDateTime": number,
         "ArchivalReason": "string"
      },
      "AttributeDefinitions": [
         {
            "AttributeName": "string",
            "AttributeType": "string"
         }
      ],
      "BillingModeSummary": {
         "BillingMode": "string",
         "LastUpdateToPayPerRequestDateTime": number
      },
      "CreationDateTime": number,
      "DeletionProtectionEnabled": boolean,
      "GlobalSecondaryIndexes": [
         {
            "Backfilling": boolean,
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "IndexStatus": "string",
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "OnDemandThroughput": {
               "MaxReadRequestUnits": number,
               "MaxWriteRequestUnits": number
            },
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            },
```

```
            "ProvisionedThroughput": {
               "LastDecreaseDateTime": number,
               "LastIncreaseDateTime": number,
               "NumberOfDecreasesToday": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "GlobalTableVersion": "string",
      "GlobalTableWitnesses": [
         {
            "RegionName": "string",
            "WitnessStatus": "string"
         }
      ],
      "ItemCount": number,
      "KeySchema": [
         {
            "AttributeName": "string",
            "KeyType": "string"
         }
      ],
      "LatestStreamArn": "string",
      "LatestStreamLabel": "string",
      "LocalSecondaryIndexes": [
         {
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "Projection": {
               "NonKeyAttributes": [ "string" ],
```

```
                "ProjectionType": "string"
            }
        }
    ],
    "MultiRegionConsistency": "string",
    "OnDemandThroughput": {
        "MaxReadRequestUnits": number,
        "MaxWriteRequestUnits": number
    },
    "ProvisionedThroughput": {
        "LastDecreaseDateTime": number,
        "LastIncreaseDateTime": number,
        "NumberOfDecreasesToday": number,
        "ReadCapacityUnits": number,
        "WriteCapacityUnits": number
    },
    "Replicas": [
        {
            "GlobalSecondaryIndexes": [
                {
                    "IndexName": "string",
                    "OnDemandThroughputOverride": {
                        "MaxReadRequestUnits": number
                    },
                    "ProvisionedThroughputOverride": {
                        "ReadCapacityUnits": number
                    },
                    "WarmThroughput": {
                        "ReadUnitsPerSecond": number,
                        "Status": "string",
                        "WriteUnitsPerSecond": number
                    }
                }
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
                "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
                "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "ReplicaInaccessibleDateTime": number,
            "ReplicaStatus": "string",
```

```
            "ReplicaStatusDescription": "string",
            "ReplicaStatusPercentProgress": "string",
            "ReplicaTableClassSummary": {
               "LastUpdateDateTime": number,
               "TableClass": "string"
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "RestoreSummary": {
         "RestoreDateTime": number,
         "RestoreInProgress": boolean,
         "SourceBackupArn": "string",
         "SourceTableArn": "string"
      },
      "SSEDescription": {
         "InaccessibleEncryptionDateTime": number,
         "KMSMasterKeyArn": "string",
         "SSEType": "string",
         "Status": "string"
      },
      "StreamSpecification": {
         "StreamEnabled": boolean,
         "StreamViewType": "string"
      },
      "TableArn": "string",
      "TableClassSummary": {
         "LastUpdateDateTime": number,
         "TableClass": "string"
      },
      "TableId": "string",
      "TableName": "string",
      "TableSizeBytes": number,
      "TableStatus": "string",
      "WarmThroughput": {
         "ReadUnitsPerSecond": number,
         "Status": "string",
         "WriteUnitsPerSecond": number
      }
   }
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TableDescription

Represents the properties of a table.

Type: TableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

InvalidRestoreTimeException

An invalid restore time was specified. RestoreDateTime must be between EarliestRestorableDateTime and LatestRestorableDateTime.

HTTP Status Code: 400

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

PointInTimeRecoveryUnavailableException

Point in time recovery has not yet been enabled for this source table.

HTTP Status Code: 400

TableAlreadyExistsException

A target table with the specified name already exists.

HTTP Status Code: 400

TableInUseException

A target table with the specified name is either being created or deleted.

HTTP Status Code: 400

TableNotFoundException

A source table with the name `TableName` does not currently exist within the subscriber's account or the subscriber is operating in the wrong AWS Region.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Scan

Service: Amazon DynamoDB

The `Scan` operation returns one or more items and item attributes by accessing every item in a table or a secondary index. To have DynamoDB return fewer items, you can provide a `FilterExpression` operation.

If the total size of scanned items exceeds the maximum dataset size limit of 1 MB, the scan completes and results are returned to the user. The `LastEvaluatedKey` value is also returned and the requestor can use the `LastEvaluatedKey` to continue the scan in a subsequent operation. Each scan response also includes number of items that were scanned (ScannedCount) as part of the request. If using a `FilterExpression`, a scan result can result in no items meeting the criteria and the `Count` will result in zero. If you did not use a `FilterExpression` in the scan request, then `Count` is the same as `ScannedCount`.

> **Note**
>
> `Count` and `ScannedCount` only return the count of items specific to a single scan request and, unless the table is less than 1MB, do not represent the total number of items in the table.

A single `Scan` operation first reads up to the maximum number of items set (if using the `Limit` parameter) or a maximum of 1 MB of data and then applies any filtering to the results if a `FilterExpression` is provided. If `LastEvaluatedKey` is present in the response, pagination is required to complete the full table scan. For more information, see Paginating the Results in the *Amazon DynamoDB Developer Guide*.

`Scan` operations proceed sequentially; however, for faster performance on a large table or secondary index, applications can request a parallel `Scan` operation by providing the `Segment` and `TotalSegments` parameters. For more information, see Parallel Scan in the *Amazon DynamoDB Developer Guide*.

By default, a `Scan` uses eventually consistent reads when accessing the items in a table. Therefore, the results from an eventually consistent `Scan` may not include the latest item changes at the time the scan iterates through each item in the table. If you require a strongly consistent read of each item as the scan iterates through the items in the table, you can set the `ConsistentRead` parameter to true. Strong consistency only relates to the consistency of the read at the item level.

> **Note**
>
> DynamoDB does not provide snapshot isolation for a scan operation when the `ConsistentRead` parameter is set to true. Thus, a DynamoDB scan operation does not guarantee that all reads in a scan see a consistent snapshot of the table when the scan operation was requested.

Request Syntax

```
{
   "AttributesToGet": [ "string" ],
   "ConditionalOperator": "string",
   "ConsistentRead": boolean,
   "ExclusiveStartKey": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ExpressionAttributeNames": {
      "string" : "string"
   },
   "ExpressionAttributeValues": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
```

```
            ],
            "M": {
               "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
         }
      },
      "FilterExpression": "string",
      "IndexName": "string",
      "Limit": number,
      "ProjectionExpression": "string",
      "ReturnConsumedCapacity": "string",
      "ScanFilter": {
         "string" : {
            "AttributeValueList": [
               {
                  "B": blob,
                  "BOOL": boolean,
                  "BS": [ blob ],
                  "L": [
                     "AttributeValue"
                  ],
                  "M": {
                     "string" : "AttributeValue"
                  },
                  "N": "string",
                  "NS": [ "string" ],
                  "NULL": boolean,
                  "S": "string",
                  "SS": [ "string" ]
               }
            ],
            "ComparisonOperator": "string"
         }
      },
      "Segment": number,
      "Select": "string",
      "TableName": "string",
      "TotalSegments": number
```

```
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table containing the requested items or if you provide `IndexName`, the name of the table to which that index belongs.

You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

AttributesToGet

This is a legacy parameter. Use `ProjectionExpression` instead. For more information, see AttributesToGet in the *Amazon DynamoDB Developer Guide*.

Type: Array of strings

Array Members: Minimum number of 1 item.

Length Constraints: Maximum length of 65535.

Required: No

ConditionalOperator

This is a legacy parameter. Use `FilterExpression` instead. For more information, see ConditionalOperator in the *Amazon DynamoDB Developer Guide*.

Type: String

Valid Values: AND | OR

Required: No

ConsistentRead

A Boolean value that determines the read consistency model during the scan:

- If ConsistentRead is false, then the data returned from Scan might not contain the results from other recently completed write operations (PutItem, UpdateItem, or DeleteItem).
- If ConsistentRead is true, then all of the write operations that completed before the Scan began are guaranteed to be contained in the Scan response.

The default setting for ConsistentRead is false.

The ConsistentRead parameter is not supported on global secondary indexes. If you scan a global secondary index with ConsistentRead set to true, you will receive a ValidationException.

Type: Boolean

Required: No

ExclusiveStartKey

The primary key of the first item that this operation will evaluate. Use the value that was returned for LastEvaluatedKey in the previous operation.

The data type for ExclusiveStartKey must be String, Number or Binary. No set data types are allowed.

In a parallel scan, a Scan request that includes ExclusiveStartKey must specify the same segment whose previous Scan returned the corresponding value of LastEvaluatedKey.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression. The following are some use cases for using ExpressionAttributeNames:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the **#** character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for `ExpressionAttributeNames`:

- `{"#P":"Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

> **Note**
>
> Tokens that begin with the **:** character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see Specifying Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression.

Use the **:** (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the `ProductStatus` attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify `ExpressionAttributeValues` as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"}, ":disc":
{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see Condition Expressions in the *Amazon DynamoDB Developer Guide*.

Type: String to AttributeValue object map

Required: No

FilterExpression

A string that contains conditions that DynamoDB applies after the `Scan` operation, but before the data is returned to you. Items that do not satisfy the `FilterExpression` criteria are not returned.

> **Note**
>
> A `FilterExpression` is applied after the items have already been read; the process of filtering does not consume any additional read capacity units.

For more information, see Filter Expressions in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

IndexName

The name of a secondary index to scan. This index can be any local secondary index or global secondary index. Note that if you use the `IndexName` parameter, you must also provide `TableName`.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Limit

The maximum number of items to evaluate (not necessarily the number of matching items). If DynamoDB processes the number of items up to the limit while processing the results, it stops the operation and returns the matching values up to that point, and a key in `LastEvaluatedKey` to apply in a subsequent operation, so that you can pick up where you left off. Also, if the processed dataset size exceeds 1 MB before DynamoDB reaches this limit, it stops the operation and returns the matching values up to the limit, and a key in `LastEvaluatedKey` to apply in a subsequent operation to continue the operation. For more information, see Working with Queries in the *Amazon DynamoDB Developer Guide*.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

ProjectionExpression

A string that identifies one or more attributes to retrieve from the specified table or index. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the expression must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see Specifying Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

ScanFilter

This is a legacy parameter. Use `FilterExpression` instead. For more information, see ScanFilter in the *Amazon DynamoDB Developer Guide*.

Type: String to Condition object map

Key Length Constraints: Maximum length of 65535.

Required: No

Segment

For a parallel `Scan` request, `Segment` identifies an individual segment to be scanned by an application worker.

Segment IDs are zero-based, so the first segment is always 0. For example, if you want to use four application threads to scan a table or an index, then the first thread specifies a `Segment` value of 0, the second thread specifies 1, and so on.

The value of `LastEvaluatedKey` returned from a parallel `Scan` request must be used as `ExclusiveStartKey` with the same segment ID in a subsequent `Scan` operation.

The value for `Segment` must be greater than or equal to 0, and less than the value provided for `TotalSegments`.

If you provide `Segment`, you must also provide `TotalSegments`.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 999999.

Required: No

Select

The attributes to be returned in the result. You can retrieve all item attributes, specific item attributes, the count of matching items, or in the case of an index, some or all of the attributes projected into the index.

- `ALL_ATTRIBUTES` - Returns all of the item attributes from the specified table or index. If you query a local secondary index, then for each matching item in the index, DynamoDB fetches the entire item from the parent table. If the index is configured to project all item attributes, then all of the data can be obtained from the local secondary index, and no fetching is required.
- `ALL_PROJECTED_ATTRIBUTES` - Allowed only when querying an index. Retrieves all attributes that have been projected into the index. If the index is configured to project all attributes, this return value is equivalent to specifying `ALL_ATTRIBUTES`.
- `COUNT` - Returns the number of matching items, rather than the matching items themselves. Note that this uses the same quantity of read capacity units as getting the items, and is subject to the same item size calculations.
- `SPECIFIC_ATTRIBUTES` - Returns only the attributes listed in `ProjectionExpression`. This return value is equivalent to specifying `ProjectionExpression` without specifying any value for `Select`.

  If you query or scan a local secondary index and request only attributes that are projected into that index, the operation reads only the index and not the table. If any of the requested attributes are not projected into the local secondary index, DynamoDB fetches each of these attributes from the parent table. This extra fetching incurs additional throughput cost and latency.

  If you query or scan a global secondary index, you can only request attributes that are projected into the index. Global secondary index queries cannot fetch attributes from the parent table.

If neither `Select` nor `ProjectionExpression` are specified, DynamoDB defaults to `ALL_ATTRIBUTES` when accessing a table, and `ALL_PROJECTED_ATTRIBUTES` when accessing an index. You cannot use both `Select` and `ProjectionExpression` together in a single

request, unless the value for `Select` is `SPECIFIC_ATTRIBUTES`. (This usage is equivalent to specifying `ProjectionExpression` without any value for `Select`.)

> **Note**
>
> If you use the `ProjectionExpression` parameter, then the value for `Select` can only be `SPECIFIC_ATTRIBUTES`. Any other value for `Select` will return an error.

Type: String

Valid Values: `ALL_ATTRIBUTES | ALL_PROJECTED_ATTRIBUTES | SPECIFIC_ATTRIBUTES | COUNT`

Required: No

TotalSegments

For a parallel `Scan` request, `TotalSegments` represents the total number of segments into which the `Scan` operation will be divided. The value of `TotalSegments` corresponds to the number of application workers that will perform the parallel scan. For example, if you want to use four application threads to scan a table or an index, specify a `TotalSegments` value of 4.

The value for `TotalSegments` must be greater than or equal to 1, and less than or equal to 1000000. If you specify a `TotalSegments` value of 1, the `Scan` operation will be sequential rather than parallel.

If you specify `TotalSegments`, you must also specify `Segment`.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000000.

Required: No

Response Syntax

```
{
   "ConsumedCapacity": {
      "CapacityUnits": number,
      "GlobalSecondaryIndexes": {
         "string" : {
```

```
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "LocalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "ReadCapacityUnits": number,
      "Table": {
         "CapacityUnits": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
      "TableName": "string",
      "WriteCapacityUnits": number
   },
   "Count": number,
   "Items": [
      {
         "string" : {
            "B": blob,
            "BOOL": boolean,
            "BS": [ blob ],
            "L": [
               "AttributeValue"
            ],
            "M": {
               "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
         }
      }
   ],
   "LastEvaluatedKey": {
      "string" : {
```

```
            "B": blob,
            "BOOL": boolean,
            "BS": [ blob ],
            "L": [
               "AttributeValue"
            ],
            "M": {
               "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
         }
      },
      "ScannedCount": number
   }
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The capacity units consumed by the `Scan` operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. `ConsumedCapacity` is only returned if the `ReturnConsumedCapacity` parameter was specified. For more information, see Capacity unit consumption for read operations in the *Amazon DynamoDB Developer Guide*.

Type: ConsumedCapacity object

Count

The number of items in the response.

If you set `ScanFilter` in the request, then `Count` is the number of items returned after the filter was applied, and `ScannedCount` is the number of matching items before the filter was applied.

If you did not use a filter in the request, then `Count` is the same as `ScannedCount`.

Type: Integer

Items

An array of item attributes that match the scan criteria. Each element in this array consists of an attribute name and the value for that attribute.

Type: Array of string to AttributeValue object maps

Key Length Constraints: Maximum length of 65535.

LastEvaluatedKey

The primary key of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If `LastEvaluatedKey` is empty, then the "last page" of results has been processed and there is no more data to be retrieved.

If `LastEvaluatedKey` is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when `LastEvaluatedKey` is empty.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

ScannedCount

The number of items evaluated, before any `ScanFilter` is applied. A high `ScannedCount` value with few, or no, `Count` results indicates an inefficient `Scan` operation. For more information, see Count and ScannedCount in the *Amazon DynamoDB Developer Guide*.

If you did not use a filter in the request, then `ScannedCount` is the same as `Count`.

Type: Integer

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

Examples

Return All Items

The following example returns all of the items in a table. No scan filter is applied.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.Scan

{
    "TableName": "Reply",
    "ReturnConsumedCapacity": "TOTAL"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "ConsumedCapacity": {
        "CapacityUnits": 0.5,
        "TableName": "Reply"
    },
    "Count": 4,
    "Items": [
        {
            "PostedBy": {
                "S": "joe@example.com"
            },
            "ReplyDateTime": {
                "S": "20130320115336"
            },
            "Id": {
                "S": "Amazon DynamoDB#How do I update multiple items?"
            },
```

```
            "Message": {
                "S": "Have you looked at BatchWriteItem?"
            }
        },
        {
            "PostedBy": {
                "S": "fred@example.com"
            },
            "ReplyDateTime": {
                "S": "20130320115342"
            },
            "Id": {
                "S": "Amazon DynamoDB#How do I update multiple items?"
            },
            "Message": {
                "S": "No, I didn't know about that.  Where can I find more
 information?"
            }
        },
        {
            "PostedBy": {
                "S": "joe@example.com"
            },
            "ReplyDateTime": {
                "S": "20130320115347"
            },
            "Id": {
                "S": "Amazon DynamoDB#How do I update multiple items?"
            },
            "Message": {
                "S": "BatchWriteItem is documented in the Amazon DynamoDB API
 Reference."
            }
        },
        {
            "PostedBy": {
                "S": "fred@example.com"
            },
            "ReplyDateTime": {
                "S": "20130320115352"
            },
            "Id": {
                "S": "Amazon DynamoDB#How do I update multiple items?"
            },
```

```
            "Message": {
                "S": "OK, I'll take a look at that.  Thanks!"
            }
        }
    ],
    "ScannedCount": 4
}
```

Use a Filter Expression

The following example returns only those items matching specific criteria.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.Scan

{
    "TableName": "Reply",
    "FilterExpression": "PostedBy = :val",
    "ExpressionAttributeValues": {":val": {"S": "joe@example.com"}},
    "ReturnConsumedCapacity": "TOTAL"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "ConsumedCapacity": {
        "CapacityUnits": 0.5,
```

```
        "TableName": "Reply"
    },
    "Count": 2,
    "Items": [
        {
            "PostedBy": {
                "S": "joe@example.com"
            },
            "ReplyDateTime": {
                "S": "20130320115336"
            },
            "Id": {
                "S": "Amazon DynamoDB#How do I update multiple items?"
            },
            "Message": {
                "S": "Have you looked at BatchWriteItem?"
            }
        },
        {
            "PostedBy": {
                "S": "joe@example.com"
            },
            "ReplyDateTime": {
                "S": "20130320115347"
            },
            "Id": {
                "S": "Amazon DynamoDB#How do I update multiple items?"
            },
            "Message": {
                "S": "BatchWriteItem is documented in the Amazon DynamoDB API
 Reference."
            }
        }
    ],
    "ScannedCount": 4
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagResource

Service: Amazon DynamoDB

Associate a set of tags with an Amazon DynamoDB resource. You can then activate these user-defined tags so that they appear on the Billing and Cost Management console for cost allocation tracking. You can call TagResource up to five times per second, per account.

- `TagResource` is an asynchronous operation. If you issue a ListTagsOfResource request immediately after a `TagResource` request, DynamoDB might return your previous tag set, if there was one, or an empty tag set. This is because `ListTagsOfResource` uses an eventually consistent query, and the metadata for your tags or table might not be available at that moment. Wait for a few seconds, and then try the `ListTagsOfResource` request again.
- The application or removal of tags using `TagResource` and `UntagResource` APIs is eventually consistent. `ListTagsOfResource` API will only reflect the changes after a few seconds.

For an overview on tagging DynamoDB resources, see Tagging for DynamoDB in the *Amazon DynamoDB Developer Guide*.

Request Syntax

```
{
   "ResourceArn": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ResourceArn

Identifies the Amazon DynamoDB resource to which tags should be added. This value is an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1283.

Required: Yes

Tags

The tags to be assigned to the Amazon DynamoDB resource.

Type: Array of Tag objects

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2

- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TransactGetItems

Service: Amazon DynamoDB

`TransactGetItems` is a synchronous operation that atomically retrieves multiple items from one or more tables (but not from indexes) in a single account and Region. A `TransactGetItems` call can contain up to 100 `TransactGetItem` objects, each of which contains a `Get` structure that specifies an item to retrieve from a table in the account and Region. A call to `TransactGetItems` cannot retrieve items from tables in more than one AWS account or Region. The aggregate size of the items in the transaction cannot exceed 4 MB.

DynamoDB rejects the entire `TransactGetItems` request if any of the following is true:

- A conflicting operation is in the process of updating an item to be read.
- There is insufficient provisioned capacity for the transaction to be completed.
- There is a user error, such as an invalid data format.
- The aggregate size of the items in the transaction exceeded 4 MB.

Request Syntax

```
{
   "ReturnConsumedCapacity": "string",
   "TransactItems": [
      {
         "Get": {
            "ExpressionAttributeNames": {
               "string" : "string"
            },
            "Key": {
               "string" : {
                  "B": blob,
                  "BOOL": boolean,
                  "BS": [ blob ],
                  "L": [
                     "AttributeValue"
                  ],
                  "M": {
                     "string" : "AttributeValue"
                  },
                  "N": "string",
                  "NS": [ "string" ],
```

```
                    "NULL": boolean,
                    "S": "string",
                    "SS": [ "string" ]
                }
            },
            "ProjectionExpression": "string",
            "TableName": "string"
        }
    }
  ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TransactItems

An ordered array of up to 100 `TransactGetItem` objects, each of which contains a `Get` structure.

Type: Array of TransactGetItem objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: Yes

ReturnConsumedCapacity

A value of `TOTAL` causes consumed capacity information to be returned, and a value of `NONE` prevents that information from being returned. No other value is valid.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

Response Syntax

```
{
   "ConsumedCapacity": [
      {
         "CapacityUnits": number,
         "GlobalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "LocalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "ReadCapacityUnits": number,
         "Table": {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         },
         "TableName": "string",
         "WriteCapacityUnits": number
      }
   ],
   "Responses": [
      {
         "Item": {
            "string" : {
               "B": blob,
               "BOOL": boolean,
               "BS": [ blob ],
               "L": [
                  "AttributeValue"
               ],
               "M": {
                  "string" : "AttributeValue"
               },
```

```
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
            }
        }
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

If the *ReturnConsumedCapacity* value was `TOTAL`, this is an array of `ConsumedCapacity` objects, one for each table addressed by `TransactGetItem` objects in the *TransactItems* parameter. These `ConsumedCapacity` objects report the read-capacity units consumed by the `TransactGetItems` call in that table.

Type: Array of ConsumedCapacity objects

Responses

An ordered array of up to 100 `ItemResponse` objects, each of which corresponds to the `TransactGetItem` object in the same position in the *TransactItems* array. Each `ItemResponse` object contains a Map of the name-value pairs that are the projected attributes of the requested item.

If a requested item could not be retrieved, the corresponding `ItemResponse` object is Null, or if the requested item has no projected attributes, the corresponding `ItemResponse` object is an empty Map.

Type: Array of ItemResponse objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

TransactionCanceledException

The entire transaction request was canceled.

DynamoDB cancels a `TransactWriteItems` request under the following circumstances:

- A condition in one of the condition expressions is not met.
- A table in the `TransactWriteItems` request is in a different account or region.
- More than one action in the `TransactWriteItems` operation targets the same item.
- There is insufficient provisioned capacity for the transaction to be completed.
- An item size becomes too large (larger than 400 KB), or a local secondary index (LSI) becomes too large, or a similar validation error occurs because of changes made by the transaction.
- There is a user error, such as an invalid data format.
- There is an ongoing `TransactWriteItems` operation that conflicts with a concurrent `TransactWriteItems` request. In this case the `TransactWriteItems` operation fails with a `TransactionCanceledException`.

DynamoDB cancels a `TransactGetItems` request under the following circumstances:

- There is an ongoing `TransactGetItems` operation that conflicts with a concurrent `PutItem`, `UpdateItem`, `DeleteItem` or `TransactWriteItems` request. In this case the `TransactGetItems` operation fails with a `TransactionCanceledException`.
- A table in the `TransactGetItems` request is in a different account or region.
- There is insufficient provisioned capacity for the transaction to be completed.
- There is a user error, such as an invalid data format.

> **Note**
>
> If using Java, DynamoDB lists the cancellation reasons on the `CancellationReasons` property. This property is not set for other languages. Transaction cancellation reasons are ordered in the order of requested items, if an item has no error it will have None code and `Null` message.

Cancellation reason codes and possible error messages:

- No Errors:

  - Code: None
  - Message: `null`
- Conditional Check Failed:
  - Code: `ConditionalCheckFailed`
  - Message: The conditional request failed.
- Item Collection Size Limit Exceeded:
  - Code: `ItemCollectionSizeLimitExceeded`
  - Message: Collection size exceeded.
- Transaction Conflict:
  - Code: `TransactionConflict`
  - Message: Transaction is ongoing for the item.
- Provisioned Throughput Exceeded:
  - Code: `ProvisionedThroughputExceeded`
  - Messages:
    - The level of configured provisioned throughput for the table was exceeded. Consider increasing your provisioning level with the UpdateTable API.

      > **Note**
      >
      > This Message is received when provisioned throughput is exceeded is on a provisioned DynamoDB table.

    - The level of configured provisioned throughput for one or more global secondary indexes of the table was exceeded. Consider increasing your provisioning level for the under-provisioned global secondary indexes with the UpdateTable API.

      > **Note**
      >
      > This message is returned when provisioned throughput is exceeded is on a provisioned GSI.

- Throttling Error:
  - Code: `ThrottlingError`
  - Messages:

- Throughput exceeds the current capacity of your table or index. DynamoDB is automatically scaling your table or index so please try again shortly. If exceptions persist, check if you have a hot key: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html.

  > **Note**
  >
  > This message is returned when writes get throttled on an On-Demand table as DynamoDB is automatically scaling the table.

- Throughput exceeds the current capacity for one or more global secondary indexes. DynamoDB is automatically scaling your index so please try again shortly.

  > **Note**
  >
  > This message is returned when writes get throttled on an On-Demand GSI as DynamoDB is automatically scaling the GSI.

- Validation Error:
  - Code: `ValidationError`
  - Messages:
    - One or more parameter values were invalid.
    - The update expression attempted to update the secondary index key beyond allowed size limits.
    - The update expression attempted to update the secondary index key to unsupported type.
    - An operand in the update expression has an incorrect data type.
    - Item size to update has exceeded the maximum allowed size.
    - Number overflow. Attempting to store a number with magnitude larger than supported range.
    - Type mismatch for attribute to update.
    - Nesting Levels have exceeded supported limits.
    - The document path provided in the update expression is invalid for update.
    - The provided expression refers to an attribute that does not exist in the item.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TransactWriteItems

Service: Amazon DynamoDB

`TransactWriteItems` is a synchronous write operation that groups up to 100 action requests. These actions can target items in different tables, but not in different AWS accounts or Regions, and no two actions can target the same item. For example, you cannot both `ConditionCheck` and `Update` the same item. The aggregate size of the items in the transaction cannot exceed 4 MB.

The actions are completed atomically so that either all of them succeed, or all of them fail. They are defined by the following objects:

- `Put` — Initiates a `PutItem` operation to write a new item. This structure specifies the primary key of the item to be written, the name of the table to write it in, an optional condition expression that must be satisfied for the write to succeed, a list of the item's attributes, and a field indicating whether to retrieve the item's attributes if the condition is not met.
- `Update` — Initiates an `UpdateItem` operation to update an existing item. This structure specifies the primary key of the item to be updated, the name of the table where it resides, an optional condition expression that must be satisfied for the update to succeed, an expression that defines one or more attributes to be updated, and a field indicating whether to retrieve the item's attributes if the condition is not met.
- `Delete` — Initiates a `DeleteItem` operation to delete an existing item. This structure specifies the primary key of the item to be deleted, the name of the table where it resides, an optional condition expression that must be satisfied for the deletion to succeed, and a field indicating whether to retrieve the item's attributes if the condition is not met.
- `ConditionCheck` — Applies a condition to an item that is not being modified by the transaction. This structure specifies the primary key of the item to be checked, the name of the table where it resides, a condition expression that must be satisfied for the transaction to succeed, and a field indicating whether to retrieve the item's attributes if the condition is not met.

DynamoDB rejects the entire `TransactWriteItems` request if any of the following is true:

- A condition in one of the condition expressions is not met.
- An ongoing operation is in the process of updating the same item.
- There is insufficient provisioned capacity for the transaction to be completed.

- An item size becomes too large (bigger than 400 KB), a local secondary index (LSI) becomes too large, or a similar validation error occurs because of changes made by the transaction.
- The aggregate size of the items in the transaction exceeds 4 MB.
- There is a user error, such as an invalid data format.

Request Syntax

```
{
   "ClientRequestToken": "string",
   "ReturnConsumedCapacity": "string",
   "ReturnItemCollectionMetrics": "string",
   "TransactItems": [
      {
         "ConditionCheck": {
            "ConditionExpression": "string",
            "ExpressionAttributeNames": {
               "string" : "string"
            },
            "ExpressionAttributeValues": {
               "string" : {
                  "B": blob,
                  "BOOL": boolean,
                  "BS": [ blob ],
                  "L": [
                     "AttributeValue"
                  ],
                  "M": {
                     "string" : "AttributeValue"
                  },
                  "N": "string",
                  "NS": [ "string" ],
                  "NULL": boolean,
                  "S": "string",
                  "SS": [ "string" ]
               }
            },
            "Key": {
               "string" : {
                  "B": blob,
                  "BOOL": boolean,
                  "BS": [ blob ],
```

```
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "ReturnValuesOnConditionCheckFailure": "string",
          "TableName": "string"
       },
       "Delete": {
          "ConditionExpression": "string",
          "ExpressionAttributeNames": {
             "string" : "string"
          },
          "ExpressionAttributeValues": {
             "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "Key": {
             "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
```

```
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "ReturnValuesOnConditionCheckFailure": "string",
          "TableName": "string"
       },
       "Put": {
          "ConditionExpression": "string",
          "ExpressionAttributeNames": {
             "string" : "string"
          },
          "ExpressionAttributeValues": {
             "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "Item": {
             "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
```

```
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "ReturnValuesOnConditionCheckFailure": "string",
          "TableName": "string"
       },
       "Update": {
          "ConditionExpression": "string",
          "ExpressionAttributeNames": {
             "string" : "string"
          },
          "ExpressionAttributeValues": {
             "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "Key": {
             "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
```

```
                    "L": [
                       "AttributeValue"
                    ],
                    "M": {
                       "string" : "AttributeValue"
                    },
                    "N": "string",
                    "NS": [ "string" ],
                    "NULL": boolean,
                    "S": "string",
                    "SS": [ "string" ]
                 }
              },
              "ReturnValuesOnConditionCheckFailure": "string",
              "TableName": "string",
              "UpdateExpression": "string"
           }
        }
    ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TransactItems

An ordered array of up to 100 `TransactWriteItem` objects, each of which contains a `ConditionCheck`, `Put`, `Update`, or `Delete` object. These can operate on items in different tables, but the tables must reside in the same AWS account and Region, and no two of them can operate on the same item.

Type: Array of TransactWriteItem objects

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Required: Yes

ClientRequestToken

Providing a `ClientRequestToken` makes the call to `TransactWriteItems` idempotent, meaning that multiple identical calls have the same effect as one single call.

Although multiple identical calls using the same client request token produce the same result on the server (no side effects), the responses to the calls might not be the same. If the `ReturnConsumedCapacity` parameter is set, then the initial `TransactWriteItems` call returns the amount of write capacity units consumed in making the changes. Subsequent `TransactWriteItems` calls with the same client token return the number of read capacity units consumed in reading the item.

A client request token is valid for 10 minutes after the first request that uses it is completed. After 10 minutes, any request with the same client token is treated as a new request. Do not resubmit the same request with the same client token for more than 10 minutes, or the result might not be idempotent.

If you submit a request with the same client token but a change in other parameters within the 10-minute idempotency window, DynamoDB returns an `IdempotentParameterMismatch` exception.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 36.

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

ReturnItemCollectionMetrics

Determines whether item collection metrics are returned. If set to `SIZE`, the response includes statistics about item collections (if any), that were modified during the operation and are returned in the response. If set to `NONE` (the default), no statistics are returned.

Type: String

Valid Values: `SIZE | NONE`

Required: No

Response Syntax

```
{
   "ConsumedCapacity": [
      {
         "CapacityUnits": number,
         "GlobalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "LocalSecondaryIndexes": {
            "string" : {
               "CapacityUnits": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            }
         },
         "ReadCapacityUnits": number,
         "Table": {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
```

```
            },
            "TableName": "string",
            "WriteCapacityUnits": number
        }
    ],
    "ItemCollectionMetrics": {
        "string" : [
            {
                "ItemCollectionKey": {
                    "string" : {
                        "B": blob,
                        "BOOL": boolean,
                        "BS": [ blob ],
                        "L": [
                            "AttributeValue"
                        ],
                        "M": {
                            "string" : "AttributeValue"
                        },
                        "N": "string",
                        "NS": [ "string" ],
                        "NULL": boolean,
                        "S": "string",
                        "SS": [ "string" ]
                    }
                },
                "SizeEstimateRangeGB": [ number ]
            }
        ]
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ConsumedCapacity

The capacity units consumed by the entire `TransactWriteItems` operation. The values of the list are ordered according to the ordering of the `TransactItems` request parameter.

Type: Array of ConsumedCapacity objects

ItemCollectionMetrics

A list of tables that were processed by `TransactWriteItems` and, for each table, information about any item collections that were affected by individual `UpdateItem`, `PutItem`, or `DeleteItem` operations.

Type: String to array of ItemCollectionMetrics objects map

Key Length Constraints: Minimum length of 1. Maximum length of 1024.

Errors

For information about the errors that are common to all actions, see Common Errors.

IdempotentParameterMismatchException

DynamoDB rejected the request because you retried a request with a different payload but with an idempotent token that was already used.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

TransactionCanceledException

The entire transaction request was canceled.

DynamoDB cancels a `TransactWriteItems` request under the following circumstances:

- A condition in one of the condition expressions is not met.
- A table in the `TransactWriteItems` request is in a different account or region.
- More than one action in the `TransactWriteItems` operation targets the same item.
- There is insufficient provisioned capacity for the transaction to be completed.
- An item size becomes too large (larger than 400 KB), or a local secondary index (LSI) becomes too large, or a similar validation error occurs because of changes made by the transaction.
- There is a user error, such as an invalid data format.
- There is an ongoing `TransactWriteItems` operation that conflicts with a concurrent `TransactWriteItems` request. In this case the `TransactWriteItems` operation fails with a `TransactionCanceledException`.

DynamoDB cancels a `TransactGetItems` request under the following circumstances:

- There is an ongoing `TransactGetItems` operation that conflicts with a concurrent `PutItem`, `UpdateItem`, `DeleteItem` or `TransactWriteItems` request. In this case the `TransactGetItems` operation fails with a `TransactionCanceledException`.
- A table in the `TransactGetItems` request is in a different account or region.
- There is insufficient provisioned capacity for the transaction to be completed.

- There is a user error, such as an invalid data format.

> **Note**
>
> If using Java, DynamoDB lists the cancellation reasons on the `CancellationReasons` property. This property is not set for other languages. Transaction cancellation reasons are ordered in the order of requested items, if an item has no error it will have None code and `Null` message.

Cancellation reason codes and possible error messages:

- No Errors:
  - Code: None
  - Message: `null`
- Conditional Check Failed:
  - Code: `ConditionalCheckFailed`
  - Message: The conditional request failed.
- Item Collection Size Limit Exceeded:
  - Code: `ItemCollectionSizeLimitExceeded`
  - Message: Collection size exceeded.
- Transaction Conflict:
  - Code: `TransactionConflict`
  - Message: Transaction is ongoing for the item.
- Provisioned Throughput Exceeded:
  - Code: `ProvisionedThroughputExceeded`
  - Messages:
    - The level of configured provisioned throughput for the table was exceeded. Consider increasing your provisioning level with the UpdateTable API.

    > **Note**
    >
    > This Message is received when provisioned throughput is exceeded is on a provisioned DynamoDB table.

- The level of configured provisioned throughput for one or more global secondary indexes of the table was exceeded. Consider increasing your provisioning level for the under-provisioned global secondary indexes with the UpdateTable API.

  > **Note**
  >
  > This message is returned when provisioned throughput is exceeded is on a provisioned GSI.

- Throttling Error:
  - Code: `ThrottlingError`
  - Messages:
    - Throughput exceeds the current capacity of your table or index. DynamoDB is automatically scaling your table or index so please try again shortly. If exceptions persist, check if you have a hot key: https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-partition-key-design.html.

      > **Note**
      >
      > This message is returned when writes get throttled on an On-Demand table as DynamoDB is automatically scaling the table.

    - Throughput exceeds the current capacity for one or more global secondary indexes. DynamoDB is automatically scaling your index so please try again shortly.

      > **Note**
      >
      > This message is returned when writes get throttled on an On-Demand GSI as DynamoDB is automatically scaling the GSI.

- Validation Error:
  - Code: `ValidationError`
  - Messages:
    - One or more parameter values were invalid.
    - The update expression attempted to update the secondary index key beyond allowed size limits.

- The update expression attempted to update the secondary index key to unsupported type.
- An operand in the update expression has an incorrect data type.
- Item size to update has exceeded the maximum allowed size.
- Number overflow. Attempting to store a number with magnitude larger than supported range.
- Type mismatch for attribute to update.
- Nesting Levels have exceeded supported limits.
- The document path provided in the update expression is invalid for update.
- The provided expression refers to an attribute that does not exist in the item.

HTTP Status Code: 400

TransactionInProgressException

The transaction with the given request token is already in progress.

Recommended Settings

> **Note**
>
> This is a general recommendation for handling the `TransactionInProgressException`. These settings help ensure that the client retries will trigger completion of the ongoing `TransactWriteItems` request.

- Set `clientExecutionTimeout` to a value that allows at least one retry to be processed after 5 seconds have elapsed since the first attempt for the `TransactWriteItems` operation.
- Set `socketTimeout` to a value a little lower than the `requestTimeout` setting.
- `requestTimeout` should be set based on the time taken for the individual retries of a single HTTP request for your use case, but setting it to 1 second or higher should work well to reduce chances of retries and `TransactionInProgressException` errors.
- Use exponential backoff when retrying and tune backoff if needed.

Assuming default retry policy, example timeout settings based on the guidelines above are as follows:

Example timeline:

- 0-1000 first attempt
- 1000-1500 first sleep/delay (default retry policy uses 500 ms as base delay for 4xx errors)
- 1500-2500 second attempt
- 2500-3500 second sleep/delay (500 * 2, exponential backoff)
- 3500-4500 third attempt
- 4500-6500 third sleep/delay (500 * 2^2)
- 6500-7500 fourth attempt (this can trigger inline recovery since 5 seconds have elapsed since the first attempt reached TC)

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagResource

Service: Amazon DynamoDB

Removes the association of tags from an Amazon DynamoDB resource. You can call `UntagResource` up to five times per second, per account.

- `UntagResource` is an asynchronous operation. If you issue a ListTagsOfResource request immediately after an `UntagResource` request, DynamoDB might return your previous tag set, if there was one, or an empty tag set. This is because `ListTagsOfResource` uses an eventually consistent query, and the metadata for your tags or table might not be available at that moment. Wait for a few seconds, and then try the `ListTagsOfResource` request again.
- The application or removal of tags using `TagResource` and `UntagResource` APIs is eventually consistent. `ListTagsOfResource` API will only reflect the changes after a few seconds.

For an overview on tagging DynamoDB resources, see Tagging for DynamoDB in the *Amazon DynamoDB Developer Guide*.

Request Syntax

```
{
   "ResourceArn": "string",
   "TagKeys": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ResourceArn

The DynamoDB resource that the tags will be removed from. This value is an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1283.

Required: Yes

TagKeys

A list of tag keys. Existing tags of the resource whose keys are members of this list will be removed from the DynamoDB resource.

Type: Array of strings

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Response Elements

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateContinuousBackups

Service: Amazon DynamoDB

`UpdateContinuousBackups` enables or disables point in time recovery for the specified table. A successful `UpdateContinuousBackups` call returns the current `ContinuousBackupsDescription`. Continuous backups are `ENABLED` on all tables at table creation. If point in time recovery is enabled, `PointInTimeRecoveryStatus` will be set to ENABLED.

Once continuous backups and point in time recovery are enabled, you can restore to any point in time within `EarliestRestorableDateTime` and `LatestRestorableDateTime`.

`LatestRestorableDateTime` is typically 5 minutes before the current time. You can restore your table to any point in time in the last 35 days. You can set the `RecoveryPeriodInDays` to any value between 1 and 35 days.

Request Syntax

```
{
   "PointInTimeRecoverySpecification": {
      "PointInTimeRecoveryEnabled": boolean,
      "RecoveryPeriodInDays": number
   },
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

PointInTimeRecoverySpecification

Represents the settings used to enable point in time recovery.

Type: PointInTimeRecoverySpecification object

Required: Yes

TableName

The name of the table. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

Response Syntax

```
{
   "ContinuousBackupsDescription": {
      "ContinuousBackupsStatus": "string",
      "PointInTimeRecoveryDescription": {
         "EarliestRestorableDateTime": number,
         "LatestRestorableDateTime": number,
         "PointInTimeRecoveryStatus": "string",
         "RecoveryPeriodInDays": number
      }
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ContinuousBackupsDescription

Represents the continuous backups and point in time recovery settings on the table.

Type: ContinuousBackupsDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

ContinuousBackupsUnavailableException

Backups have not yet been enabled for this table.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

TableNotFoundException

A source table with the name `TableName` does not currently exist within the subscriber's account or the subscriber is operating in the wrong AWS Region.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateContributorInsights

Service: Amazon DynamoDB

Updates the status for contributor insights for a specific table or index. CloudWatch Contributor Insights for DynamoDB graphs display the partition key and (if applicable) sort key of frequently accessed items and frequently throttled items in plaintext. If you require the use of AWS Key Management Service (KMS) to encrypt this table's partition key and sort key data with an AWS managed key or customer managed key, you should not enable CloudWatch Contributor Insights for DynamoDB for this table.

Request Syntax

```
{
   "ContributorInsightsAction": "string",
   "ContributorInsightsMode": "string",
   "IndexName": "string",
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ContributorInsightsAction

Represents the contributor insights action.

Type: String

Valid Values: `ENABLE | DISABLE`

Required: Yes

TableName

The name of the table. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ContributorInsightsMode

Specifies whether to track all access and throttled events or throttled events only for the DynamoDB table or index.

Type: String

Valid Values: `ACCESSED_AND_THROTTLED_KEYS | THROTTLED_KEYS`

Required: No

IndexName

The global secondary index name, if applicable.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Response Syntax

```
{
   "ContributorInsightsMode": "string",
   "ContributorInsightsStatus": "string",
   "IndexName": "string",
   "TableName": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ContributorInsightsMode

The updated mode of CloudWatch Contributor Insights that determines whether to monitor all access and throttled events or to track throttled events exclusively.

Type: String

Valid Values: `ACCESSED_AND_THROTTLED_KEYS | THROTTLED_KEYS`

ContributorInsightsStatus

The status of contributor insights

Type: String

Valid Values: `ENABLING | ENABLED | DISABLING | DISABLED | FAILED`

IndexName

The name of the global secondary index, if applicable.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

TableName

The name of the table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateGlobalTable

Service: Amazon DynamoDB

Adds or removes replicas in the specified global table. The global table must already exist to be able to use this operation. Any replica to be added must be empty, have the same name as the global table, have the same key schema, have DynamoDB Streams enabled, and have the same provisioned and maximum write capacity units.

> **Important**
>
> This documentation is for version 2017.11.29 (Legacy) of global tables, which should be avoided for new global tables. Customers should use Global Tables version 2019.11.21 (Current) when possible, because it provides greater flexibility, higher efficiency, and consumes less write capacity than 2017.11.29 (Legacy).
> To determine which version you're using, see Determining the global table version you are using. To update existing global tables from version 2017.11.29 (Legacy) to version 2019.11.21 (Current), see Upgrading global tables.

> **Note**
>
> If you are using global tables Version 2019.11.21 (Current) you can use UpdateTable instead.
> Although you can use `UpdateGlobalTable` to add replicas and remove replicas in a single request, for simplicity we recommend that you issue separate requests for adding or removing replicas.

If global secondary indexes are specified, then the following conditions must also be met:

- The global secondary indexes must have the same name.
- The global secondary indexes must have the same hash key and sort key (if present).
- The global secondary indexes must have the same provisioned and maximum write capacity units.

Request Syntax

```
{
   "GlobalTableName": "string",
   "ReplicaUpdates": [
      {
         "Create": {
            "RegionName": "string"
         },
         "Delete": {
            "RegionName": "string"
         }
      }
   ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

GlobalTableName

The global table name.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

ReplicaUpdates

A list of Regions that should be added or removed from the global table.

Type: Array of ReplicaUpdate objects

Required: Yes

Response Syntax

```
{
   "GlobalTableDescription": {
      "CreationDateTime": number,
      "GlobalTableArn": "string",
      "GlobalTableName": "string",
      "GlobalTableStatus": "string",
      "ReplicationGroup": [
         {
            "GlobalSecondaryIndexes": [
               {
                  "IndexName": "string",
                  "OnDemandThroughputOverride": {
                     "MaxReadRequestUnits": number
                  },
                  "ProvisionedThroughputOverride": {
                     "ReadCapacityUnits": number
                  },
                  "WarmThroughput": {
                     "ReadUnitsPerSecond": number,
                     "Status": "string",
                     "WriteUnitsPerSecond": number
                  }
               }
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
               "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
               "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "ReplicaInaccessibleDateTime": number,
            "ReplicaStatus": "string",
            "ReplicaStatusDescription": "string",
            "ReplicaStatusPercentProgress": "string",
            "ReplicaTableClassSummary": {
               "LastUpdateDateTime": number,
               "TableClass": "string"
```

```
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ]
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GlobalTableDescription

Contains the details of the global table.

Type: GlobalTableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

GlobalTableNotFoundException

The specified global table does not exist.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ReplicaAlreadyExistsException

The specified replica is already part of the global table.

HTTP Status Code: 400

ReplicaNotFoundException

The specified replica is no longer part of the global table.

HTTP Status Code: 400

TableNotFoundException

A source table with the name `TableName` does not currently exist within the subscriber's account or the subscriber is operating in the wrong AWS Region.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateGlobalTableSettings

Service: Amazon DynamoDB

Updates settings for a global table.

> **Important**
>
> This documentation is for version 2017.11.29 (Legacy) of global tables, which should be avoided for new global tables. Customers should use Global Tables version 2019.11.21 (Current) when possible, because it provides greater flexibility, higher efficiency, and consumes less write capacity than 2017.11.29 (Legacy).
> To determine which version you're using, see Determining the global table version you are using. To update existing global tables from version 2017.11.29 (Legacy) to version 2019.11.21 (Current), see Upgrading global tables.

Request Syntax

```
{
   "GlobalTableBillingMode": "string",
   "GlobalTableGlobalSecondaryIndexSettingsUpdate": [
      {
         "IndexName": "string",
         "ProvisionedWriteCapacityAutoScalingSettingsUpdate": {
            "AutoScalingDisabled": boolean,
            "AutoScalingRoleArn": "string",
            "MaximumUnits": number,
            "MinimumUnits": number,
            "ScalingPolicyUpdate": {
               "PolicyName": "string",
               "TargetTrackingScalingPolicyConfiguration": {
                  "DisableScaleIn": boolean,
                  "ScaleInCooldown": number,
                  "ScaleOutCooldown": number,
                  "TargetValue": number
               }
            }
         },
         "ProvisionedWriteCapacityUnits": number
      }
   ],
```

```
    "GlobalTableName": "string",
    "GlobalTableProvisionedWriteCapacityAutoScalingSettingsUpdate": {
       "AutoScalingDisabled": boolean,
       "AutoScalingRoleArn": "string",
       "MaximumUnits": number,
       "MinimumUnits": number,
       "ScalingPolicyUpdate": {
          "PolicyName": "string",
          "TargetTrackingScalingPolicyConfiguration": {
             "DisableScaleIn": boolean,
             "ScaleInCooldown": number,
             "ScaleOutCooldown": number,
             "TargetValue": number
          }
       }
    },
    "GlobalTableProvisionedWriteCapacityUnits": number,
    "ReplicaSettingsUpdate": [
       {
          "RegionName": "string",
          "ReplicaGlobalSecondaryIndexSettingsUpdate": [
             {
                "IndexName": "string",
                "ProvisionedReadCapacityAutoScalingSettingsUpdate": {
                   "AutoScalingDisabled": boolean,
                   "AutoScalingRoleArn": "string",
                   "MaximumUnits": number,
                   "MinimumUnits": number,
                   "ScalingPolicyUpdate": {
                      "PolicyName": "string",
                      "TargetTrackingScalingPolicyConfiguration": {
                         "DisableScaleIn": boolean,
                         "ScaleInCooldown": number,
                         "ScaleOutCooldown": number,
                         "TargetValue": number
                      }
                   }
                },
                "ProvisionedReadCapacityUnits": number
             }
          ],
          "ReplicaProvisionedReadCapacityAutoScalingSettingsUpdate": {
             "AutoScalingDisabled": boolean,
             "AutoScalingRoleArn": "string",
```

```
            "MaximumUnits": number,
            "MinimumUnits": number,
            "ScalingPolicyUpdate": {
               "PolicyName": "string",
               "TargetTrackingScalingPolicyConfiguration": {
                  "DisableScaleIn": boolean,
                  "ScaleInCooldown": number,
                  "ScaleOutCooldown": number,
                  "TargetValue": number
               }
            }
         },
         "ReplicaProvisionedReadCapacityUnits": number,
         "ReplicaTableClass": "string"
      }
   ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

GlobalTableName

The name of the global table

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

GlobalTableBillingMode

The billing mode of the global table. If `GlobalTableBillingMode` is not specified, the global table defaults to `PROVISIONED` capacity billing mode.

- `PROVISIONED` - We recommend using `PROVISIONED` for predictable workloads. `PROVISIONED` sets the billing mode to Provisioned capacity mode.
- `PAY_PER_REQUEST` - We recommend using `PAY_PER_REQUEST` for unpredictable workloads. `PAY_PER_REQUEST` sets the billing mode to On-demand capacity mode.

Type: String

Valid Values: `PROVISIONED | PAY_PER_REQUEST`

Required: No

GlobalTableGlobalSecondaryIndexSettingsUpdate

Represents the settings of a global secondary index for a global table that will be modified.

Type: Array of GlobalTableGlobalSecondaryIndexSettingsUpdate objects

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Required: No

GlobalTableProvisionedWriteCapacityAutoScalingSettingsUpdate

Auto scaling settings for managing provisioned write capacity for the global table.

Type: AutoScalingSettingsUpdate object

Required: No

GlobalTableProvisionedWriteCapacityUnits

The maximum number of writes consumed per second before DynamoDB returns a `ThrottlingException`.

Type: Long

Valid Range: Minimum value of 1.

Required: No

ReplicaSettingsUpdate

Represents the settings for a global table in a Region that will be modified.

Type: Array of ReplicaSettingsUpdate objects

Array Members: Minimum number of 1 item. Maximum number of 50 items.

Required: No

Response Syntax

```
{
   "GlobalTableName": "string",
   "ReplicaSettings": [
      {
         "RegionName": "string",
         "ReplicaBillingModeSummary": {
            "BillingMode": "string",
            "LastUpdateToPayPerRequestDateTime": number
         },
         "ReplicaGlobalSecondaryIndexSettings": [
            {
               "IndexName": "string",
               "IndexStatus": "string",
               "ProvisionedReadCapacityAutoScalingSettings": {
                  "AutoScalingDisabled": boolean,
                  "AutoScalingRoleArn": "string",
                  "MaximumUnits": number,
                  "MinimumUnits": number,
                  "ScalingPolicies": [
                     {
                        "PolicyName": "string",
                        "TargetTrackingScalingPolicyConfiguration": {
                           "DisableScaleIn": boolean,
                           "ScaleInCooldown": number,
                           "ScaleOutCooldown": number,
                           "TargetValue": number
                        }
                     }
                  ]
               },
               "ProvisionedReadCapacityUnits": number,
               "ProvisionedWriteCapacityAutoScalingSettings": {
                  "AutoScalingDisabled": boolean,
                  "AutoScalingRoleArn": "string",
                  "MaximumUnits": number,
                  "MinimumUnits": number,
                  "ScalingPolicies": [
                     {
                        "PolicyName": "string",
```

```
                    "TargetTrackingScalingPolicyConfiguration": {
                       "DisableScaleIn": boolean,
                       "ScaleInCooldown": number,
                       "ScaleOutCooldown": number,
                       "TargetValue": number
                    }
                 }
              ]
           },
           "ProvisionedWriteCapacityUnits": number
        }
     ],
     "ReplicaProvisionedReadCapacityAutoScalingSettings": {
        "AutoScalingDisabled": boolean,
        "AutoScalingRoleArn": "string",
        "MaximumUnits": number,
        "MinimumUnits": number,
        "ScalingPolicies": [
           {
              "PolicyName": "string",
              "TargetTrackingScalingPolicyConfiguration": {
                 "DisableScaleIn": boolean,
                 "ScaleInCooldown": number,
                 "ScaleOutCooldown": number,
                 "TargetValue": number
              }
           }
        ]
     },
     "ReplicaProvisionedReadCapacityUnits": number,
     "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
        "AutoScalingDisabled": boolean,
        "AutoScalingRoleArn": "string",
        "MaximumUnits": number,
        "MinimumUnits": number,
        "ScalingPolicies": [
           {
              "PolicyName": "string",
              "TargetTrackingScalingPolicyConfiguration": {
                 "DisableScaleIn": boolean,
                 "ScaleInCooldown": number,
                 "ScaleOutCooldown": number,
                 "TargetValue": number
              }
```

```
                }
            ]
        },
        "ReplicaProvisionedWriteCapacityUnits": number,
        "ReplicaStatus": "string",
        "ReplicaTableClassSummary": {
            "LastUpdateDateTime": number,
            "TableClass": "string"
        }
    }
  ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

GlobalTableName

The name of the global table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

ReplicaSettings

The Region-specific settings for the global table.

Type: Array of ReplicaSettingsDescription objects

Errors

For information about the errors that are common to all actions, see Common Errors.

GlobalTableNotFoundException

The specified global table does not exist.

HTTP Status Code: 400

IndexNotFoundException

The operation tried to access a nonexistent index.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ReplicaNotFoundException

The specified replica is no longer part of the global table.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateItem

Service: Amazon DynamoDB

Edits an existing item's attributes, or adds a new item to the table if it does not already exist. You can put, delete, or add attribute values. You can also perform a conditional update on an existing item (insert a new attribute name-value pair if it doesn't exist, or replace an existing name-value pair if it has certain expected attribute values).

You can also return the item's attribute values in the same `UpdateItem` operation using the `ReturnValues` parameter.

Request Syntax

```
{
   "AttributeUpdates": {
      "string" : {
         "Action": "string",
         "Value": {
            "B": blob,
            "BOOL": boolean,
            "BS": [ blob ],
            "L": [
               "AttributeValue"
            ],
            "M": {
               "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
         }
      }
   },
   "ConditionalOperator": "string",
   "ConditionExpression": "string",
   "Expected": {
      "string" : {
         "AttributeValueList": [
            {
               "B": blob,
```

```
                "BOOL": boolean,
                "BS": [ blob ],
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          ],
          "ComparisonOperator": "string",
          "Exists": boolean,
          "Value": {
             "B": blob,
             "BOOL": boolean,
             "BS": [ blob ],
             "L": [
                "AttributeValue"
             ],
             "M": {
                "string" : "AttributeValue"
             },
             "N": "string",
             "NS": [ "string" ],
             "NULL": boolean,
             "S": "string",
             "SS": [ "string" ]
          }
       }
    },
    "ExpressionAttributeNames": {
       "string" : "string"
    },
    "ExpressionAttributeValues": {
       "string" : {
          "B": blob,
          "BOOL": boolean,
          "BS": [ blob ],
          "L": [
```

```
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "Key": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ReturnConsumedCapacity": "string",
   "ReturnItemCollectionMetrics": "string",
   "ReturnValues": "string",
   "ReturnValuesOnConditionCheckFailure": "string",
   "TableName": "string",
   "UpdateExpression": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

Key

The primary key of the item to be updated. Each element consists of an attribute name and a value for that attribute.

For the primary key, you must provide all of the attributes. For example, with a simple primary key, you only need to provide a value for the partition key. For a composite primary key, you must provide values for both the partition key and the sort key.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

The name of the table containing the item to update. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

AttributeUpdates

This is a legacy parameter. Use `UpdateExpression` instead. For more information, see AttributeUpdates in the *Amazon DynamoDB Developer Guide*.

Type: String to AttributeValueUpdate object map

Key Length Constraints: Maximum length of 65535.

Required: No

ConditionalOperator

This is a legacy parameter. Use `ConditionExpression` instead. For more information, see ConditionalOperator in the *Amazon DynamoDB Developer Guide*.

Type: String

Valid Values: `AND | OR`

Required: No

ConditionExpression

A condition that must be satisfied in order for a conditional update to succeed.

An expression can contain any of the following:

- Functions: `attribute_exists | attribute_not_exists | attribute_type | contains | begins_with | size`

  These function names are case-sensitive.
- Comparison operators: `= | <> | < | > | <= | >= | BETWEEN | IN`
- Logical operators: `AND | OR | NOT`

For more information about condition expressions, see Specifying Conditions in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

Expected

This is a legacy parameter. Use `ConditionExpression` instead. For more information, see Expected in the *Amazon DynamoDB Developer Guide*.

Type: String to ExpectedAttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression. The following are some use cases for using `ExpressionAttributeNames`:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.

- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the **#** character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*.) To work around this, you could specify the following for `ExpressionAttributeNames`:

- `{"#P":"Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

> **Note**
>
> Tokens that begin with the **:** character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information about expression attribute names, see Specifying Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression.

Use the **:** (colon) character in an expression to dereference an attribute value. For example, suppose that you wanted to check whether the value of the `ProductStatus` attribute was one of the following:

```
Available | Backordered | Discontinued
```

You would first need to specify `ExpressionAttributeValues` as follows:

```
{ ":avail":{"S":"Available"}, ":back":{"S":"Backordered"}, ":disc":
{"S":"Discontinued"} }
```

You could then use these values in an expression, such as this:

```
ProductStatus IN (:avail, :back, :disc)
```

For more information on expression attribute values, see Condition Expressions in the *Amazon DynamoDB Developer Guide*.

Type: String to AttributeValue object map

Required: No

ReturnConsumedCapacity

Determines the level of detail about either provisioned or on-demand throughput consumption that is returned in the response:

- `INDEXES` - The response includes the aggregate `ConsumedCapacity` for the operation, together with `ConsumedCapacity` for each table and secondary index that was accessed.

  Note that some operations, such as `GetItem` and `BatchGetItem`, do not access any indexes at all. In these cases, specifying `INDEXES` will only return `ConsumedCapacity` information for table(s).
- `TOTAL` - The response includes only the aggregate `ConsumedCapacity` for the operation.
- `NONE` - No `ConsumedCapacity` details are included in the response.

Type: String

Valid Values: `INDEXES | TOTAL | NONE`

Required: No

ReturnItemCollectionMetrics

Determines whether item collection metrics are returned. If set to `SIZE`, the response includes statistics about item collections, if any, that were modified during the operation are returned in the response. If set to `NONE` (the default), no statistics are returned.

Type: String

Valid Values: `SIZE | NONE`

Required: No

ReturnValues

Use `ReturnValues` if you want to get the item attributes as they appear before or after they are successfully updated. For `UpdateItem`, the valid values are:

- `NONE` - If `ReturnValues` is not specified, or if its value is `NONE`, then nothing is returned. (This setting is the default for `ReturnValues`.)
- `ALL_OLD` - Returns all of the attributes of the item, as they appeared before the UpdateItem operation.
- `UPDATED_OLD` - Returns only the updated attributes, as they appeared before the UpdateItem operation.
- `ALL_NEW` - Returns all of the attributes of the item, as they appear after the UpdateItem operation.
- `UPDATED_NEW` - Returns only the updated attributes, as they appear after the UpdateItem operation.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

The values returned are strongly consistent.

Type: String

Valid Values: `NONE | ALL_OLD | UPDATED_OLD | ALL_NEW | UPDATED_NEW`

Required: No

ReturnValuesOnConditionCheckFailure

An optional parameter that returns the item attributes for an `UpdateItem` operation that failed a condition check.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

Type: String

Valid Values: `ALL_OLD | NONE`

Required: No

UpdateExpression

An expression that defines one or more attributes to be updated, the action to be performed on them, and new values for them.

The following action values are available for `UpdateExpression`.

- `SET` - Adds one or more attributes and values to an item. If any of these attributes already exist, they are replaced by the new values. You can also use `SET` to add or subtract from an attribute that is of type Number. For example: `SET myNum = myNum + :val`

  `SET` supports the following functions:

  - `if_not_exists (path, operand)` - if the item does not contain an attribute at the specified path, then `if_not_exists` evaluates to operand; otherwise, it evaluates to path. You can use this function to avoid overwriting an attribute that may already be present in the item.
  - `list_append (operand, operand)` - evaluates to a list with a new element added to it. You can append the new element to the start or the end of the list by reversing the order of the operands.

  These function names are case-sensitive.
- `REMOVE` - Removes one or more attributes from an item.
- `ADD` - Adds the specified value to the item, if the attribute does not already exist. If the attribute does exist, then the behavior of `ADD` depends on the data type of the attribute:
  - If the existing attribute is a number, and if `Value` is also a number, then `Value` is mathematically added to the existing attribute. If `Value` is a negative number, then it is subtracted from the existing attribute.

> **Note**
>
> If you use `ADD` to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses `0` as the initial value.
> Similarly, if you use `ADD` for an existing item to increment or decrement an attribute value that doesn't exist before the update, DynamoDB uses `0` as the initial value. For example, suppose that the item you want to update doesn't have an attribute named `itemcount`, but you decide to `ADD` the number `3` to this attribute anyway. DynamoDB will create the `itemcount` attribute, set its initial value to `0`, and finally

add 3 to it. The result will be a new `itemcount` attribute in the item, with a value of 3.

- If the existing data type is a set and if `Value` is also a set, then `Value` is added to the existing set. For example, if the attribute value is the set `[1,2]`, and the `ADD` action specified `[3]`, then the final attribute value is `[1,2,3]`. An error occurs if an `ADD` action is specified for a set attribute and the attribute type specified does not match the existing set type.

  Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, the `Value` must also be a set of strings.

> **Important**
>
> The `ADD` action only supports Number and set data types. In addition, `ADD` can only be used on top-level attributes, not nested attributes.

- `DELETE` - Deletes an element from a set.

  If a set of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set `[a,b,c]` and the `DELETE` action specifies `[a,c]`, then the final attribute value is `[b]`. Specifying an empty set is an error.

> **Important**
>
> The `DELETE` action only supports set data types. In addition, `DELETE` can only be used on top-level attributes, not nested attributes.

You can have many actions in a single expression, such as the following: `SET a=:value1, b=:value2 DELETE :value3, :value4, :value5`

For more information on update expressions, see Modifying Items and Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

Response Syntax

```
{
   "Attributes": {
      "string" : {
         "B": blob,
         "BOOL": boolean,
         "BS": [ blob ],
         "L": [
            "AttributeValue"
         ],
         "M": {
            "string" : "AttributeValue"
         },
         "N": "string",
         "NS": [ "string" ],
         "NULL": boolean,
         "S": "string",
         "SS": [ "string" ]
      }
   },
   "ConsumedCapacity": {
      "CapacityUnits": number,
      "GlobalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "LocalSecondaryIndexes": {
         "string" : {
            "CapacityUnits": number,
            "ReadCapacityUnits": number,
            "WriteCapacityUnits": number
         }
      },
      "ReadCapacityUnits": number,
      "Table": {
         "CapacityUnits": number,
         "ReadCapacityUnits": number,
         "WriteCapacityUnits": number
      },
```

```
      "TableName": "string",
      "WriteCapacityUnits": number
   },
   "ItemCollectionMetrics": {
      "ItemCollectionKey": {
         "string" : {
            "B": blob,
            "BOOL": boolean,
            "BS": [ blob ],
            "L": [
               "AttributeValue"
            ],
            "M": {
               "string" : "AttributeValue"
            },
            "N": "string",
            "NS": [ "string" ],
            "NULL": boolean,
            "S": "string",
            "SS": [ "string" ]
         }
      },
      "SizeEstimateRangeGB": [ number ]
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Attributes

A map of attribute values as they appear before or after the `UpdateItem` operation, as determined by the `ReturnValues` parameter.

The `Attributes` map is only present if the update was successful and `ReturnValues` was specified as something other than `NONE` in the request. Each element represents one attribute.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

ConsumedCapacity

The capacity units consumed by the `UpdateItem` operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. `ConsumedCapacity` is only returned if the `ReturnConsumedCapacity` parameter was specified. For more information, see Capacity unity consumption for write operations in the *Amazon DynamoDB Developer Guide*.

Type: ConsumedCapacity object

ItemCollectionMetrics

Information about item collections, if any, that were affected by the `UpdateItem` operation. `ItemCollectionMetrics` is only returned if the `ReturnItemCollectionMetrics` parameter was specified. If the table does not have any local secondary indexes, this information is not returned in the response.

Each `ItemCollectionMetrics` element consists of:

- `ItemCollectionKey` - The partition key value of the item collection. This is the same as the partition key value of the item itself.
- `SizeEstimateRangeGB` - An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

  The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

Type: ItemCollectionMetrics object

Errors

For information about the errors that are common to all actions, see Common Errors.

ConditionalCheckFailedException

A condition specified in the operation failed to be evaluated.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ItemCollectionSizeLimitExceededException

An item collection is too large. This exception is only returned for tables that have one or more local secondary indexes.

HTTP Status Code: 400

ProvisionedThroughputExceededException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. The AWS SDKs for DynamoDB automatically retry requests that receive this exception. Your request is eventually successful, unless your retry queue is too large to finish. Reduce the frequency of requests and use exponential backoff. For more information, go to Error Retries and Exponential Backoff in the *Amazon DynamoDB Developer Guide*.

HTTP Status Code: 400

ReplicatedWriteConflictException

The request was rejected because one or more items in the request are being modified by a request in another Region.

HTTP Status Code: 400

RequestLimitExceeded

Throughput exceeds the current throughput quota for your account. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception. Contact Support to request a quota increase.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ThrottlingException

The request was denied due to request throttling. For detailed information about why the request was throttled and the ARN of the impacted resource, find the ThrottlingReason field in the returned exception.

HTTP Status Code: 400

TransactionConflictException

Operation was rejected because there is an ongoing transaction for the item.

HTTP Status Code: 400

Examples

Conditional Update

This example updates the `Thread` table, changing the `LastPostedBy` attribute, but only if `LastPostedBy` is currently "`fred@example.com`". All of the item's attributes, as they appear after the update, are returned in the response.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.UpdateItem

{
    "TableName": "Thread",
    "Key": {
        "ForumName": {
            "S": "Amazon DynamoDB"
        },
        "Subject": {
            "S": "Maximum number of items?"
```

```
        }
    },
    "UpdateExpression": "set LastPostedBy = :val1",
    "ConditionExpression": "LastPostedBy = :val2",
    "ExpressionAttributeValues": {
        ":val1": {"S": "alice@example.com"},
        ":val2": {"S": "fred@example.com"}
    },
    "ReturnValues": "ALL_NEW"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "Attributes": {
        "LastPostedBy": {
            "S": "alice@example.com"
        },
        "ForumName": {
            "S": "Amazon DynamoDB"
        },
        "LastPostDateTime": {
            "S": "20130320010350"
        },
        "Tags": {
            "SS": ["Update","Multiple Items","HelpMe"]
        },
        "Subject": {
            "S": "Maximum number of items?"
        },
        "Views": {
            "N": "5"
        },
        "Message": {
            "S": "I want to put 10 million data items to an Amazon DynamoDB table.  Is
 there an upper limit?"
        }
```

```
    }
}
```

Atomic Counter

The following example increments the `Replies` attribute in the `Thread` table whenever someone posts a reply. Because `ReturnValues` is set to `NONE`, no output appears in the response payload.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.UpdateItem

{
    "TableName": "Thread",
    "Key": {
        "ForumName": {
            "S": "Amazon DynamoDB"
        },
        "Subject": {
            "S": "A question about updates"
        }
    },
    "UpdateExpression": "set Replies = Replies + :num",
    "ExpressionAttributeValues": {
        ":num": {"N": "1"}
    },
    "ReturnValues" : "NONE"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
```

```
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateKinesisStreamingDestination

Service: Amazon DynamoDB

The command to update the Kinesis stream destination.

Request Syntax

```
{
   "StreamArn": "string",
   "TableName": "string",
   "UpdateKinesisStreamingConfiguration": {
      "ApproximateCreationDateTimePrecision": "string"
   }
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

StreamArn

The Amazon Resource Name (ARN) for the Kinesis stream input.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

TableName

The table name for the Kinesis streaming destination input. You can also provide the ARN of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

UpdateKinesisStreamingConfiguration

The command to update the Kinesis stream configuration.

Type: UpdateKinesisStreamingConfiguration object

Required: No

Response Syntax

```
{
   "DestinationStatus": "string",
   "StreamArn": "string",
   "TableName": "string",
   "UpdateKinesisStreamingConfiguration": {
      "ApproximateCreationDateTimePrecision": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DestinationStatus

The status of the attempt to update the Kinesis streaming destination output.

Type: String

Valid Values: `ENABLING | ACTIVE | DISABLING | DISABLED | ENABLE_FAILED | UPDATING`

StreamArn

The ARN for the Kinesis stream input.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

TableName

The table name for the Kinesis streaming destination output.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

UpdateKinesisStreamingConfiguration

The command to update the Kinesis streaming destination configuration.

Type: UpdateKinesisStreamingConfiguration object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Update the configuration of a kinesis streaming destination on a table

This example updates a kinesis streaming destination to produce records with ApproximateCreationDateTime timestamps at millisecond precision.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.UpdateKinesisStreamingDestination
```

```
{
    "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/example_stream",
    "TableName": "example_table",
    "UpdateKinesisStreamingConfiguration": {
    "ApproximateCreationDateTimePrecision": "MILLISECOND"
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "StreamArn": "arn:aws:kinesis:us-east-1:123456789012:stream/example_stream",
    "TableName": "example_table",
    "DestinationStatus": "UPDATING",
    "UpdateKinesisStreamingConfiguration": {
        "ApproximateCreationDateTimePrecision": "MILLISECOND"
    }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateTable

Service: Amazon DynamoDB

Modifies the provisioned throughput settings, global secondary indexes, or DynamoDB Streams settings for a given table.

You can only perform one of the following operations at once:

- Modify the provisioned throughput settings of the table.
- Remove a global secondary index from the table.
- Create a new global secondary index on the table. After the index begins backfilling, you can use `UpdateTable` to perform other operations.

`UpdateTable` is an asynchronous operation; while it's executing, the table status changes from `ACTIVE` to `UPDATING`. While it's `UPDATING`, you can't issue another `UpdateTable` request. When the table returns to the `ACTIVE` state, the `UpdateTable` operation is complete.

Request Syntax

```
{
   "AttributeDefinitions": [
      {
         "AttributeName": "string",
         "AttributeType": "string"
      }
   ],
   "BillingMode": "string",
   "DeletionProtectionEnabled": boolean,
   "GlobalSecondaryIndexUpdates": [
      {
         "Create": {
            "IndexName": "string",
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "OnDemandThroughput": {
               "MaxReadRequestUnits": number,
               "MaxWriteRequestUnits": number
```

```
            },
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            },
            "ProvisionedThroughput": {
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "WriteUnitsPerSecond": number
            }
         },
         "Delete": {
            "IndexName": "string"
         },
         "Update": {
            "IndexName": "string",
            "OnDemandThroughput": {
               "MaxReadRequestUnits": number,
               "MaxWriteRequestUnits": number
            },
            "ProvisionedThroughput": {
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "WriteUnitsPerSecond": number
            }
         }
      }
   ],
   "GlobalTableWitnessUpdates": [
      {
         "Create": {
            "RegionName": "string"
         },
         "Delete": {
            "RegionName": "string"
         }
      }
   ],
```

```
    "MultiRegionConsistency": "string",
    "OnDemandThroughput": {
       "MaxReadRequestUnits": number,
       "MaxWriteRequestUnits": number
    },
    "ProvisionedThroughput": {
       "ReadCapacityUnits": number,
       "WriteCapacityUnits": number
    },
    "ReplicaUpdates": [
       {
          "Create": {
             "GlobalSecondaryIndexes": [
                {
                   "IndexName": "string",
                   "OnDemandThroughputOverride": {
                      "MaxReadRequestUnits": number
                   },
                   "ProvisionedThroughputOverride": {
                      "ReadCapacityUnits": number
                   }
                }
             ],
             "KMSMasterKeyId": "string",
             "OnDemandThroughputOverride": {
                "MaxReadRequestUnits": number
             },
             "ProvisionedThroughputOverride": {
                "ReadCapacityUnits": number
             },
             "RegionName": "string",
             "TableClassOverride": "string"
          },
          "Delete": {
             "RegionName": "string"
          },
          "Update": {
             "GlobalSecondaryIndexes": [
                {
                   "IndexName": "string",
                   "OnDemandThroughputOverride": {
                      "MaxReadRequestUnits": number
                   },
                   "ProvisionedThroughputOverride": {
```

```
                        "ReadCapacityUnits": number
                    }
                }
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
                "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
                "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "TableClassOverride": "string"
        }
    }
    ],
    "SSESpecification": {
        "Enabled": boolean,
        "KMSMasterKeyId": "string",
        "SSEType": "string"
    },
    "StreamSpecification": {
        "StreamEnabled": boolean,
        "StreamViewType": "string"
    },
    "TableClass": "string",
    "TableName": "string",
    "WarmThroughput": {
        "ReadUnitsPerSecond": number,
        "WriteUnitsPerSecond": number
    }
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table to be updated. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

AttributeDefinitions

An array of attributes that describe the key schema for the table and indexes. If you are adding a new global secondary index to the table, `AttributeDefinitions` must include the key element(s) of the new index.

Type: Array of AttributeDefinition objects

Required: No

BillingMode

Controls how you are charged for read and write throughput and how you manage capacity. When switching from pay-per-request to provisioned capacity, initial provisioned capacity values must be set. The initial provisioned capacity values are estimated based on the consumed read and write capacity of your table and global secondary indexes over the past 30 minutes.

- `PAY_PER_REQUEST` - We recommend using `PAY_PER_REQUEST` for most DynamoDB workloads. `PAY_PER_REQUEST` sets the billing mode to On-demand capacity mode.
- `PROVISIONED` - We recommend using `PROVISIONED` for steady workloads with predictable growth where capacity requirements can be reliably forecasted. `PROVISIONED` sets the billing mode to Provisioned capacity mode.

Type: String

Valid Values: `PROVISIONED | PAY_PER_REQUEST`

Required: No

DeletionProtectionEnabled

Indicates whether deletion protection is to be enabled (true) or disabled (false) on the table.

Type: Boolean

Required: No

GlobalSecondaryIndexUpdates

An array of one or more global secondary indexes for the table. For each index in the array, you can request one action:

- `Create` - add a new global secondary index to the table.
- `Update` - modify the provisioned throughput settings of an existing global secondary index.
- `Delete` - remove a global secondary index from the table.

You can create or delete only one global secondary index per `UpdateTable` operation.

For more information, see Managing Global Secondary Indexes in the *Amazon DynamoDB Developer Guide*.

Type: Array of GlobalSecondaryIndexUpdate objects

Required: No

GlobalTableWitnessUpdates

A list of witness updates for a MRSC global table. A witness provides a cost-effective alternative to a full replica in a MRSC global table by maintaining replicated change data written to global table replicas. You cannot perform read or write operations on a witness. For each witness, you can request one action:

- `Create` - add a new witness to the global table.
- `Delete` - remove a witness from the global table.

You can create or delete only one witness per `UpdateTable` operation.

For more information, see Multi-Region strong consistency (MRSC) in the Amazon DynamoDB Developer Guide

Type: Array of GlobalTableWitnessGroupUpdate objects

Array Members: Fixed number of 1 item.

Required: No

MultiRegionConsistency

Specifies the consistency mode for a new global table. This parameter is only valid when you create a global table by specifying one or more Create actions in the ReplicaUpdates action list.

You can specify one of the following consistency modes:

- `EVENTUAL`: Configures a new global table for multi-Region eventual consistency (MREC). This is the default consistency mode for global tables.
- `STRONG`: Configures a new global table for multi-Region strong consistency (MRSC).

If you don't specify this field, the global table consistency mode defaults to `EVENTUAL`. For more information about global tables consistency modes, see Consistency modes in DynamoDB developer guide.

Type: String

Valid Values: `EVENTUAL` | `STRONG`

Required: No

OnDemandThroughput

Updates the maximum number of read and write units for the specified table in on-demand capacity mode. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughput

The new provisioned throughput settings for the specified table or index.

Type: ProvisionedThroughput object

Required: No

ReplicaUpdates

A list of replica update actions (create, delete, or update) for the table.

Type: Array of ReplicationGroupUpdate objects

Array Members: Minimum number of 1 item.

Required: No

SSESpecification

The new server-side encryption settings for the specified table.

Type: SSESpecification object

Required: No

StreamSpecification

Represents the DynamoDB Streams configuration for the table.

> **Note**
>
> You receive a `ValidationException` if you try to enable a stream on a table that already has a stream, or if you try to disable a stream on a table that doesn't have a stream.

Type: StreamSpecification object

Required: No

TableClass

The table class of the table to be updated. Valid values are `STANDARD` and `STANDARD_INFREQUENT_ACCESS`.

Type: String

Valid Values: `STANDARD | STANDARD_INFREQUENT_ACCESS`

Required: No

WarmThroughput

Represents the warm throughput (in read units per second and write units per second) for updating a table.

Type: WarmThroughput object

Required: No

Response Syntax

```
{
   "TableDescription": {
      "ArchivalSummary": {
         "ArchivalBackupArn": "string",
         "ArchivalDateTime": number,
         "ArchivalReason": "string"
      },
      "AttributeDefinitions": [
         {
            "AttributeName": "string",
            "AttributeType": "string"
         }
      ],
      "BillingModeSummary": {
         "BillingMode": "string",
         "LastUpdateToPayPerRequestDateTime": number
      },
      "CreationDateTime": number,
      "DeletionProtectionEnabled": boolean,
      "GlobalSecondaryIndexes": [
         {
            "Backfilling": boolean,
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "IndexStatus": "string",
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "OnDemandThroughput": {
               "MaxReadRequestUnits": number,
               "MaxWriteRequestUnits": number
            },
            "Projection": {
               "NonKeyAttributes": [ "string" ],
               "ProjectionType": "string"
            },
```

```
            "ProvisionedThroughput": {
               "LastDecreaseDateTime": number,
               "LastIncreaseDateTime": number,
               "NumberOfDecreasesToday": number,
               "ReadCapacityUnits": number,
               "WriteCapacityUnits": number
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "GlobalTableVersion": "string",
      "GlobalTableWitnesses": [
         {
            "RegionName": "string",
            "WitnessStatus": "string"
         }
      ],
      "ItemCount": number,
      "KeySchema": [
         {
            "AttributeName": "string",
            "KeyType": "string"
         }
      ],
      "LatestStreamArn": "string",
      "LatestStreamLabel": "string",
      "LocalSecondaryIndexes": [
         {
            "IndexArn": "string",
            "IndexName": "string",
            "IndexSizeBytes": number,
            "ItemCount": number,
            "KeySchema": [
               {
                  "AttributeName": "string",
                  "KeyType": "string"
               }
            ],
            "Projection": {
               "NonKeyAttributes": [ "string" ],
```

```
                "ProjectionType": "string"
            }
        }
    ],
    "MultiRegionConsistency": "string",
    "OnDemandThroughput": {
        "MaxReadRequestUnits": number,
        "MaxWriteRequestUnits": number
    },
    "ProvisionedThroughput": {
        "LastDecreaseDateTime": number,
        "LastIncreaseDateTime": number,
        "NumberOfDecreasesToday": number,
        "ReadCapacityUnits": number,
        "WriteCapacityUnits": number
    },
    "Replicas": [
        {
            "GlobalSecondaryIndexes": [
                {
                    "IndexName": "string",
                    "OnDemandThroughputOverride": {
                        "MaxReadRequestUnits": number
                    },
                    "ProvisionedThroughputOverride": {
                        "ReadCapacityUnits": number
                    },
                    "WarmThroughput": {
                        "ReadUnitsPerSecond": number,
                        "Status": "string",
                        "WriteUnitsPerSecond": number
                    }
                }
            ],
            "KMSMasterKeyId": "string",
            "OnDemandThroughputOverride": {
                "MaxReadRequestUnits": number
            },
            "ProvisionedThroughputOverride": {
                "ReadCapacityUnits": number
            },
            "RegionName": "string",
            "ReplicaInaccessibleDateTime": number,
            "ReplicaStatus": "string",
```

```
            "ReplicaStatusDescription": "string",
            "ReplicaStatusPercentProgress": "string",
            "ReplicaTableClassSummary": {
               "LastUpdateDateTime": number,
               "TableClass": "string"
            },
            "WarmThroughput": {
               "ReadUnitsPerSecond": number,
               "Status": "string",
               "WriteUnitsPerSecond": number
            }
         }
      ],
      "RestoreSummary": {
         "RestoreDateTime": number,
         "RestoreInProgress": boolean,
         "SourceBackupArn": "string",
         "SourceTableArn": "string"
      },
      "SSEDescription": {
         "InaccessibleEncryptionDateTime": number,
         "KMSMasterKeyArn": "string",
         "SSEType": "string",
         "Status": "string"
      },
      "StreamSpecification": {
         "StreamEnabled": boolean,
         "StreamViewType": "string"
      },
      "TableArn": "string",
      "TableClassSummary": {
         "LastUpdateDateTime": number,
         "TableClass": "string"
      },
      "TableId": "string",
      "TableName": "string",
      "TableSizeBytes": number,
      "TableStatus": "string",
      "WarmThroughput": {
         "ReadUnitsPerSecond": number,
         "Status": "string",
         "WriteUnitsPerSecond": number
      }
   }
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TableDescription

Represents the properties of the table.

Type: TableDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Modify Provisioned Write Throughput

This example changes both the provisioned read and write throughput of the `Thread` table to 10 capacity units.

Sample Request

```
POST / HTTP/1.1
Host: dynamodb.<region>.<domain>;
Accept-Encoding: identity
Content-Length: <PayloadSizeBytes>
User-Agent: <UserAgentString>
Content-Type: application/x-amz-json-1.0
Authorization: AWS4-HMAC-SHA256 Credential=<Credential>, SignedHeaders=<Headers>,
 Signature=<Signature>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDB_20120810.UpdateTable
```

```
{
    "TableName": "Thread",
    "ProvisionedThroughput": {
        "ReadCapacityUnits": 10,
        "WriteCapacityUnits": 10
    }
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>
 {
    "TableDescription": {
        "TableArn": "arn:aws:dynamodb:us-west-2:123456789012:table/Thread",
        "AttributeDefinitions": [
            {
                "AttributeName": "ForumName",
                "AttributeType": "S"
            },
            {
                "AttributeName": "LastPostDateTime",
                "AttributeType": "S"
            },
            {
                "AttributeName": "Subject",
                "AttributeType": "S"
            }
        ],
        "CreationDateTime": 1.363801528686E9,
        "ItemCount": 0,
        "KeySchema": [
            {
                "AttributeName": "ForumName",
                "KeyType": "HASH"
            },
            {
                "AttributeName": "Subject",
```

```
                "KeyType": "RANGE"
            }
        ],
        "LocalSecondaryIndexes": [
            {
                "IndexName": "LastPostIndex",
                "IndexSizeBytes": 0,
                "ItemCount": 0,
                "KeySchema": [
                    {
                        "AttributeName": "ForumName",
                        "KeyType": "HASH"
                    },
                    {
                        "AttributeName": "LastPostDateTime",
                        "KeyType": "RANGE"
                    }
                ],
                "Projection": {
                    "ProjectionType": "KEYS_ONLY"
                }
            }
        ],
        "ProvisionedThroughput": {
            "LastIncreaseDateTime": 1.363801701282E9,
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableName": "Thread",
        "TableSizeBytes": 0,
        "TableStatus": "UPDATING"
    }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateTableReplicaAutoScaling

Service: Amazon DynamoDB

Updates auto scaling settings on your global tables at once.

Request Syntax

```
{
   "GlobalSecondaryIndexUpdates": [
      {
         "IndexName": "string",
         "ProvisionedWriteCapacityAutoScalingUpdate": {
            "AutoScalingDisabled": boolean,
            "AutoScalingRoleArn": "string",
            "MaximumUnits": number,
            "MinimumUnits": number,
            "ScalingPolicyUpdate": {
               "PolicyName": "string",
               "TargetTrackingScalingPolicyConfiguration": {
                  "DisableScaleIn": boolean,
                  "ScaleInCooldown": number,
                  "ScaleOutCooldown": number,
                  "TargetValue": number
               }
            }
         }
      }
   ],
   "ProvisionedWriteCapacityAutoScalingUpdate": {
      "AutoScalingDisabled": boolean,
      "AutoScalingRoleArn": "string",
      "MaximumUnits": number,
      "MinimumUnits": number,
      "ScalingPolicyUpdate": {
         "PolicyName": "string",
         "TargetTrackingScalingPolicyConfiguration": {
            "DisableScaleIn": boolean,
            "ScaleInCooldown": number,
            "ScaleOutCooldown": number,
            "TargetValue": number
         }
      }
   },
```

```
   "ReplicaUpdates": [
      {
         "RegionName": "string",
         "ReplicaGlobalSecondaryIndexUpdates": [
            {
               "IndexName": "string",
               "ProvisionedReadCapacityAutoScalingUpdate": {
                  "AutoScalingDisabled": boolean,
                  "AutoScalingRoleArn": "string",
                  "MaximumUnits": number,
                  "MinimumUnits": number,
                  "ScalingPolicyUpdate": {
                     "PolicyName": "string",
                     "TargetTrackingScalingPolicyConfiguration": {
                        "DisableScaleIn": boolean,
                        "ScaleInCooldown": number,
                        "ScaleOutCooldown": number,
                        "TargetValue": number
                     }
                  }
               }
            }
         ],
         "ReplicaProvisionedReadCapacityAutoScalingUpdate": {
            "AutoScalingDisabled": boolean,
            "AutoScalingRoleArn": "string",
            "MaximumUnits": number,
            "MinimumUnits": number,
            "ScalingPolicyUpdate": {
               "PolicyName": "string",
               "TargetTrackingScalingPolicyConfiguration": {
                  "DisableScaleIn": boolean,
                  "ScaleInCooldown": number,
                  "ScaleOutCooldown": number,
                  "TargetValue": number
               }
            }
         }
      }
   ],
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the global table to be updated. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

GlobalSecondaryIndexUpdates

Represents the auto scaling settings of the global secondary indexes of the replica to be updated.

Type: Array of GlobalSecondaryIndexAutoScalingUpdate objects

Array Members: Minimum number of 1 item.

Required: No

ProvisionedWriteCapacityAutoScalingUpdate

Represents the auto scaling settings to be modified for a global table or global secondary index.

Type: AutoScalingSettingsUpdate object

Required: No

ReplicaUpdates

Represents the auto scaling settings of replicas of the table that will be modified.

Type: Array of ReplicaAutoScalingUpdate objects

Array Members: Minimum number of 1 item.

Required: No

Response Syntax

```
{
   "TableAutoScalingDescription": {
      "Replicas": [
         {
            "GlobalSecondaryIndexes": [
               {
                  "IndexName": "string",
                  "IndexStatus": "string",
                  "ProvisionedReadCapacityAutoScalingSettings": {
                     "AutoScalingDisabled": boolean,
                     "AutoScalingRoleArn": "string",
                     "MaximumUnits": number,
                     "MinimumUnits": number,
                     "ScalingPolicies": [
                        {
                           "PolicyName": "string",
                           "TargetTrackingScalingPolicyConfiguration": {
                              "DisableScaleIn": boolean,
                              "ScaleInCooldown": number,
                              "ScaleOutCooldown": number,
                              "TargetValue": number
                           }
                        }
                     ]
                  },
                  "ProvisionedWriteCapacityAutoScalingSettings": {
                     "AutoScalingDisabled": boolean,
                     "AutoScalingRoleArn": "string",
                     "MaximumUnits": number,
                     "MinimumUnits": number,
                     "ScalingPolicies": [
                        {
                           "PolicyName": "string",
                           "TargetTrackingScalingPolicyConfiguration": {
                              "DisableScaleIn": boolean,
                              "ScaleInCooldown": number,
                              "ScaleOutCooldown": number,
                              "TargetValue": number
                           }
```

```
                        }
                    ]
                }
            }
        ],
        "RegionName": "string",
        "ReplicaProvisionedReadCapacityAutoScalingSettings": {
            "AutoScalingDisabled": boolean,
            "AutoScalingRoleArn": "string",
            "MaximumUnits": number,
            "MinimumUnits": number,
            "ScalingPolicies": [
                {
                    "PolicyName": "string",
                    "TargetTrackingScalingPolicyConfiguration": {
                        "DisableScaleIn": boolean,
                        "ScaleInCooldown": number,
                        "ScaleOutCooldown": number,
                        "TargetValue": number
                    }
                }
            ]
        },
        "ReplicaProvisionedWriteCapacityAutoScalingSettings": {
            "AutoScalingDisabled": boolean,
            "AutoScalingRoleArn": "string",
            "MaximumUnits": number,
            "MinimumUnits": number,
            "ScalingPolicies": [
                {
                    "PolicyName": "string",
                    "TargetTrackingScalingPolicyConfiguration": {
                        "DisableScaleIn": boolean,
                        "ScaleInCooldown": number,
                        "ScaleOutCooldown": number,
                        "TargetValue": number
                    }
                }
            ]
        },
        "ReplicaStatus": "string"
    }
],
"TableName": "string",
```

```
        "TableStatus": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TableAutoScalingDescription

Returns information about the auto scaling settings of a table with replicas.

Type: TableAutoScalingDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

UpdateTimeToLive

Service: Amazon DynamoDB

The `UpdateTimeToLive` method enables or disables Time to Live (TTL) for the specified table. A successful `UpdateTimeToLive` call returns the current `TimeToLiveSpecification`. It can take up to one hour for the change to fully process. Any additional `UpdateTimeToLive` calls for the same table during this one hour duration result in a `ValidationException`.

TTL compares the current time in epoch time format to the time stored in the TTL attribute of an item. If the epoch time value stored in the attribute is less than the current time, the item is marked as expired and subsequently deleted.

> **Note**
>
> The epoch time format is the number of seconds elapsed since 12:00:00 AM January 1, 1970 UTC.

DynamoDB deletes expired items on a best-effort basis to ensure availability of throughput for other data operations.

> **Important**
>
> DynamoDB typically deletes expired items within two days of expiration. The exact duration within which an item gets deleted after expiration is specific to the nature of the workload. Items that have expired and not been deleted will still show up in reads, queries, and scans.

As items are deleted, they are removed from any local secondary index and global secondary index immediately in the same eventually consistent way as a standard delete operation.

For more information, see Time To Live in the Amazon DynamoDB Developer Guide.

Request Syntax

```
{
   "TableName": "string",
   "TimeToLiveSpecification": {
```

```
        "AttributeName": "string",
        "Enabled": boolean
    }
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

TableName

The name of the table to be configured. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

TimeToLiveSpecification

Represents the settings used to enable or disable Time to Live for the specified table.

Type: TimeToLiveSpecification object

Required: Yes

Response Syntax

```
{
    "TimeToLiveSpecification": {
        "AttributeName": "string",
        "Enabled": boolean
    }
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

TimeToLiveSpecification

Represents the output of an `UpdateTimeToLive` operation.

Type: TimeToLiveSpecification object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceInUseException

The operation conflicts with the resource's availability. For example:

- You attempted to recreate an existing table.
- You tried to delete a table currently in the `CREATING` state.
- You tried to update a resource that was already being updated.

When appropriate, wait for the ongoing update to complete and attempt the request again.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DynamoDB Accelerator

The following actions are supported by DynamoDB Accelerator:

- CreateCluster
- CreateParameterGroup
- CreateSubnetGroup
- DecreaseReplicationFactor
- DeleteCluster
- DeleteParameterGroup
- DeleteSubnetGroup
- DescribeClusters
- DescribeDefaultParameters
- DescribeEvents
- DescribeParameterGroups
- DescribeParameters
- DescribeSubnetGroups
- IncreaseReplicationFactor
- ListTags
- RebootNode
- TagResource
- UntagResource
- UpdateCluster
- UpdateParameterGroup
- UpdateSubnetGroup

CreateCluster

Service: DynamoDB Accelerator

Creates a DAX cluster. All nodes in the cluster run the same DAX caching software.

Request Syntax

```
{
   "AvailabilityZones": [ "string" ],
   "ClusterEndpointEncryptionType": "string",
   "ClusterName": "string",
   "Description": "string",
   "IamRoleArn": "string",
   "NodeType": "string",
   "NotificationTopicArn": "string",
   "ParameterGroupName": "string",
   "PreferredMaintenanceWindow": "string",
   "ReplicationFactor": number,
   "SecurityGroupIds": [ "string" ],
   "SSESpecification": {
      "Enabled": boolean
   },
   "SubnetGroupName": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ClusterName

The cluster identifier. This parameter is stored as a lowercase string.

Constraints:

- A name must contain from 1 to 20 alphanumeric characters or hyphens.
- The first character must be a letter.
- A name cannot end with a hyphen or contain two consecutive hyphens.

Type: String

Required: Yes

IamRoleArn

A valid Amazon Resource Name (ARN) that identifies an IAM role. At runtime, DAX will assume this role and use the role's permissions to access DynamoDB on your behalf.

Type: String

Required: Yes

NodeType

The compute and memory capacity of the nodes in the cluster.

Type: String

Required: Yes

ReplicationFactor

The number of nodes in the DAX cluster. A replication factor of 1 will create a single-node cluster, without any read replicas. For additional fault tolerance, you can create a multiple node cluster with one or more read replicas. To do this, set `ReplicationFactor` to a number between 3 (one primary and two read replicas) and 10 (one primary and nine read replicas). `If the AvailabilityZones` parameter is provided, its length must equal the `ReplicationFactor`.

> **Note**
>
> AWS recommends that you have at least two read replicas per cluster.

Type: Integer

Required: Yes

AvailabilityZones

The Availability Zones (AZs) in which the cluster nodes will reside after the cluster has been created or updated. If provided, the length of this list must equal the `ReplicationFactor` parameter. If you omit this parameter, DAX will spread the nodes across Availability Zones for the highest availability.

Type: Array of strings

Required: No

ClusterEndpointEncryptionType

The type of encryption the cluster's endpoint should support. Values are:

- `NONE` for no encryption
- `TLS` for Transport Layer Security

Type: String

Valid Values: `NONE` | `TLS`

Required: No

Description

A description of the cluster.

Type: String

Required: No

NotificationTopicArn

The Amazon Resource Name (ARN) of the Amazon SNS topic to which notifications will be sent.

> **Note**
>
> The Amazon SNS topic owner must be same as the DAX cluster owner.

Type: String

Required: No

ParameterGroupName

The parameter group to be associated with the DAX cluster.

Type: String

Required: No

PreferredMaintenanceWindow

Specifies the weekly time range during which maintenance on the DAX cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are:

- `sun`
- `mon`
- `tue`
- `wed`
- `thu`
- `fri`
- `sat`

Example: `sun:05:00-sun:09:00`

> **Note**
>
> If you don't specify a preferred maintenance window when you create or modify a cache cluster, DAX assigns a 60-minute maintenance window on a randomly selected day of the week.

Type: String

Required: No

SecurityGroupIds

A list of security group IDs to be assigned to each node in the DAX cluster. (Each of the security group ID is system-generated.)

If this parameter is not specified, DAX assigns the default VPC security group to each node.

Type: Array of strings

Required: No

SSESpecification

Represents the settings used to enable server-side encryption on the cluster.

Type: SSESpecification object

Required: No

SubnetGroupName

The name of the subnet group to be used for the replication group.

> **Important**
>
> DAX clusters can only run in an Amazon VPC environment. All of the subnets that you specify in a subnet group must exist in the same VPC.

Type: String

Required: No

Tags

A set of tags to associate with the DAX cluster.

Type: Array of Tag objects

Required: No

Response Syntax

```
{
   "Cluster": {
      "ActiveNodes": number,
      "ClusterArn": "string",
      "ClusterDiscoveryEndpoint": {
         "Address": "string",
         "Port": number,
         "URL": "string"
      },
```

```
        "ClusterEndpointEncryptionType": "string",
        "ClusterName": "string",
        "Description": "string",
        "IamRoleArn": "string",
        "NodeIdsToRemove": [ "string" ],
        "Nodes": [
           {
              "AvailabilityZone": "string",
              "Endpoint": {
                 "Address": "string",
                 "Port": number,
                 "URL": "string"
              },
              "NodeCreateTime": number,
              "NodeId": "string",
              "NodeStatus": "string",
              "ParameterGroupStatus": "string"
           }
        ],
        "NodeType": "string",
        "NotificationConfiguration": {
           "TopicArn": "string",
           "TopicStatus": "string"
        },
        "ParameterGroup": {
           "NodeIdsToReboot": [ "string" ],
           "ParameterApplyStatus": "string",
           "ParameterGroupName": "string"
        },
        "PreferredMaintenanceWindow": "string",
        "SecurityGroups": [
           {
              "SecurityGroupIdentifier": "string",
              "Status": "string"
           }
        ],
        "SSEDescription": {
           "Status": "string"
        },
        "Status": "string",
        "SubnetGroup": "string",
        "TotalNodes": number
    }
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Cluster

A description of the DAX cluster that you have created.

Type: Cluster object

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterAlreadyExistsFault

You already have a DAX cluster with the given identifier.

HTTP Status Code: 400

ClusterQuotaForCustomerExceededFault

You have attempted to exceed the maximum number of DAX clusters for your AWS account.

HTTP Status Code: 400

InsufficientClusterCapacityFault

There are not enough system resources to create the cluster you requested (or to resize an already-existing cluster).

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterGroupStateFault

One or more parameters in a parameter group are in an invalid state.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

The VPC network is in an invalid state.

HTTP Status Code: 400

NodeQuotaForClusterExceededFault

You have attempted to exceed the maximum number of nodes for a DAX cluster.

HTTP Status Code: 400

NodeQuotaForCustomerExceededFault

You have attempted to exceed the maximum number of nodes for your AWS account.

HTTP Status Code: 400

ParameterGroupNotFoundFault

The specified parameter group does not exist.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

ServiceQuotaExceededException

You have reached the maximum number of x509 certificates that can be created for encrypted clusters in a 30 day period. Contact AWS customer support to discuss options for continuing to create encrypted clusters.

HTTP Status Code: 400

SubnetGroupNotFoundFault

The requested subnet group name does not refer to an existing subnet group.

HTTP Status Code: 400

TagQuotaPerResourceExceeded

You have exceeded the maximum number of tags for this DAX cluster.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateParameterGroup

Service: DynamoDB Accelerator

Creates a new parameter group. A parameter group is a collection of parameters that you apply to all of the nodes in a DAX cluster.

Request Syntax

```
{
   "Description": "string",
   "ParameterGroupName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ParameterGroupName

The name of the parameter group to apply to all of the clusters in this replication group.

Type: String

Required: Yes

Description

A description of the parameter group.

Type: String

Required: No

Response Syntax

```
{
```

```
    "ParameterGroup": {
       "Description": "string",
       "ParameterGroupName": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ParameterGroup

Represents the output of a *CreateParameterGroup* action.

Type: ParameterGroup object

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterGroupStateFault

One or more parameters in a parameter group are in an invalid state.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ParameterGroupAlreadyExistsFault

The specified parameter group already exists.

HTTP Status Code: 400

ParameterGroupQuotaExceededFault

You have attempted to exceed the maximum number of parameter groups.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateSubnetGroup

Service: DynamoDB Accelerator

Creates a new subnet group.

Request Syntax

```
{
   "Description": "string",
   "SubnetGroupName": "string",
   "SubnetIds": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

SubnetGroupName

A name for the subnet group. This value is stored as a lowercase string.

Type: String

Required: Yes

SubnetIds

A list of VPC subnet IDs for the subnet group.

Type: Array of strings

Required: Yes

Description

A description for the subnet group

Type: String

Required: No

Response Syntax

```
{
   "SubnetGroup": {
      "Description": "string",
      "SubnetGroupName": "string",
      "Subnets": [
         {
            "SubnetAvailabilityZone": "string",
            "SubnetIdentifier": "string"
         }
      ],
      "VpcId": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

SubnetGroup

Represents the output of a *CreateSubnetGroup* operation.

Type: SubnetGroup object

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidSubnet

An invalid subnet identifier was specified.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

SubnetGroupAlreadyExistsFault

The specified subnet group already exists.

HTTP Status Code: 400

SubnetGroupQuotaExceededFault

The request cannot be processed because it would exceed the allowed number of subnets in a subnet group.

HTTP Status Code: 400

SubnetQuotaExceededFault

The request cannot be processed because it would exceed the allowed number of subnets in a subnet group.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python

- AWS SDK for Ruby V3

DecreaseReplicationFactor

Service: DynamoDB Accelerator

Removes one or more nodes from a DAX cluster.

> **Note**
>
> You cannot use `DecreaseReplicationFactor` to remove the last node in a DAX cluster. If you need to do this, use `DeleteCluster` instead.

Request Syntax

```
{
   "AvailabilityZones": [ "string" ],
   "ClusterName": "string",
   "NewReplicationFactor": number,
   "NodeIdsToRemove": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ClusterName

The name of the DAX cluster from which you want to remove nodes.

Type: String

Required: Yes

NewReplicationFactor

The new number of nodes for the DAX cluster.

Type: Integer

Required: Yes

AvailabilityZones

The Availability Zone(s) from which to remove nodes.

Type: Array of strings

Required: No

NodeIdsToRemove

The unique identifiers of the nodes to be removed from the cluster.

Type: Array of strings

Required: No

Response Syntax

```
{
   "Cluster": {
      "ActiveNodes": number,
      "ClusterArn": "string",
      "ClusterDiscoveryEndpoint": {
         "Address": "string",
         "Port": number,
         "URL": "string"
      },
      "ClusterEndpointEncryptionType": "string",
      "ClusterName": "string",
      "Description": "string",
      "IamRoleArn": "string",
      "NodeIdsToRemove": [ "string" ],
      "Nodes": [
         {
            "AvailabilityZone": "string",
            "Endpoint": {
               "Address": "string",
               "Port": number,
               "URL": "string"
```

```
            },
            "NodeCreateTime": number,
            "NodeId": "string",
            "NodeStatus": "string",
            "ParameterGroupStatus": "string"
         }
      ],
      "NodeType": "string",
      "NotificationConfiguration": {
         "TopicArn": "string",
         "TopicStatus": "string"
      },
      "ParameterGroup": {
         "NodeIdsToReboot": [ "string" ],
         "ParameterApplyStatus": "string",
         "ParameterGroupName": "string"
      },
      "PreferredMaintenanceWindow": "string",
      "SecurityGroups": [
         {
            "SecurityGroupIdentifier": "string",
            "Status": "string"
         }
      ],
      "SSEDescription": {
         "Status": "string"
      },
      "Status": "string",
      "SubnetGroup": "string",
      "TotalNodes": number
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Cluster

A description of the DAX cluster, after you have decreased its replication factor.

Type: Cluster object

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

NodeNotFoundFault

None of the nodes in the cluster have the given node ID.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface

- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteCluster

Service: DynamoDB Accelerator

Deletes a previously provisioned DAX cluster. *DeleteCluster* deletes all associated nodes, node endpoints and the DAX cluster itself. When you receive a successful response from this action, DAX immediately begins deleting the cluster; you cannot cancel or revert this action.

Request Syntax

```
{
   "ClusterName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ClusterName

The name of the cluster to be deleted.

Type: String

Required: Yes

Response Syntax

```
{
   "Cluster": {
      "ActiveNodes": number,
      "ClusterArn": "string",
      "ClusterDiscoveryEndpoint": {
         "Address": "string",
         "Port": number,
         "URL": "string"
      },
```

```
        "ClusterEndpointEncryptionType": "string",
        "ClusterName": "string",
        "Description": "string",
        "IamRoleArn": "string",
        "NodeIdsToRemove": [ "string" ],
        "Nodes": [
           {
              "AvailabilityZone": "string",
              "Endpoint": {
                 "Address": "string",
                 "Port": number,
                 "URL": "string"
              },
              "NodeCreateTime": number,
              "NodeId": "string",
              "NodeStatus": "string",
              "ParameterGroupStatus": "string"
           }
        ],
        "NodeType": "string",
        "NotificationConfiguration": {
           "TopicArn": "string",
           "TopicStatus": "string"
        },
        "ParameterGroup": {
           "NodeIdsToReboot": [ "string" ],
           "ParameterApplyStatus": "string",
           "ParameterGroupName": "string"
        },
        "PreferredMaintenanceWindow": "string",
        "SecurityGroups": [
           {
              "SecurityGroupIdentifier": "string",
              "Status": "string"
           }
        ],
        "SSEDescription": {
           "Status": "string"
        },
        "Status": "string",
        "SubnetGroup": "string",
        "TotalNodes": number
    }
```

```
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Cluster

A description of the DAX cluster that is being deleted.

Type: Cluster object

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteParameterGroup

Service: DynamoDB Accelerator

Deletes the specified parameter group. You cannot delete a parameter group if it is associated with any DAX clusters.

Request Syntax

```
{
   "ParameterGroupName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ParameterGroupName

The name of the parameter group to delete.

Type: String

Required: Yes

Response Syntax

```
{
   "DeletionMessage": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DeletionMessage

A user-specified message for this action (i.e., a reason for deleting the parameter group).

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterGroupStateFault

One or more parameters in a parameter group are in an invalid state.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ParameterGroupNotFoundFault

The specified parameter group does not exist.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteSubnetGroup

Service: DynamoDB Accelerator

Deletes a subnet group.

> **Note**
>
> You cannot delete a subnet group if it is associated with any DAX clusters.

Request Syntax

```
{
   "SubnetGroupName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

SubnetGroupName

The name of the subnet group to delete.

Type: String

Required: Yes

Response Syntax

```
{
   "DeletionMessage": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

DeletionMessage

A user-specified message for this action (i.e., a reason for deleting the subnet group).

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

SubnetGroupInUseFault

The specified subnet group is currently in use.

HTTP Status Code: 400

SubnetGroupNotFoundFault

The requested subnet group name does not refer to an existing subnet group.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeClusters

Service: DynamoDB Accelerator

Returns information about all provisioned DAX clusters if no cluster identifier is specified, or about a specific DAX cluster if a cluster identifier is supplied.

If the cluster is in the CREATING state, only cluster level information will be displayed until all of the nodes are successfully provisioned.

If the cluster is in the DELETING state, only cluster level information will be displayed.

If nodes are currently being added to the DAX cluster, node endpoint information and creation time for the additional nodes will not be displayed until they are completely provisioned. When the DAX cluster state is *available*, the cluster is ready for use.

If nodes are currently being removed from the DAX cluster, no endpoint information for the removed nodes is displayed.

Request Syntax

```
{
   "ClusterNames": [ "string" ],
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ClusterNames

The names of the DAX clusters being described.

Type: Array of strings

Required: No

MaxResults

The maximum number of results to include in the response. If more results exist than the specified `MaxResults` value, a token is included in the response so that the remaining results can be retrieved.

The value for `MaxResults` must be between 20 and 100.

Type: Integer

Required: No

NextToken

An optional token returned from a prior request. Use this token for pagination of results from this action. If this parameter is specified, the response includes only results beyond the token, up to the value specified by `MaxResults`.

Type: String

Required: No

Response Syntax

```
{
   "Clusters": [
      {
         "ActiveNodes": number,
         "ClusterArn": "string",
         "ClusterDiscoveryEndpoint": {
            "Address": "string",
            "Port": number,
            "URL": "string"
         },
         "ClusterEndpointEncryptionType": "string",
         "ClusterName": "string",
         "Description": "string",
         "IamRoleArn": "string",
         "NodeIdsToRemove": [ "string" ],
         "Nodes": [
            {
               "AvailabilityZone": "string",
```

```
                "Endpoint": {
                   "Address": "string",
                   "Port": number,
                   "URL": "string"
                },
                "NodeCreateTime": number,
                "NodeId": "string",
                "NodeStatus": "string",
                "ParameterGroupStatus": "string"
             }
          ],
          "NodeType": "string",
          "NotificationConfiguration": {
             "TopicArn": "string",
             "TopicStatus": "string"
          },
          "ParameterGroup": {
             "NodeIdsToReboot": [ "string" ],
             "ParameterApplyStatus": "string",
             "ParameterGroupName": "string"
          },
          "PreferredMaintenanceWindow": "string",
          "SecurityGroups": [
             {
                "SecurityGroupIdentifier": "string",
                "Status": "string"
             }
          ],
          "SSEDescription": {
             "Status": "string"
          },
          "Status": "string",
          "SubnetGroup": "string",
          "TotalNodes": number
       }
    ],
    "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Clusters

The descriptions of your DAX clusters, in response to a *DescribeClusters* request.

Type: Array of Cluster objects

NextToken

Provides an identifier to allow retrieval of paginated results.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeDefaultParameters

Service: DynamoDB Accelerator

Returns the default system parameter information for the DAX caching software.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

MaxResults

The maximum number of results to include in the response. If more results exist than the specified `MaxResults` value, a token is included in the response so that the remaining results can be retrieved.

The value for `MaxResults` must be between 20 and 100.

Type: Integer

Required: No

NextToken

An optional token returned from a prior request. Use this token for pagination of results from this action. If this parameter is specified, the response includes only results beyond the token, up to the value specified by `MaxResults`.

Type: String

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "Parameters": [
      {
         "AllowedValues": "string",
         "ChangeType": "string",
         "DataType": "string",
         "Description": "string",
         "IsModifiable": "string",
         "NodeTypeSpecificValues": [
            {
               "NodeType": "string",
               "Value": "string"
            }
         ],
         "ParameterName": "string",
         "ParameterType": "string",
         "ParameterValue": "string",
         "Source": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

Provides an identifier to allow retrieval of paginated results.

Type: String

Parameters

A list of parameters. Each element in the list represents one parameter.

Type: Array of Parameter objects

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeEvents

Service: DynamoDB Accelerator

Returns events related to DAX clusters and parameter groups. You can obtain events specific to a particular DAX cluster or parameter group by providing the name as a parameter.

By default, only the events occurring within the last 24 hours are returned; however, you can retrieve up to 14 days' worth of events if necessary.

Request Syntax

```
{
   "Duration": number,
   "EndTime": number,
   "MaxResults": number,
   "NextToken": "string",
   "SourceName": "string",
   "SourceType": "string",
   "StartTime": number
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

Duration

The number of minutes' worth of events to retrieve.

Type: Integer

Required: No

EndTime

The end of the time interval for which to retrieve events, specified in ISO 8601 format.

Type: Timestamp

Required: No

MaxResults

The maximum number of results to include in the response. If more results exist than the specified `MaxResults` value, a token is included in the response so that the remaining results can be retrieved.

The value for `MaxResults` must be between 20 and 100.

Type: Integer

Required: No

NextToken

An optional token returned from a prior request. Use this token for pagination of results from this action. If this parameter is specified, the response includes only results beyond the token, up to the value specified by `MaxResults`.

Type: String

Required: No

SourceName

The identifier of the event source for which events will be returned. If not specified, then all sources are included in the response.

Type: String

Required: No

SourceType

The event source to retrieve events for. If no value is specified, all events are returned.

Type: String

Valid Values: `CLUSTER | PARAMETER_GROUP | SUBNET_GROUP`

Required: No

StartTime

The beginning of the time interval to retrieve events for, specified in ISO 8601 format.

Type: Timestamp

Required: No

Response Syntax

```
{
   "Events": [
      {
         "Date": number,
         "Message": "string",
         "SourceName": "string",
         "SourceType": "string"
      }
   ],
   "NextToken": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Events

An array of events. Each element in the array represents one event.

Type: Array of Event objects

NextToken

Provides an identifier to allow retrieval of paginated results.

Type: String

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeParameterGroups

Service: DynamoDB Accelerator

Returns a list of parameter group descriptions. If a parameter group name is specified, the list will contain only the descriptions for that group.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ParameterGroupNames": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

MaxResults

The maximum number of results to include in the response. If more results exist than the specified `MaxResults` value, a token is included in the response so that the remaining results can be retrieved.

The value for `MaxResults` must be between 20 and 100.

Type: Integer

Required: No

NextToken

An optional token returned from a prior request. Use this token for pagination of results from this action. If this parameter is specified, the response includes only results beyond the token, up to the value specified by `MaxResults`.

Type: String

Required: No

ParameterGroupNames

The names of the parameter groups.

Type: Array of strings

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "ParameterGroups": [
      {
         "Description": "string",
         "ParameterGroupName": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

Provides an identifier to allow retrieval of paginated results.

Type: String

ParameterGroups

An array of parameter groups. Each element in the array represents one parameter group.

Type: Array of ParameterGroup objects

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ParameterGroupNotFoundFault

The specified parameter group does not exist.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeParameters

Service: DynamoDB Accelerator

Returns the detailed parameter list for a particular parameter group.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ParameterGroupName": "string",
   "Source": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ParameterGroupName

The name of the parameter group.

Type: String

Required: Yes

MaxResults

The maximum number of results to include in the response. If more results exist than the specified `MaxResults` value, a token is included in the response so that the remaining results can be retrieved.

The value for `MaxResults` must be between 20 and 100.

Type: Integer

Required: No

NextToken

An optional token returned from a prior request. Use this token for pagination of results from this action. If this parameter is specified, the response includes only results beyond the token, up to the value specified by `MaxResults`.

Type: String

Required: No

Source

How the parameter is defined. For example, `system` denotes a system-defined parameter.

Type: String

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "Parameters": [
      {
         "AllowedValues": "string",
         "ChangeType": "string",
         "DataType": "string",
         "Description": "string",
         "IsModifiable": "string",
         "NodeTypeSpecificValues": [
            {
               "NodeType": "string",
               "Value": "string"
            }
         ],
         "ParameterName": "string",
         "ParameterType": "string",
         "ParameterValue": "string",
         "Source": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

Provides an identifier to allow retrieval of paginated results.

Type: String

Parameters

A list of parameters within a parameter group. Each element in the list represents one parameter.

Type: Array of Parameter objects

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ParameterGroupNotFoundFault

The specified parameter group does not exist.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeSubnetGroups

Service: DynamoDB Accelerator

Returns a list of subnet group descriptions. If a subnet group name is specified, the list will contain only the description of that group.

Request Syntax

```
{
   "MaxResults": number,
   "NextToken": "string",
   "SubnetGroupNames": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

MaxResults

The maximum number of results to include in the response. If more results exist than the specified `MaxResults` value, a token is included in the response so that the remaining results can be retrieved.

The value for `MaxResults` must be between 20 and 100.

Type: Integer

Required: No

NextToken

An optional token returned from a prior request. Use this token for pagination of results from this action. If this parameter is specified, the response includes only results beyond the token, up to the value specified by `MaxResults`.

Type: String

Required: No

SubnetGroupNames

The name of the subnet group.

Type: Array of strings

Required: No

Response Syntax

```
{
   "NextToken": "string",
   "SubnetGroups": [
      {
         "Description": "string",
         "SubnetGroupName": "string",
         "Subnets": [
            {
               "SubnetAvailabilityZone": "string",
               "SubnetIdentifier": "string"
            }
         ],
         "VpcId": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

Provides an identifier to allow retrieval of paginated results.

Type: String

SubnetGroups

An array of subnet groups. Each element in the array represents a single subnet group.

Type: Array of SubnetGroup objects

Errors

For information about the errors that are common to all actions, see Common Errors.

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

SubnetGroupNotFoundFault

The requested subnet group name does not refer to an existing subnet group.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

IncreaseReplicationFactor

Service: DynamoDB Accelerator

Adds one or more nodes to a DAX cluster.

Request Syntax

```
{
   "AvailabilityZones": [ "string" ],
   "ClusterName": "string",
   "NewReplicationFactor": number
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ClusterName

The name of the DAX cluster that will receive additional nodes.

Type: String

Required: Yes

NewReplicationFactor

The new number of nodes for the DAX cluster.

Type: Integer

Required: Yes

AvailabilityZones

The Availability Zones (AZs) in which the cluster nodes will be created. All nodes belonging to the cluster are placed in these Availability Zones. Use this parameter if you want to distribute the nodes across multiple AZs.

Type: Array of strings

Required: No

Response Syntax

```
{
   "Cluster": {
      "ActiveNodes": number,
      "ClusterArn": "string",
      "ClusterDiscoveryEndpoint": {
         "Address": "string",
         "Port": number,
         "URL": "string"
      },
      "ClusterEndpointEncryptionType": "string",
      "ClusterName": "string",
      "Description": "string",
      "IamRoleArn": "string",
      "NodeIdsToRemove": [ "string" ],
      "Nodes": [
         {
            "AvailabilityZone": "string",
            "Endpoint": {
               "Address": "string",
               "Port": number,
               "URL": "string"
            },
            "NodeCreateTime": number,
            "NodeId": "string",
            "NodeStatus": "string",
            "ParameterGroupStatus": "string"
         }
      ],
      "NodeType": "string",
      "NotificationConfiguration": {
         "TopicArn": "string",
         "TopicStatus": "string"
      },
      "ParameterGroup": {
         "NodeIdsToReboot": [ "string" ],
         "ParameterApplyStatus": "string",
         "ParameterGroupName": "string"
```

```
        },
        "PreferredMaintenanceWindow": "string",
        "SecurityGroups": [
           {
              "SecurityGroupIdentifier": "string",
              "Status": "string"
           }
        ],
        "SSEDescription": {
           "Status": "string"
        },
        "Status": "string",
        "SubnetGroup": "string",
        "TotalNodes": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Cluster

A description of the DAX cluster, with its new replication factor.

Type: Cluster object

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InsufficientClusterCapacityFault

There are not enough system resources to create the cluster you requested (or to resize an already-existing cluster).

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

InvalidVPCNetworkStateFault

The VPC network is in an invalid state.

HTTP Status Code: 400

NodeQuotaForClusterExceededFault

You have attempted to exceed the maximum number of nodes for a DAX cluster.

HTTP Status Code: 400

NodeQuotaForCustomerExceededFault

You have attempted to exceed the maximum number of nodes for your AWS account.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTags

Service: DynamoDB Accelerator

List all of the tags for a DAX cluster. You can call `ListTags` up to 10 times per second, per account.

Request Syntax

```
{
   "NextToken": "string",
   "ResourceName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ResourceName

The name of the DAX resource to which the tags belong.

Type: String

Required: Yes

NextToken

An optional token returned from a prior request. Use this token for pagination of results from this action. If this parameter is specified, the response includes only results beyond the token.

Type: String

Required: No

Response Syntax

```
{
```

```
    "NextToken": "string",
    "Tags": [
       {
          "Key": "string",
          "Value": "string"
       }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextToken

If this value is present, there are additional results to be displayed. To retrieve them, call `ListTags` again, with `NextToken` set to this value.

Type: String

Tags

A list of tags currently associated with the DAX cluster.

Type: Array of Tag objects

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InvalidARNFault

The Amazon Resource Name (ARN) supplied in the request is not valid.

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

RebootNode

Service: DynamoDB Accelerator

Reboots a single node of a DAX cluster. The reboot action takes place as soon as possible. During the reboot, the node status is set to REBOOTING.

> **Note**
>
> RebootNode restarts the DAX engine process and does not remove the contents of the cache.

Request Syntax

```
{
   "ClusterName": "string",
   "NodeId": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ClusterName

The name of the DAX cluster containing the node to be rebooted.

Type: String

Required: Yes

NodeId

The system-assigned ID of the node to be rebooted.

Type: String

Required: Yes

Response Syntax

```
{
   "Cluster": {
      "ActiveNodes": number,
      "ClusterArn": "string",
      "ClusterDiscoveryEndpoint": {
         "Address": "string",
         "Port": number,
         "URL": "string"
      },
      "ClusterEndpointEncryptionType": "string",
      "ClusterName": "string",
      "Description": "string",
      "IamRoleArn": "string",
      "NodeIdsToRemove": [ "string" ],
      "Nodes": [
         {
            "AvailabilityZone": "string",
            "Endpoint": {
               "Address": "string",
               "Port": number,
               "URL": "string"
            },
            "NodeCreateTime": number,
            "NodeId": "string",
            "NodeStatus": "string",
            "ParameterGroupStatus": "string"
         }
      ],
      "NodeType": "string",
      "NotificationConfiguration": {
         "TopicArn": "string",
         "TopicStatus": "string"
      },
      "ParameterGroup": {
         "NodeIdsToReboot": [ "string" ],
         "ParameterApplyStatus": "string",
         "ParameterGroupName": "string"
      },
      "PreferredMaintenanceWindow": "string",
```

```
        "SecurityGroups": [
           {
              "SecurityGroupIdentifier": "string",
              "Status": "string"
           }
        ],
        "SSEDescription": {
           "Status": "string"
        },
        "Status": "string",
        "SubnetGroup": "string",
        "TotalNodes": number
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Cluster

A description of the DAX cluster after a node has been rebooted.

Type: Cluster object

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

NodeNotFoundFault

None of the nodes in the cluster have the given node ID.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagResource

Service: DynamoDB Accelerator

Associates a set of tags with a DAX resource. You can call `TagResource` up to 5 times per second, per account.

Request Syntax

```
{
   "ResourceName": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ResourceName

The name of the DAX resource to which tags should be added.

Type: String

Required: Yes

Tags

The tags to be assigned to the DAX resource.

Type: Array of Tag objects

Required: Yes

Response Syntax

```
{
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Tags

The list of tags that are associated with the DAX resource.

Type: Array of Tag objects

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InvalidARNFault

The Amazon Resource Name (ARN) supplied in the request is not valid.

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

TagQuotaPerResourceExceeded

You have exceeded the maximum number of tags for this DAX cluster.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UntagResource

Service: DynamoDB Accelerator

Removes the association of tags from a DAX resource. You can call `UntagResource` up to 5 times per second, per account.

Request Syntax

```
{
   "ResourceName": "string",
   "TagKeys": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ResourceName

The name of the DAX resource from which the tags should be removed.

Type: String

Required: Yes

TagKeys

A list of tag keys. If the DAX cluster has any tags with these keys, then the tags are removed from the cluster.

Type: Array of strings

Required: Yes

Response Syntax

```
{
```

```
    "Tags": [
       {
          "Key": "string",
          "Value": "string"
       }
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Tags

The tag keys that have been removed from the cluster.

Type: Array of Tag objects

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InvalidARNFault

The Amazon Resource Name (ARN) supplied in the request is not valid.

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

TagNotFoundFault

The tag does not exist.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateCluster

Service: DynamoDB Accelerator

Modifies the settings for a DAX cluster. You can use this action to change one or more cluster configuration parameters by specifying the parameters and the new values.

Request Syntax

```
{
   "ClusterName": "string",
   "Description": "string",
   "NotificationTopicArn": "string",
   "NotificationTopicStatus": "string",
   "ParameterGroupName": "string",
   "PreferredMaintenanceWindow": "string",
   "SecurityGroupIds": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ClusterName

The name of the DAX cluster to be modified.

Type: String

Required: Yes

Description

A description of the changes being made to the cluster.

Type: String

Required: No

NotificationTopicArn

The Amazon Resource Name (ARN) that identifies the topic.

Type: String

Required: No

NotificationTopicStatus

The current state of the topic. A value of “active” means that notifications will be sent to the topic. A value of “inactive” means that notifications will not be sent to the topic.

Type: String

Required: No

ParameterGroupName

The name of a parameter group for this cluster.

Type: String

Required: No

PreferredMaintenanceWindow

A range of time when maintenance of DAX cluster software will be performed. For example: `sun:01:00-sun:09:00`. Cluster maintenance normally takes less than 30 minutes, and is performed automatically within the maintenance window.

Type: String

Required: No

SecurityGroupIds

A list of user-specified security group IDs to be assigned to each node in the DAX cluster. If this parameter is not specified, DAX assigns the default VPC security group to each node.

Type: Array of strings

Required: No

Response Syntax

```
{
   "Cluster": {
      "ActiveNodes": number,
      "ClusterArn": "string",
      "ClusterDiscoveryEndpoint": {
         "Address": "string",
         "Port": number,
         "URL": "string"
      },
      "ClusterEndpointEncryptionType": "string",
      "ClusterName": "string",
      "Description": "string",
      "IamRoleArn": "string",
      "NodeIdsToRemove": [ "string" ],
      "Nodes": [
         {
            "AvailabilityZone": "string",
            "Endpoint": {
               "Address": "string",
               "Port": number,
               "URL": "string"
            },
            "NodeCreateTime": number,
            "NodeId": "string",
            "NodeStatus": "string",
            "ParameterGroupStatus": "string"
         }
      ],
      "NodeType": "string",
      "NotificationConfiguration": {
         "TopicArn": "string",
         "TopicStatus": "string"
      },
      "ParameterGroup": {
         "NodeIdsToReboot": [ "string" ],
         "ParameterApplyStatus": "string",
         "ParameterGroupName": "string"
      },
      "PreferredMaintenanceWindow": "string",
      "SecurityGroups": [
         {
```

```
            "SecurityGroupIdentifier": "string",
            "Status": "string"
         }
      ],
      "SSEDescription": {
         "Status": "string"
      },
      "Status": "string",
      "SubnetGroup": "string",
      "TotalNodes": number
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

Cluster

A description of the DAX cluster, after it has been modified.

Type: Cluster object

Errors

For information about the errors that are common to all actions, see Common Errors.

ClusterNotFoundFault

The requested cluster ID does not refer to an existing DAX cluster.

HTTP Status Code: 400

InvalidClusterStateFault

The requested DAX cluster is not in the *available* state.

HTTP Status Code: 400

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterGroupStateFault

One or more parameters in a parameter group are in an invalid state.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ParameterGroupNotFoundFault

The specified parameter group does not exist.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateParameterGroup

Service: DynamoDB Accelerator

Modifies the parameters of a parameter group. You can modify up to 20 parameters in a single request by submitting a list parameter name and value pairs.

Request Syntax

```
{
   "ParameterGroupName": "string",
   "ParameterNameValues": [
      {
         "ParameterName": "string",
         "ParameterValue": "string"
      }
   ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ParameterGroupName

The name of the parameter group.

Type: String

Required: Yes

ParameterNameValues

An array of name-value pairs for the parameters in the group. Each element in the array represents a single parameter.

> **Note**
>
> `record-ttl-millis` and `query-ttl-millis` are the only supported parameter names. For more details, see Configuring TTL Settings.

Type: Array of ParameterNameValue objects

Required: Yes

Response Syntax

```
{
   "ParameterGroup": {
      "Description": "string",
      "ParameterGroupName": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ParameterGroup

The parameter group that has been modified.

Type: ParameterGroup object

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidParameterCombinationException

Two or more incompatible parameters were specified.

HTTP Status Code: 400

InvalidParameterGroupStateFault

One or more parameters in a parameter group are in an invalid state.

HTTP Status Code: 400

InvalidParameterValueException

The value for a parameter is invalid.

HTTP Status Code: 400

ParameterGroupNotFoundFault

The specified parameter group does not exist.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateSubnetGroup

Service: DynamoDB Accelerator

Modifies an existing subnet group.

Request Syntax

```
{
   "Description": "string",
   "SubnetGroupName": "string",
   "SubnetIds": [ "string" ]
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

SubnetGroupName

The name of the subnet group.

Type: String

Required: Yes

Description

A description of the subnet group.

Type: String

Required: No

SubnetIds

A list of subnet IDs in the subnet group.

Type: Array of strings

Required: No

Response Syntax

```
{
   "SubnetGroup": {
      "Description": "string",
      "SubnetGroupName": "string",
      "Subnets": [
         {
            "SubnetAvailabilityZone": "string",
            "SubnetIdentifier": "string"
         }
      ],
      "VpcId": "string"
   }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

SubnetGroup

The subnet group that has been modified.

Type: SubnetGroup object

Errors

For information about the errors that are common to all actions, see Common Errors.

InvalidSubnet

An invalid subnet identifier was specified.

HTTP Status Code: 400

ServiceLinkedRoleNotFoundFault

The specified service linked role (SLR) was not found.

HTTP Status Code: 400

SubnetGroupNotFoundFault

The requested subnet group name does not refer to an existing subnet group.

HTTP Status Code: 400

SubnetInUse

The requested subnet is being used by another subnet group.

HTTP Status Code: 400

SubnetQuotaExceededFault

The request cannot be processed because it would exceed the allowed number of subnets in a subnet group.

HTTP Status Code: 400

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Amazon DynamoDB Streams

The following actions are supported by Amazon DynamoDB Streams:

- DescribeStream
- GetRecords
- GetShardIterator
- ListStreams

DescribeStream

Service: Amazon DynamoDB Streams

Returns information about a stream, including the current status of the stream, its Amazon Resource Name (ARN), the composition of its shards, and its corresponding DynamoDB table.

> **Note**
>
> You can call `DescribeStream` at a maximum rate of 10 times per second.

Each shard in the stream has a `SequenceNumberRange` associated with it. If the `SequenceNumberRange` has a `StartingSequenceNumber` but no `EndingSequenceNumber`, then the shard is still open (able to receive more stream records). If both `StartingSequenceNumber` and `EndingSequenceNumber` are present, then that shard is closed and can no longer receive more data.

Request Syntax

```
{
   "ExclusiveStartShardId": "string",
   "Limit": number,
   "ShardFilter": {
      "ShardId": "string",
      "Type": "string"
   },
   "StreamArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

StreamArn

The Amazon Resource Name (ARN) for the stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

ExclusiveStartShardId

The shard ID of the first item that this operation will evaluate. Use the value that was returned for `LastEvaluatedShardId` in the previous operation.

Type: String

Length Constraints: Minimum length of 28. Maximum length of 65.

Required: No

Limit

The maximum number of shard objects to return. The upper limit is 100.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

ShardFilter

This optional field contains the filter definition for the `DescribeStream` API.

Type: ShardFilter object

Required: No

Response Syntax

```
{
   "StreamDescription": {
      "CreationRequestDateTime": number,
```

```
        "KeySchema": [
           {
              "AttributeName": "string",
              "KeyType": "string"
           }
        ],
        "LastEvaluatedShardId": "string",
        "Shards": [
           {
              "ParentShardId": "string",
              "SequenceNumberRange": {
                 "EndingSequenceNumber": "string",
                 "StartingSequenceNumber": "string"
              },
              "ShardId": "string"
           }
        ],
        "StreamArn": "string",
        "StreamLabel": "string",
        "StreamStatus": "string",
        "StreamViewType": "string",
        "TableName": "string"
    }
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

StreamDescription

A complete description of the stream, including its creation date and time, the DynamoDB table associated with the stream, the shard IDs within the stream, and the beginning and ending sequence numbers of stream records within the shards.

Type: StreamDescription object

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Describe A Stream

The following sample returns a description of a stream with a given stream ARN. All of the shards in the stream are listed in the response, along with the beginning and ending sequence numbers of stream records within the shards. Note that one of the shards is still open, because it does not have an `EndingSequenceNumber`.

Sample Request

```
POST / HTTP/1.1
x-amzn-RequestId: <RequestID>
x-amzn-crc32: <CRC32>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDBStreams_20120810.DescribeStream

{
    "StreamArn": "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/
stream/2015-05-20T20:51:10.252"
    "ShardFilter":"{
    "Type": "CHILD_SHARDS",
    "ShardId": "shardId-00000001741631711871-1f6a72cf"
    }"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "StreamDescription": {
        "StreamArn": "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/
stream/2015-05-20T20:51:10.252",
        "StreamLabel": "2015-05-20T20:51:10.252",
        "StreamStatus": "ENABLED",
        "StreamViewType": "NEW_AND_OLD_IMAGES",
        "CreationRequestDateTime": "Wed May 20 13:51:10 PDT 2015",
        "TableName": "Forum",
        "KeySchema": [
            {"AttributeName": "ForumName","KeyType": "HASH"},
            {"AttributeName": "Subject","KeyType": "RANGE"}
        ],
        "Shards": [
            {
                "SequenceNumberRange": {
                    "EndingSequenceNumber": "20500000000000000910398",
                    "StartingSequenceNumber": "20500000000000000910398"
                },
                "ShardId": "shardId-00000001414562045508-2bac9cd2"
            },
            {
                "ParentShardId": "shardId-00000001414562045508-2bac9cd2",
                "SequenceNumberRange": {
                    "EndingSequenceNumber": "820400000000000001192334",
                    "StartingSequenceNumber": "820400000000000001192334"
                },
                "ShardId": "shardId-00000001414576573621-f55eea83"
            },
            {
                "ParentShardId": "shardId-00000001414576573621-f55eea83",
                "SequenceNumberRange": {
                    "EndingSequenceNumber": "168370000000000001135967",
                    "StartingSequenceNumber": "168370000000000001135967"
```

```
                },
                "ShardId": "shardId-00000001414592258131-674fd923"
            },
            {
                "ParentShardId": "shardId-00000001414592258131-674fd923",
                "SequenceNumberRange": {"StartingSequenceNumber":
 "25746000000000000000935255"},
                "ShardId": "shardId-00000001414608446368-3a1afbaf"
            }
        ],
    }
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetRecords

Service: Amazon DynamoDB Streams

Retrieves the stream records from a given shard.

Specify a shard iterator using the `ShardIterator` parameter. The shard iterator specifies the position in the shard from which you want to start reading stream records sequentially. If there are no stream records available in the portion of the shard that the iterator points to, `GetRecords` returns an empty list. Note that it might take multiple calls to get to a portion of the shard that contains stream records.

> **Note**
>
> `GetRecords` can retrieve a maximum of 1 MB of data or 1000 stream records, whichever comes first.

Request Syntax

```
{
   "Limit": number,
   "ShardIterator": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ShardIterator

A shard iterator that was retrieved from a previous GetShardIterator operation. This iterator can be used to access the stream records in this shard.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: Yes

Limit

The maximum number of records to return from the shard. The upper limit is 1000.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

Response Syntax

```
{
   "NextShardIterator": "string",
   "Records": [
      {
         "awsRegion": "string",
         "dynamodb": {
            "ApproximateCreationDateTime": number,
            "Keys": {
               "string" : {
                  "B": blob,
                  "BOOL": boolean,
                  "BS": [ blob ],
                  "L": [
                     "AttributeValue"
                  ],
                  "M": {
                     "string" : "AttributeValue"
                  },
                  "N": "string",
                  "NS": [ "string" ],
                  "NULL": boolean,
                  "S": "string",
                  "SS": [ "string" ]
               }
            },
            "NewImage": {
               "string" : {
                  "B": blob,
                  "BOOL": boolean,
                  "BS": [ blob ],
```

```
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "OldImage": {
             "string" : {
                "B": blob,
                "BOOL": boolean,
                "BS": [ blob ],
                "L": [
                   "AttributeValue"
                ],
                "M": {
                   "string" : "AttributeValue"
                },
                "N": "string",
                "NS": [ "string" ],
                "NULL": boolean,
                "S": "string",
                "SS": [ "string" ]
             }
          },
          "SequenceNumber": "string",
          "SizeBytes": number,
          "StreamViewType": "string"
       },
       "eventID": "string",
       "eventName": "string",
       "eventSource": "string",
       "eventVersion": "string",
       "userIdentity": {
          "PrprincipalId": "string",
          "Type": "string"
       }
    }
```

```
    ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

NextShardIterator

The next position in the shard from which to start sequentially reading stream records. If set to `null`, the shard has been closed and the requested iterator will not return any more data.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Records

The stream records from the shard, which were retrieved using the shard iterator.

Type: Array of Record objects

Errors

For information about the errors that are common to all actions, see Common Errors.

ExpiredIteratorException

The shard iterator has expired and can no longer be used to retrieve stream records. A shard iterator expires 15 minutes after it is retrieved using the `GetShardIterator` action.

HTTP Status Code: 400

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

LimitExceededException

There is no limit to the number of daily on-demand backups that can be taken.

For most purposes, up to 500 simultaneous table operations are allowed per account. These operations include `CreateTable`, `UpdateTable`, `DeleteTable`,`UpdateTimeToLive`, `RestoreTableFromBackup`, and `RestoreTableToPointInTime`.

When you are creating a table with one or more secondary indexes, you can have up to 250 such requests running at a time. However, if the table or index specifications are complex, then DynamoDB might temporarily reduce the number of concurrent operations.

When importing into DynamoDB, up to 50 simultaneous import table operations are allowed per account.

There is a soft account quota of 2,500 tables.

GetRecords was called with a value of more than 1000 for the limit request parameter.

More than 2 processes are reading from the same streams shard at the same time. Exceeding this limit may result in request throttling.

HTTP Status Code: 400

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

TrimmedDataAccessException

The operation attempted to read past the oldest stream record in a shard.

In DynamoDB Streams, there is a 24 hour limit on data retention. Stream records whose age exceeds this limit are subject to removal (trimming) from the stream. You might receive a TrimmedDataAccessException if:

- You request a shard iterator with a sequence number older than the trim point (24 hours).
- You obtain a shard iterator, but before you use the iterator in a `GetRecords` request, a stream record in the shard exceeds the 24 hour period and is trimmed. This causes the iterator to access a record that no longer exists.

HTTP Status Code: 400

Examples

Retrieve stream records from a shard

The following sample retrieves all the stream records from a shard. To do this, it uses a `ShardIterator` that was obtained from a previous `GetShardIterator` call.

Sample Request

```
POST / HTTP/1.1
x-amzn-RequestId: <RequestID>
x-amzn-crc32: <CRC32>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDBStreams_20120810.GetRecords

{
    "ShardIterator": "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/
stream/2015-05-20T20:51:10.252|1|AAAAAAAAAAEvJp6D+zaQ...  <remaining characters
 omitted> ..."
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "NextShardIterator": "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/
stream/2015-05-20T20:51:10.252|1|AAAAAAAAAAGQBYshYDEe ... <remaining characters
 omitted> ...",
    "Records": [
        {
            "awsRegion": "us-west-2",
            "dynamodb": {
```

```
                "ApproximateCreationDateTime": 1.46480431E9,
                "Keys": {
                    "ForumName": {"S": "DynamoDB"},
                    "Subject": {"S": "DynamoDB Thread 3"}
                },
                "SequenceNumber": "30000000000000000499659",
                "SizeBytes": 41,
                "StreamViewType": "KEYS_ONLY"
            },
            "eventID": "e2fd9c34eff2d779b297b26f5fef4206",
            "eventName": "INSERT",
            "eventSource": "aws:dynamodb",
            "eventVersion": "1.0"
        },
        {
            "awsRegion": "us-west-2",
            "dynamodb": {
                "ApproximateCreationDateTime": 1.46480527E9,
                "Keys": {
                    "ForumName": {"S": "DynamoDB"},
                    "Subject": {"S": "DynamoDB Thread 1"}
                },
                "SequenceNumber": "40000000000000000499660",
                "SizeBytes": 41,
                "StreamViewType": "KEYS_ONLY"
            },
            "eventID": "4b25bd0da9a181a155114127e4837252",
            "eventName": "MODIFY",
            "eventSource": "aws:dynamodb",
            "eventVersion": "1.0"
        },
        {
            "awsRegion": "us-west-2",
            "dynamodb": {
                "ApproximateCreationDateTime": 1.46480646E9,
                "Keys": {
                    "ForumName": {"S": "DynamoDB"},
                    "Subject": {"S": "DynamoDB Thread 2"}
                },
                "SequenceNumber": "50000000000000000499661",
                "SizeBytes": 41,
                "StreamViewType": "KEYS_ONLY"
            },
            "eventID": "740280c73a3df7842edab3548a1b08ad",
```

```
            "eventName": "REMOVE",
            "eventSource": "aws:dynamodb",
            "eventVersion": "1.0"
        }
    ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

GetShardIterator

Service: Amazon DynamoDB Streams

Returns a shard iterator. A shard iterator provides information about how to retrieve the stream records from within a shard. Use the shard iterator in a subsequent `GetRecords` request to read the stream records from the shard.

> **Note**
>
> A shard iterator expires 15 minutes after it is returned to the requester.

Request Syntax

```
{
   "SequenceNumber": "string",
   "ShardId": "string",
   "ShardIteratorType": "string",
   "StreamArn": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ShardId

The identifier of the shard. The iterator will be returned for this shard ID.

Type: String

Length Constraints: Minimum length of 28. Maximum length of 65.

Required: Yes

ShardIteratorType

Determines how the shard iterator is used to start reading stream records from the shard:

- `AT_SEQUENCE_NUMBER` - Start reading exactly from the position denoted by a specific sequence number.
- `AFTER_SEQUENCE_NUMBER` - Start reading right after the position denoted by a specific sequence number.
- `TRIM_HORIZON` - Start reading at the last (untrimmed) stream record, which is the oldest record in the shard. In DynamoDB Streams, there is a 24 hour limit on data retention. Stream records whose age exceeds this limit are subject to removal (trimming) from the stream.
- `LATEST` - Start reading just after the most recent stream record in the shard, so that you always read the most recent data in the shard.

Type: String

Valid Values: `TRIM_HORIZON | LATEST | AT_SEQUENCE_NUMBER | AFTER_SEQUENCE_NUMBER`

Required: Yes

StreamArn

The Amazon Resource Name (ARN) for the stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

SequenceNumber

The sequence number of a stream record in the shard from which to start reading.

Type: String

Length Constraints: Minimum length of 21. Maximum length of 40.

Required: No

Response Syntax

```
{
```

```
    "ShardIterator": "string"
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

ShardIterator

The position in the shard from which to start reading stream records sequentially. A shard iterator specifies this position using the sequence number of a stream record in a shard.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

TrimmedDataAccessException

The operation attempted to read past the oldest stream record in a shard.

In DynamoDB Streams, there is a 24 hour limit on data retention. Stream records whose age exceeds this limit are subject to removal (trimming) from the stream. You might receive a TrimmedDataAccessException if:

- You request a shard iterator with a sequence number older than the trim point (24 hours).
- You obtain a shard iterator, but before you use the iterator in a `GetRecords` request, a stream record in the shard exceeds the 24 hour period and is trimmed. This causes the iterator to access a record that no longer exists.

HTTP Status Code: 400

Examples

Retrieve a Shard Iterator For a Stream

The following sample returns a shard iterator for the provided stream ARN and shard ID. The shard iterator will allow access to stream records beginning with the given sequence number.

Sample Request

```
POST / HTTP/1.1
x-amzn-RequestId: <RequestID>
x-amzn-crc32: <CRC32>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDBStreams_20120810.GetShardIterator

{
    "StreamArn": "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/
stream/2015-05-20T20:51:10.252",
    "ShardId": "00000001414576573621-f55eea83",
    "ShardIteratorType": "TRIM_HORIZON"
}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
```

```
Date: <Date>

{
    "ShardIterator": "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/
stream/2015-05-20T20:51:10.252|1|AAAAAAAAAAEvJp6D+zaQ...  <remaining characters
 omitted> ..."
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListStreams

Service: Amazon DynamoDB Streams

Returns an array of stream ARNs associated with the current account and endpoint. If the `TableName` parameter is present, then `ListStreams` will return only the streams ARNs for that table.

> **Note**
>
> You can call `ListStreams` at a maximum rate of 5 times per second.

Request Syntax

```
{
   "ExclusiveStartStreamArn": "string",
   "Limit": number,
   "TableName": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

> **Note**
>
> In the following list, the required parameters are described first.

ExclusiveStartStreamArn

The ARN (Amazon Resource Name) of the first item that this operation will evaluate. Use the value that was returned for `LastEvaluatedStreamArn` in the previous operation.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

Limit

The maximum number of streams to return. The upper limit is 100.

Type: Integer

Valid Range: Minimum value of 1.

Required: No

TableName

If this parameter is provided, then only the streams associated with this table name are returned.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

Response Syntax

```
{
   "LastEvaluatedStreamArn": "string",
   "Streams": [
      {
         "StreamArn": "string",
         "StreamLabel": "string",
         "TableName": "string"
      }
   ]
}
```

Response Elements

If the action is successful, the service sends back an HTTP 200 response.

The following data is returned in JSON format by the service.

LastEvaluatedStreamArn

The stream ARN of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If `LastEvaluatedStreamArn` is empty, then the "last page" of results has been processed and there is no more data to be retrieved.

If `LastEvaluatedStreamArn` is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when `LastEvaluatedStreamArn` is empty.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Streams

A list of stream descriptors associated with the current account and endpoint.

Type: Array of Stream objects

Errors

For information about the errors that are common to all actions, see Common Errors.

InternalServerError

An error occurred on the server side.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a nonexistent table or index. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

Examples

Retrieve All Stream ARNs

The following sample returns all of the stream ARNs.

Sample Request

```
POST / HTTP/1.1
x-amzn-RequestId: <RequestID>
x-amzn-crc32: <CRC32>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
X-Amz-Date: <Date>
X-Amz-Target: DynamoDBStreams_20120810.ListStreams

{}
```

Sample Response

```
HTTP/1.1 200 OK
x-amzn-RequestId: <RequestId>
x-amz-crc32: <Checksum>
Content-Type: application/x-amz-json-1.0
Content-Length: <PayloadSizeBytes>
Date: <Date>

{
    "Streams": [
        {
            "StreamArn": "arn:aws:dynamodb:us-wesst-2:111122223333:table/Forum/
stream/2015-05-20T20:51:10.252",
            "TableName": "Forum",
            "StreamLabel": "2015-05-20T20:51:10.252"
        },
        {
            "StreamArn": "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/
stream/2015-05-20T20:50:02.714",
            "TableName": "Forum",
            "StreamLabel": "2015-05-20T20:50:02.714"
        },
        {
            "StreamArn": "arn:aws:dynamodb:us-west-2:111122223333:table/Forum/
stream/2015-05-19T23:03:50.641",
            "TableName": "Forum",
            "StreamLabel": "2015-05-19T23:03:50.641"
```

```
        },

        ...remaining output omitted...

    ]
}
```

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go v2
- AWS SDK for Java V2
- AWS SDK for JavaScript V3
- AWS SDK for Kotlin
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Data Types

The following data types are supported by Amazon DynamoDB:

- ArchivalSummary
- AttributeDefinition
- AttributeValue
- AttributeValueUpdate
- AutoScalingPolicyDescription
- AutoScalingPolicyUpdate
- AutoScalingSettingsDescription
- AutoScalingSettingsUpdate
- AutoScalingTargetTrackingScalingPolicyConfigurationDescription
- AutoScalingTargetTrackingScalingPolicyConfigurationUpdate
- BackupDescription
- BackupDetails
- BackupSummary
- BatchStatementError
- BatchStatementRequest
- BatchStatementResponse
- BillingModeSummary
- CancellationReason
- Capacity
- Condition
- ConditionCheck
- ConsumedCapacity
- ContinuousBackupsDescription
- ContributorInsightsSummary
- CreateGlobalSecondaryIndexAction
- CreateGlobalTableWitnessGroupMemberAction

- CreateReplicaAction
- CreateReplicationGroupMemberAction
- CsvOptions
- Delete
- DeleteGlobalSecondaryIndexAction
- DeleteGlobalTableWitnessGroupMemberAction
- DeleteReplicaAction
- DeleteReplicationGroupMemberAction
- DeleteRequest
- EnableKinesisStreamingConfiguration
- Endpoint
- ExpectedAttributeValue
- ExportDescription
- ExportSummary
- FailureException
- Get
- GlobalSecondaryIndex
- GlobalSecondaryIndexAutoScalingUpdate
- GlobalSecondaryIndexDescription
- GlobalSecondaryIndexInfo
- GlobalSecondaryIndexUpdate
- GlobalSecondaryIndexWarmThroughputDescription
- GlobalTable
- GlobalTableDescription
- GlobalTableGlobalSecondaryIndexSettingsUpdate
- GlobalTableWitnessDescription
- GlobalTableWitnessGroupUpdate
- ImportSummary
- ImportTableDescription

- IncrementalExportSpecification
- InputFormatOptions
- ItemCollectionMetrics
- ItemResponse
- KeysAndAttributes
- KeySchemaElement
- KinesisDataStreamDestination
- LocalSecondaryIndex
- LocalSecondaryIndexDescription
- LocalSecondaryIndexInfo
- OnDemandThroughput
- OnDemandThroughputOverride
- ParameterizedStatement
- PointInTimeRecoveryDescription
- PointInTimeRecoverySpecification
- Projection
- ProvisionedThroughput
- ProvisionedThroughputDescription
- ProvisionedThroughputOverride
- Put
- PutRequest
- Replica
- ReplicaAutoScalingDescription
- ReplicaAutoScalingUpdate
- ReplicaDescription
- ReplicaGlobalSecondaryIndex
- ReplicaGlobalSecondaryIndexAutoScalingDescription
- ReplicaGlobalSecondaryIndexAutoScalingUpdate
- ReplicaGlobalSecondaryIndexDescription

- ReplicaGlobalSecondaryIndexSettingsDescription
- ReplicaGlobalSecondaryIndexSettingsUpdate
- ReplicaSettingsDescription
- ReplicaSettingsUpdate
- ReplicationGroupUpdate
- ReplicaUpdate
- RestoreSummary
- S3BucketSource
- SourceTableDetails
- SourceTableFeatureDetails
- SSEDescription
- SSESpecification
- StreamSpecification
- TableAutoScalingDescription
- TableClassSummary
- TableCreationParameters
- TableDescription
- TableWarmThroughputDescription
- Tag
- ThrottlingReason
- TimeToLiveDescription
- TimeToLiveSpecification
- TransactGetItem
- TransactWriteItem
- Update
- UpdateGlobalSecondaryIndexAction
- UpdateKinesisStreamingConfiguration
- UpdateReplicationGroupMemberAction
- WarmThroughput
- WriteRequest

The following data types are supported by DynamoDB Accelerator:

- Cluster
- Endpoint
- Event
- Node
- NodeTypeSpecificValue
- NotificationConfiguration
- Parameter
- ParameterGroup
- ParameterGroupStatus
- ParameterNameValue
- SecurityGroupMembership
- SSEDescription
- SSESpecification
- Subnet
- SubnetGroup
- Tag

The following data types are supported by Amazon DynamoDB Streams:

- AttributeValue
- Identity
- KeySchemaElement
- Record
- SequenceNumberRange
- Shard
- ShardFilter
- Stream
- StreamDescription
- StreamRecord

Amazon DynamoDB

The following data types are supported by Amazon DynamoDB:

- ArchivalSummary
- AttributeDefinition
- AttributeValue
- AttributeValueUpdate
- AutoScalingPolicyDescription
- AutoScalingPolicyUpdate
- AutoScalingSettingsDescription
- AutoScalingSettingsUpdate
- AutoScalingTargetTrackingScalingPolicyConfigurationDescription
- AutoScalingTargetTrackingScalingPolicyConfigurationUpdate
- BackupDescription
- BackupDetails
- BackupSummary
- BatchStatementError
- BatchStatementRequest
- BatchStatementResponse
- BillingModeSummary
- CancellationReason
- Capacity
- Condition
- ConditionCheck
- ConsumedCapacity
- ContinuousBackupsDescription
- ContributorInsightsSummary
- CreateGlobalSecondaryIndexAction
- CreateGlobalTableWitnessGroupMemberAction
- CreateReplicaAction

- CreateReplicationGroupMemberAction
- CsvOptions
- Delete
- DeleteGlobalSecondaryIndexAction
- DeleteGlobalTableWitnessGroupMemberAction
- DeleteReplicaAction
- DeleteReplicationGroupMemberAction
- DeleteRequest
- EnableKinesisStreamingConfiguration
- Endpoint
- ExpectedAttributeValue
- ExportDescription
- ExportSummary
- FailureException
- Get
- GlobalSecondaryIndex
- GlobalSecondaryIndexAutoScalingUpdate
- GlobalSecondaryIndexDescription
- GlobalSecondaryIndexInfo
- GlobalSecondaryIndexUpdate
- GlobalSecondaryIndexWarmThroughputDescription
- GlobalTable
- GlobalTableDescription
- GlobalTableGlobalSecondaryIndexSettingsUpdate
- GlobalTableWitnessDescription
- GlobalTableWitnessGroupUpdate
- ImportSummary
- ImportTableDescription
- IncrementalExportSpecification
- InputFormatOptions

- ItemCollectionMetrics
- ItemResponse
- KeysAndAttributes
- KeySchemaElement
- KinesisDataStreamDestination
- LocalSecondaryIndex
- LocalSecondaryIndexDescription
- LocalSecondaryIndexInfo
- OnDemandThroughput
- OnDemandThroughputOverride
- ParameterizedStatement
- PointInTimeRecoveryDescription
- PointInTimeRecoverySpecification
- Projection
- ProvisionedThroughput
- ProvisionedThroughputDescription
- ProvisionedThroughputOverride
- Put
- PutRequest
- Replica
- ReplicaAutoScalingDescription
- ReplicaAutoScalingUpdate
- ReplicaDescription
- ReplicaGlobalSecondaryIndex
- ReplicaGlobalSecondaryIndexAutoScalingDescription
- ReplicaGlobalSecondaryIndexAutoScalingUpdate
- ReplicaGlobalSecondaryIndexDescription
- ReplicaGlobalSecondaryIndexSettingsDescription
- ReplicaGlobalSecondaryIndexSettingsUpdate
- ReplicaSettingsDescription

- ReplicaSettingsUpdate
- ReplicationGroupUpdate
- ReplicaUpdate
- RestoreSummary
- S3BucketSource
- SourceTableDetails
- SourceTableFeatureDetails
- SSEDescription
- SSESpecification
- StreamSpecification
- TableAutoScalingDescription
- TableClassSummary
- TableCreationParameters
- TableDescription
- TableWarmThroughputDescription
- Tag
- ThrottlingReason
- TimeToLiveDescription
- TimeToLiveSpecification
- TransactGetItem
- TransactWriteItem
- Update
- UpdateGlobalSecondaryIndexAction
- UpdateKinesisStreamingConfiguration
- UpdateReplicationGroupMemberAction
- WarmThroughput
- WriteRequest

ArchivalSummary

Service: Amazon DynamoDB

Contains details of a table archival operation.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ArchivalBackupArn

The Amazon Resource Name (ARN) of the backup the table was archived to, when applicable in the archival reason. If you wish to restore this backup to the same table name, you will need to delete the original table.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

ArchivalDateTime

The date and time when table archival was initiated by DynamoDB, in UNIX epoch time format.

Type: Timestamp

Required: No

ArchivalReason

The reason DynamoDB archived the table. Currently, the only possible value is:

- `INACCESSIBLE_ENCRYPTION_CREDENTIALS` - The table was archived due to the table's AWS KMS key being inaccessible for more than seven days. An On-Demand backup was created at the archival time.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AttributeDefinition

Service: Amazon DynamoDB

Represents an attribute for describing the schema for the table and indexes.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AttributeName

A name for the attribute.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

AttributeType

The data type for the attribute, where:

- S - the attribute is of type String
- N - the attribute is of type Number
- B - the attribute is of type Binary

Type: String

Valid Values: S | N | B

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

AttributeValue

Service: Amazon DynamoDB

Represents the data for an attribute.

Each attribute value is described as a name-value pair. The name is the data type, and the value is the data itself.

For more information, see Data Types in the *Amazon DynamoDB Developer Guide*.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

B

An attribute of type Binary. For example:

```
"B": "dGhpcyB0ZXh0IGlzIGJhc2U2NC1lbmNvZGVk"
```

Type: Base64-encoded binary data object

Required: No

BOOL

An attribute of type Boolean. For example:

```
"BOOL": true
```

Type: Boolean

Required: No

BS

An attribute of type Binary Set. For example:

```
"BS": ["U3Vubnk=", "UmFpbnk=", "U25vd3k="]
```

Type: Array of Base64-encoded binary data objects

Required: No

L

An attribute of type List. For example:

```
"L": [ {"S": "Cookies"} , {"S": "Coffee"}, {"N": "3.14159"}]
```

Type: Array of AttributeValue objects

Required: No

M

An attribute of type Map. For example:

```
"M": {"Name": {"S": "Joe"}, "Age": {"N": "35"}}
```

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

N

An attribute of type Number. For example:

```
"N": "123.45"
```

Numbers are sent across the network to DynamoDB as strings, to maximize compatibility across languages and libraries. However, DynamoDB treats them as number type attributes for mathematical operations.

Type: String

Required: No

NS

An attribute of type Number Set. For example:

```
"NS": ["42.2", "-19", "7.5", "3.14"]
```

Numbers are sent across the network to DynamoDB as strings, to maximize compatibility across languages and libraries. However, DynamoDB treats them as number type attributes for mathematical operations.

Type: Array of strings

Required: No

NULL

An attribute of type Null. For example:

```
"NULL": true
```

Type: Boolean

Required: No

S

An attribute of type String. For example:

```
"S": "Hello"
```

Type: String

Required: No

SS

An attribute of type String Set. For example:

```
"SS": ["Giraffe", "Hippo" ,"Zebra"]
```

Type: Array of strings

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

AttributeValueUpdate

Service: Amazon DynamoDB

For the `UpdateItem` operation, represents the attributes to be modified, the action to perform on each, and the new value for each.

> **Note**
>
> You cannot use `UpdateItem` to update any primary key attributes. Instead, you will need to delete the item, and then use `PutItem` to create a new item with new attributes.

Attribute values cannot be null; string and binary type attributes must have lengths greater than zero; and set type attributes must not be empty. Requests with empty values will be rejected with a `ValidationException` exception.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Action

Specifies how to perform the update. Valid values are `PUT` (default), `DELETE`, and `ADD`. The behavior depends on whether the specified primary key already exists in the table.

If an item with the specified *Key* is found in the table:

- `PUT` - Adds the specified attribute to the item. If the attribute already exists, it is replaced by the new value.
- `DELETE` - If no value is specified, the attribute and its value are removed from the item. The data type of the specified value must match the existing value's data type.

  If a *set* of values is specified, then those values are subtracted from the old set. For example, if the attribute value was the set `[a,b,c]` and the `DELETE` action specified `[a,c]`, then the final attribute value would be `[b]`. Specifying an empty set is an error.

- ADD - If the attribute does not already exist, then the attribute and its values are added to the item. If the attribute does exist, then the behavior of ADD depends on the data type of the attribute:
  - If the existing attribute is a number, and if `Value` is also a number, then the `Value` is mathematically added to the existing attribute. If `Value` is a negative number, then it is subtracted from the existing attribute.

    > **Note**
    >
    > If you use ADD to increment or decrement a number value for an item that doesn't exist before the update, DynamoDB uses 0 as the initial value.
    > In addition, if you use ADD to update an existing item, and intend to increment or decrement an attribute value which does not yet exist, DynamoDB uses `0` as the initial value. For example, suppose that the item you want to update does not yet have an attribute named *itemcount*, but you decide to ADD the number `3` to this attribute anyway, even though it currently does not exist. DynamoDB will create the *itemcount* attribute, set its initial value to `0`, and finally add `3` to it. The result will be a new *itemcount* attribute in the item, with a value of `3`.

  - If the existing data type is a set, and if the `Value` is also a set, then the `Value` is added to the existing set. (This is a *set* operation, not mathematical addition.) For example, if the attribute value was the set `[1,2]`, and the ADD action specified `[3]`, then the final attribute value would be `[1,2,3]`. An error occurs if an Add action is specified for a set attribute and the attribute type specified does not match the existing set type.

    Both sets must have the same primitive data type. For example, if the existing data type is a set of strings, the `Value` must also be a set of strings. The same holds true for number sets and binary sets.

  This action is only valid for an existing attribute whose data type is number or is a set. Do not use ADD for any other data types.

If no item with the specified *Key* is found:

- PUT - DynamoDB creates a new item with the specified primary key, and then adds the attribute.
- DELETE - Nothing happens; there is no attribute to delete.

- ADD - DynamoDB creates a new item with the supplied primary key and number (or set) for the attribute value. The only data types allowed are number, number set, string set or binary set.

Type: String

Valid Values: `ADD` | `PUT` | `DELETE`

Required: No

Value

Represents the data for an attribute.

Each attribute value is described as a name-value pair. The name is the data type, and the value is the data itself.

For more information, see Data Types in the *Amazon DynamoDB Developer Guide*.

Type: AttributeValue object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AutoScalingPolicyDescription

Service: Amazon DynamoDB

Represents the properties of the scaling policy.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

PolicyName

The name of the scaling policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `\p{Print}+`

Required: No

TargetTrackingScalingPolicyConfiguration

Represents a target tracking scaling policy configuration.

Type: AutoScalingTargetTrackingScalingPolicyConfigurationDescription object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AutoScalingPolicyUpdate

Service: Amazon DynamoDB

Represents the auto scaling policy to be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

TargetTrackingScalingPolicyConfiguration

Represents a target tracking scaling policy configuration.

Type: AutoScalingTargetTrackingScalingPolicyConfigurationUpdate object

Required: Yes

PolicyName

The name of the scaling policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: `\p{Print}+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AutoScalingSettingsDescription

Service: Amazon DynamoDB

Represents the auto scaling settings for a global table or global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AutoScalingDisabled

Disabled auto scaling for this global table or global secondary index.

Type: Boolean

Required: No

AutoScalingRoleArn

Role ARN used for configuring the auto scaling policy.

Type: String

Required: No

MaximumUnits

The maximum capacity units that a global table or global secondary index should be scaled up to.

Type: Long

Valid Range: Minimum value of 1.

Required: No

MinimumUnits

The minimum capacity units that a global table or global secondary index should be scaled down to.

Type: Long

Valid Range: Minimum value of 1.

Required: No

ScalingPolicies

Information about the scaling policies.

Type: Array of AutoScalingPolicyDescription objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AutoScalingSettingsUpdate

Service: Amazon DynamoDB

Represents the auto scaling settings to be modified for a global table or global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AutoScalingDisabled

Disabled auto scaling for this global table or global secondary index.

Type: Boolean

Required: No

AutoScalingRoleArn

Role ARN used for configuring auto scaling policy.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1600.

Pattern: `[\u0020-\uD7FF\uE000-\uFFFD\uD800\uDC00-\uDBFF\uDFFF\r\n\t]*`

Required: No

MaximumUnits

The maximum capacity units that a global table or global secondary index should be scaled up to.

Type: Long

Valid Range: Minimum value of 1.

Required: No

MinimumUnits

The minimum capacity units that a global table or global secondary index should be scaled down to.

Type: Long

Valid Range: Minimum value of 1.

Required: No

ScalingPolicyUpdate

The scaling policy to apply for scaling target global table or global secondary index capacity units.

Type: AutoScalingPolicyUpdate object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AutoScalingTargetTrackingScalingPolicyConfigurationDescription

Service: Amazon DynamoDB

Represents the properties of a target tracking scaling policy.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

TargetValue

The target value for the metric. The range is 8.515920e-109 to 1.174271e+108 (Base 10) or 2e-360 to 2e360 (Base 2).

Type: Double

Required: Yes

DisableScaleIn

Indicates whether scale in by the target tracking policy is disabled. If the value is true, scale in is disabled and the target tracking policy won't remove capacity from the scalable resource. Otherwise, scale in is enabled and the target tracking policy can remove capacity from the scalable resource. The default value is false.

Type: Boolean

Required: No

ScaleInCooldown

The amount of time, in seconds, after a scale in activity completes before another scale in activity can start. The cooldown period is used to block subsequent scale in requests until it has expired. You should scale in conservatively to protect your application's availability. However, if another alarm triggers a scale out policy during the cooldown period after a scale-in, application auto scaling scales out your scalable target immediately.

Type: Integer

Required: No

ScaleOutCooldown

The amount of time, in seconds, after a scale out activity completes before another scale out activity can start. While the cooldown period is in effect, the capacity that has been added by the previous scale out event that initiated the cooldown is calculated as part of the desired capacity for the next scale out. You should continuously (but not excessively) scale out.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

AutoScalingTargetTrackingScalingPolicyConfigurationUpdate

Service: Amazon DynamoDB

Represents the settings of a target tracking scaling policy that will be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

TargetValue

The target value for the metric. The range is 8.515920e-109 to 1.174271e+108 (Base 10) or 2e-360 to 2e360 (Base 2).

Type: Double

Required: Yes

DisableScaleIn

Indicates whether scale in by the target tracking policy is disabled. If the value is true, scale in is disabled and the target tracking policy won't remove capacity from the scalable resource. Otherwise, scale in is enabled and the target tracking policy can remove capacity from the scalable resource. The default value is false.

Type: Boolean

Required: No

ScaleInCooldown

The amount of time, in seconds, after a scale in activity completes before another scale in activity can start. The cooldown period is used to block subsequent scale in requests until it has expired. You should scale in conservatively to protect your application's availability. However, if another alarm triggers a scale out policy during the cooldown period after a scale-in, application auto scaling scales out your scalable target immediately.

Type: Integer

Required: No

ScaleOutCooldown

The amount of time, in seconds, after a scale out activity completes before another scale out activity can start. While the cooldown period is in effect, the capacity that has been added by the previous scale out event that initiated the cooldown is calculated as part of the desired capacity for the next scale out. You should continuously (but not excessively) scale out.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BackupDescription

Service: Amazon DynamoDB

Contains the description of the backup created for the table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

BackupDetails

Contains the details of the backup created for the table.

Type: BackupDetails object

Required: No

SourceTableDetails

Contains the details of the table when the backup was created.

Type: SourceTableDetails object

Required: No

SourceTableFeatureDetails

Contains the details of the features enabled on the table when the backup was created. For example, LSIs, GSIs, streams, TTL.

Type: SourceTableFeatureDetails object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

BackupDetails

Service: Amazon DynamoDB

Contains the details of the backup created for the table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

BackupArn

ARN associated with the backup.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: Yes

BackupCreationDateTime

Time at which the backup was created. This is the request time of the backup.

Type: Timestamp

Required: Yes

BackupName

Name of the requested backup.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

BackupStatus

Backup can be in one of the following states: CREATING, ACTIVE, DELETED.

Type: String

Valid Values: `CREATING | DELETED | AVAILABLE`

Required: Yes

BackupType

BackupType:

- `USER` - You create and manage these using the on-demand backup feature.
- `SYSTEM` - If you delete a table with point-in-time recovery enabled, a `SYSTEM` backup is automatically created and is retained for 35 days (at no additional cost). System backups allow you to restore the deleted table to the state it was in just before the point of deletion.
- `AWS_BACKUP` - On-demand backup created by you from AWS Backup service.

Type: String

Valid Values: `USER | SYSTEM | AWS_BACKUP`

Required: Yes

BackupExpiryDateTime

Time at which the automatic on-demand backup created by DynamoDB will expire. This `SYSTEM` on-demand backup expires automatically 35 days after its creation.

Type: Timestamp

Required: No

BackupSizeBytes

Size of the backup in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

Type: Long

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BackupSummary

Service: Amazon DynamoDB

Contains details for the backup.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

BackupArn

ARN associated with the backup.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

BackupCreationDateTime

Time at which the backup was created.

Type: Timestamp

Required: No

BackupExpiryDateTime

Time at which the automatic on-demand backup created by DynamoDB will expire. This `SYSTEM` on-demand backup expires automatically 35 days after its creation.

Type: Timestamp

Required: No

BackupName

Name of the specified backup.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

BackupSizeBytes

Size of the backup in bytes.

Type: Long

Valid Range: Minimum value of 0.

Required: No

BackupStatus

Backup can be in one of the following states: CREATING, ACTIVE, DELETED.

Type: String

Valid Values: `CREATING | DELETED | AVAILABLE`

Required: No

BackupType

BackupType:

- `USER` - You create and manage these using the on-demand backup feature.
- `SYSTEM` - If you delete a table with point-in-time recovery enabled, a `SYSTEM` backup is automatically created and is retained for 35 days (at no additional cost). System backups allow you to restore the deleted table to the state it was in just before the point of deletion.
- `AWS_BACKUP` - On-demand backup created by you from AWS Backup service.

Type: String

Valid Values: `USER | SYSTEM | AWS_BACKUP`

Required: No

TableArn

ARN associated with the table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

TableId

Unique identifier for the table.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

TableName

Name of the table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BatchStatementError

Service: Amazon DynamoDB

An error associated with a statement in a PartiQL batch that was run.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Code

The error code associated with the failed PartiQL batch statement.

Type: String

Valid Values: `ConditionalCheckFailed | ItemCollectionSizeLimitExceeded | RequestLimitExceeded | ValidationError | ProvisionedThroughputExceeded | TransactionConflict | ThrottlingError | InternalServerError | ResourceNotFound | AccessDenied | DuplicateItem`

Required: No

Item

The item which caused the condition check to fail. This will be set if ReturnValuesOnConditionCheckFailure is specified as `ALL_OLD`.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

Message

The error message associated with the PartiQL batch response.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BatchStatementRequest

Service: Amazon DynamoDB

A PartiQL batch statement request.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Statement

A valid PartiQL statement.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: Yes

ConsistentRead

The read consistency of the PartiQL batch request.

Type: Boolean

Required: No

Parameters

The parameters associated with a PartiQL statement in the batch request.

Type: Array of AttributeValue objects

Array Members: Minimum number of 1 item.

Required: No

ReturnValuesOnConditionCheckFailure

An optional parameter that returns the item attributes for a PartiQL batch request operation that failed a condition check.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

Type: String

Valid Values: `ALL_OLD` | `NONE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BatchStatementResponse

Service: Amazon DynamoDB

A PartiQL batch statement response..

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Error

The error associated with a failed PartiQL batch statement.

Type: BatchStatementError object

Required: No

Item

A DynamoDB item associated with a BatchStatementResponse

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

TableName

The table name associated with a failed PartiQL batch statement.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

BillingModeSummary

Service: Amazon DynamoDB

Contains the details for the read/write capacity mode. This page talks about `PROVISIONED` and `PAY_PER_REQUEST` billing modes. For more information about these modes, see Read/write capacity mode.

> **Note**
>
> You may need to switch to on-demand mode at least once in order to return a `BillingModeSummary` response.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

BillingMode

Controls how you are charged for read and write throughput and how you manage capacity. This setting can be changed later.

- `PROVISIONED` - Sets the read/write capacity mode to `PROVISIONED`. We recommend using `PROVISIONED` for predictable workloads.
- `PAY_PER_REQUEST` - Sets the read/write capacity mode to `PAY_PER_REQUEST`. We recommend using `PAY_PER_REQUEST` for unpredictable workloads.

Type: String

Valid Values: `PROVISIONED | PAY_PER_REQUEST`

Required: No

LastUpdateToPayPerRequestDateTime

Represents the time when `PAY_PER_REQUEST` was last set as the read/write capacity mode.

Type: Timestamp

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CancellationReason

Service: Amazon DynamoDB

An ordered list of errors for each item in the request which caused the transaction to get cancelled. The values of the list are ordered according to the ordering of the `TransactWriteItems` request parameter. If no error occurred for the associated item an error with a Null code and Null message will be present.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Code

Status code for the result of the cancelled transaction.

Type: String

Required: No

Item

Item in the request which caused the transaction to get cancelled.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

Message

Cancellation reason message description.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Capacity

Service: Amazon DynamoDB

Represents the amount of provisioned throughput capacity consumed on a table or an index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

CapacityUnits

The total number of capacity units consumed on a table or an index.

Type: Double

Required: No

ReadCapacityUnits

The total number of read capacity units consumed on a table or an index.

Type: Double

Required: No

WriteCapacityUnits

The total number of write capacity units consumed on a table or an index.

Type: Double

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

Condition

Service: Amazon DynamoDB

Represents the selection criteria for a `Query` or `Scan` operation:

- For a `Query` operation, `Condition` is used for specifying the `KeyConditions` to use when querying a table or an index. For `KeyConditions`, only the following comparison operators are supported:

  `EQ | LE | LT | GE | GT | BEGINS_WITH | BETWEEN`

  `Condition` is also used in a `QueryFilter`, which evaluates the query results and returns only the desired values.
- For a `Scan` operation, `Condition` is used in a `ScanFilter`, which evaluates the scan results and returns only the desired values.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ComparisonOperator

A comparator for evaluating attributes. For example, equals, greater than, less than, etc.

The following comparison operators are available:

`EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH | IN | BETWEEN`

The following are descriptions of each comparison operator.

- `EQ` : Equal. `EQ` is supported for all data types, including lists and maps.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value

does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not equal `{"NS":["6", "2", "1"]}`.

- `NE` : Not equal. `NE` is supported for all data types, including lists and maps.

  `AttributeValueList` can contain only one `AttributeValue` of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an `AttributeValue` of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not equal `{"NS":["6", "2", "1"]}`.

- `LE` : Less than or equal.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`.

- `LT` : Less than.

  `AttributeValueList` can contain only one `AttributeValue` of type String, Number, or Binary (not a set type). If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`.

- `GE` : Greater than or equal.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`.

- `GT` : Greater than.

`AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`.

- `NOT_NULL` : The attribute exists. `NOT_NULL` is supported for all data types, including lists and maps.

  > **Note**
  >
  > This operator tests for the existence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using `NOT_NULL`, the result is a Boolean `true`. This result is because the attribute "a" exists; its data type is not relevant to the `NOT_NULL` comparison operator.

- `NULL` : The attribute does not exist. `NULL` is supported for all data types, including lists and maps.

  > **Note**
  >
  > This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using `NULL`, the result is a Boolean `false`. This is because the attribute "a" exists; its data type is not relevant to the `NULL` comparison operator.

- `CONTAINS` : Checks for a subsequence, or value in a set.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("SS", "NS", or "BS"), then the operator evaluates to true if it finds an exact match with any member of the set.

  CONTAINS is supported for lists: When evaluating "a `CONTAINS` b", "a" can be a list; however, "b" cannot be a set, a map, or a list.

- `NOT_CONTAINS` : Checks for absence of a subsequence, or absence of a value in a set.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("`SS`", "`NS`", or "`BS`"), then the operator evaluates to true if it *does not* find an exact match with any member of the set.

  NOT_CONTAINS is supported for lists: When evaluating "a `NOT CONTAINS` b", "a" can be a list; however, "b" cannot be a set, a map, or a list.
- `BEGINS_WITH` : Checks for a prefix.

  `AttributeValueList` can contain only one `AttributeValue` of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).

- `IN` : Checks for matching elements in a list.

  `AttributeValueList` can contain one or more `AttributeValue` elements of type String, Number, or Binary. These attributes are compared against an existing attribute of an item. If any elements of the input are equal to the item attribute, the expression evaluates to true.
- `BETWEEN` : Greater than or equal to the first value, and less than or equal to the second value.

  `AttributeValueList` must contain two `AttributeValue` elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not compare to `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`

For usage examples of `AttributeValueList` and `ComparisonOperator`, see Legacy Conditional Parameters in the *Amazon DynamoDB Developer Guide*.

Type: String

Valid Values: `EQ | NE | IN | LE | LT | GE | GT | BETWEEN | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH`

Required: Yes

AttributeValueList

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the `ComparisonOperator` being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

Type: Array of AttributeValue objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ConditionCheck

Service: Amazon DynamoDB

Represents a request to perform a check that an item exists or to check the condition of specific attributes of the item.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ConditionExpression

A condition that must be satisfied in order for a conditional update to succeed. For more information, see Condition expressions in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: Yes

Key

The primary key of the item to be checked. Each element consists of an attribute name and a value for that attribute.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

Name of the table for the check item request. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression. For more information, see Expression attribute names in the *Amazon DynamoDB Developer Guide*.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression. For more information, see Condition expressions in the *Amazon DynamoDB Developer Guide*.

Type: String to AttributeValue object map

Required: No

ReturnValuesOnConditionCheckFailure

Use `ReturnValuesOnConditionCheckFailure` to get the item attributes if the `ConditionCheck` condition fails. For `ReturnValuesOnConditionCheckFailure`, the valid values are: NONE and ALL_OLD.

Type: String

Valid Values: `ALL_OLD | NONE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ConsumedCapacity

Service: Amazon DynamoDB

The capacity units consumed by an operation. The data returned includes the total provisioned throughput consumed, along with statistics for the table and any indexes involved in the operation. `ConsumedCapacity` is only returned if the request asked for it. For more information, see Provisioned capacity mode in the *Amazon DynamoDB Developer Guide*.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

CapacityUnits

The total number of capacity units consumed by the operation.

Type: Double

Required: No

GlobalSecondaryIndexes

The amount of throughput consumed on each global index affected by the operation.

Type: String to Capacity object map

Key Length Constraints: Minimum length of 3. Maximum length of 255.

Key Pattern: `[a-zA-Z0-9_.-]+`

Required: No

LocalSecondaryIndexes

The amount of throughput consumed on each local index affected by the operation.

Type: String to Capacity object map

Key Length Constraints: Minimum length of 3. Maximum length of 255.

Key Pattern: `[a-zA-Z0-9_.-]+`

Required: No

ReadCapacityUnits

The total number of read capacity units consumed by the operation.

Type: Double

Required: No

Table

The amount of throughput consumed on the table affected by the operation.

Type: Capacity object

Required: No

TableName

The name of the table that was affected by the operation. If you had specified the Amazon Resource Name (ARN) of a table in the input, you'll see the table ARN in the response.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

WriteCapacityUnits

The total number of write capacity units consumed by the operation.

Type: Double

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContinuousBackupsDescription

Service: Amazon DynamoDB

Represents the continuous backups and point in time recovery settings on the table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ContinuousBackupsStatus

`ContinuousBackupsStatus` can be one of the following states: ENABLED, DISABLED

Type: String

Valid Values: `ENABLED | DISABLED`

Required: Yes

PointInTimeRecoveryDescription

The description of the point in time recovery settings applied to the table.

Type: PointInTimeRecoveryDescription object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ContributorInsightsSummary

Service: Amazon DynamoDB

Represents a Contributor Insights summary entry.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ContributorInsightsMode

Indicates the current mode of CloudWatch Contributor Insights, specifying whether it tracks all access and throttled events or throttled events only for the DynamoDB table or index.

Type: String

Valid Values: `ACCESSED_AND_THROTTLED_KEYS | THROTTLED_KEYS`

Required: No

ContributorInsightsStatus

Describes the current status for contributor insights for the given table and index, if applicable.

Type: String

Valid Values: `ENABLING | ENABLED | DISABLING | DISABLED | FAILED`

Required: No

IndexName

Name of the index associated with the summary, if any.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

TableName

Name of the table associated with the summary.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CreateGlobalSecondaryIndexAction

Service: Amazon DynamoDB

Represents a new global secondary index to be added to an existing table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index to be created.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

KeySchema

The key schema for the global secondary index.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: Yes

Projection

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

Type: Projection object

Required: Yes

OnDemandThroughput

The maximum number of read and write units for the global secondary index being created. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both. You must use either `OnDemandThroughput` or `ProvisionedThroughput` based on your table's capacity mode.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughput

Represents the provisioned throughput settings for the specified global secondary index.

For current minimum and maximum provisioned throughput values, see Service, Account, and Table Quotas in the *Amazon DynamoDB Developer Guide*.

Type: ProvisionedThroughput object

Required: No

WarmThroughput

Represents the warm throughput value (in read units per second and write units per second) when creating a secondary index.

Type: WarmThroughput object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CreateGlobalTableWitnessGroupMemberAction

Service: Amazon DynamoDB

Specifies the action to add a new witness Region to a MRSC global table. A MRSC global table can be configured with either three replicas, or with two replicas and one witness.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The AWS Region name to be added as a witness Region for the MRSC global table. The witness must be in a different Region than the replicas and within the same Region set:

- US Region set: US East (N. Virginia), US East (Ohio), US West (Oregon)
- EU Region set: Europe (Ireland), Europe (London), Europe (Paris), Europe (Frankfurt)
- AP Region set: Asia Pacific (Tokyo), Asia Pacific (Seoul), Asia Pacific (Osaka)

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CreateReplicaAction

Service: Amazon DynamoDB

Represents a replica to be added.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region of the replica to be added.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CreateReplicationGroupMemberAction

Service: Amazon DynamoDB

Represents a replica to be created.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region where the new replica will be created.

Type: String

Required: Yes

GlobalSecondaryIndexes

Replica-specific global secondary index settings.

Type: Array of ReplicaGlobalSecondaryIndex objects

Array Members: Minimum number of 1 item.

Required: No

KMSMasterKeyId

The AWS KMS key that should be used for AWS KMS encryption in the new replica. To specify a key, use its key ID, Amazon Resource Name (ARN), alias name, or alias ARN. Note that you should only provide this parameter if the key is different from the default DynamoDB KMS key `alias/aws/dynamodb`.

Type: String

Required: No

OnDemandThroughputOverride

The maximum on-demand throughput settings for the specified replica table being created. You can only modify `MaxReadRequestUnits`, because you can't modify `MaxWriteRequestUnits` for individual replica tables.

Type: OnDemandThroughputOverride object

Required: No

ProvisionedThroughputOverride

Replica-specific provisioned throughput. If not specified, uses the source table's provisioned throughput settings.

Type: ProvisionedThroughputOverride object

Required: No

TableClassOverride

Replica-specific table class. If not specified, uses the source table's table class.

Type: String

Valid Values: `STANDARD | STANDARD_INFREQUENT_ACCESS`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CsvOptions

Service: Amazon DynamoDB

Processing options for the CSV file being imported.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Delimiter

The delimiter used for separating items in the CSV file being imported.

Type: String

Length Constraints: Fixed length of 1.

Pattern: `[,;:|\t ]`

Required: No

HeaderList

List of the headers used to specify a common header for all source CSV files being imported. If this field is specified then the first line of each CSV file is treated as data instead of the header. If this field is not specified the the first line of each CSV file is treated as the header.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 255 items.

Length Constraints: Minimum length of 1. Maximum length of 65536.

Pattern: `[\x20-\x21\x23-\x2B\x2D-\x7E]*`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Delete

Service: Amazon DynamoDB

Represents a request to perform a `DeleteItem` operation.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Key

The primary key of the item to be deleted. Each element consists of an attribute name and a value for that attribute.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

Name of the table in which the item to be deleted resides. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ConditionExpression

A condition that must be satisfied in order for a conditional delete to succeed.

Type: String

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression.

Type: String to AttributeValue object map

Required: No

ReturnValuesOnConditionCheckFailure

Use `ReturnValuesOnConditionCheckFailure` to get the item attributes if the `Delete` condition fails. For `ReturnValuesOnConditionCheckFailure`, the valid values are: NONE and ALL_OLD.

Type: String

Valid Values: `ALL_OLD | NONE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeleteGlobalSecondaryIndexAction

Service: Amazon DynamoDB

Represents a global secondary index to be deleted from an existing table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index to be deleted.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeleteGlobalTableWitnessGroupMemberAction

Service: Amazon DynamoDB

Specifies the action to remove a witness Region from a MRSC global table. You cannot delete a single witness from a MRSC global table - you must delete both a replica and the witness together. The deletion of both a witness and replica converts the remaining replica to a single-Region DynamoDB table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The witness Region name to be removed from the MRSC global table.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeleteReplicaAction

Service: Amazon DynamoDB

Represents a replica to be removed.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region of the replica to be removed.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeleteReplicationGroupMemberAction

Service: Amazon DynamoDB

Represents a replica to be deleted.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region where the replica exists.

Type: String

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DeleteRequest

Service: Amazon DynamoDB

Represents a request to perform a `DeleteItem` operation on an item.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Key

A map of attribute name to attribute values, representing the primary key of the item to delete. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EnableKinesisStreamingConfiguration

Service: Amazon DynamoDB

Enables setting the configuration for Kinesis Streaming.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ApproximateCreationDateTimePrecision

Toggle for the precision of Kinesis data stream timestamp. The values are either `MILLISECOND` or `MICROSECOND`.

Type: String

Valid Values: `MILLISECOND | MICROSECOND`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Endpoint

Service: Amazon DynamoDB

An endpoint information details.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Address

IP address of the endpoint.

Type: String

Required: Yes

CachePeriodInMinutes

Endpoint cache time to live (TTL) value.

Type: Long

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ExpectedAttributeValue

Service: Amazon DynamoDB

Represents a condition to be compared with an attribute value. This condition can be used with `DeleteItem`, `PutItem`, or `UpdateItem` operations; if the comparison evaluates to true, the operation succeeds; if not, the operation fails. You can use `ExpectedAttributeValue` in one of two different ways:

- Use `AttributeValueList` to specify one or more values to compare against an attribute. Use `ComparisonOperator` to specify how you want to perform the comparison. If the comparison evaluates to true, then the conditional operation succeeds.
- Use `Value` to specify a value that DynamoDB will compare against an attribute. If the values match, then `ExpectedAttributeValue` evaluates to true and the conditional operation succeeds. Optionally, you can also set `Exists` to false, indicating that you *do not* expect to find the attribute value in the table. In this case, the conditional operation succeeds only if the comparison evaluates to false.

`Value` and `Exists` are incompatible with `AttributeValueList` and `ComparisonOperator`. Note that if you use both sets of parameters at once, DynamoDB will return a `ValidationException` exception.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AttributeValueList

One or more values to evaluate against the supplied attribute. The number of values in the list depends on the `ComparisonOperator` being used.

For type Number, value comparisons are numeric.

String value comparisons for greater than, equals, or less than are based on ASCII character code values. For example, a is greater than A, and a is greater than B. For a list of code values, see http://en.wikipedia.org/wiki/ASCII#ASCII_printable_characters.

For Binary, DynamoDB treats each byte of the binary data as unsigned when it compares binary values.

For information on specifying data types in JSON, see JSON Data Format in the *Amazon DynamoDB Developer Guide*.

Type: Array of AttributeValue objects

Required: No

ComparisonOperator

A comparator for evaluating attributes in the `AttributeValueList`. For example, equals, greater than, less than, etc.

The following comparison operators are available:

```
EQ | NE | LE | LT | GE | GT | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS
| BEGINS_WITH | IN | BETWEEN
```

The following are descriptions of each comparison operator.

- `EQ` : Equal. `EQ` is supported for all data types, including lists and maps.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not equal `{"NS":["6", "2", "1"]}`.

- `NE` : Not equal. `NE` is supported for all data types, including lists and maps.

  `AttributeValueList` can contain only one `AttributeValue` of type String, Number, Binary, String Set, Number Set, or Binary Set. If an item contains an `AttributeValue` of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not equal `{"NS":["6", "2", "1"]}`.

- `LE` : Less than or equal.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If an item contains an `AttributeValue` element of a

different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`.

- `LT` : Less than.

  `AttributeValueList` can contain only one `AttributeValue` of type String, Number, or Binary (not a set type). If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`.

- `GE` : Greater than or equal.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`.

- `GT` : Greater than.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not equal `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`.

- `NOT_NULL` : The attribute exists. `NOT_NULL` is supported for all data types, including lists and maps.

  > **Note**
  >
  > This operator tests for the existence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using `NOT_NULL`, the result is a Boolean `true`. This result is because the attribute "a" exists; its data type is not relevant to the `NOT_NULL` comparison operator.

- `NULL` : The attribute does not exist. `NULL` is supported for all data types, including lists and maps.

  > **Note**
  >
  > This operator tests for the nonexistence of an attribute, not its data type. If the data type of attribute "a" is null, and you evaluate it using `NULL`, the result is a Boolean `false`. This is because the attribute "a" exists; its data type is not relevant to the `NULL` comparison operator.

- `CONTAINS` : Checks for a subsequence, or value in a set.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is of type String, then the operator checks for a substring match. If the target attribute of the comparison is of type Binary, then the operator looks for a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("`SS`", "`NS`", or "`BS`"), then the operator evaluates to true if it finds an exact match with any member of the set.

  CONTAINS is supported for lists: When evaluating "a `CONTAINS` b", "a" can be a list; however, "b" cannot be a set, a map, or a list.

- `NOT_CONTAINS` : Checks for absence of a subsequence, or absence of a value in a set.

  `AttributeValueList` can contain only one `AttributeValue` element of type String, Number, or Binary (not a set type). If the target attribute of the comparison is a String, then the operator checks for the absence of a substring match. If the target attribute of the comparison is Binary, then the operator checks for the absence of a subsequence of the target that matches the input. If the target attribute of the comparison is a set ("`SS`", "`NS`", or "`BS`"), then the operator evaluates to true if it *does not* find an exact match with any member of the set.

  NOT_CONTAINS is supported for lists: When evaluating "a `NOT CONTAINS` b", "a" can be a list; however, "b" cannot be a set, a map, or a list.

- `BEGINS_WITH` : Checks for a prefix.

  `AttributeValueList` can contain only one `AttributeValue` of type String or Binary (not a Number or a set type). The target attribute of the comparison must be of type String or Binary (not a Number or a set type).

- `IN` : Checks for matching elements in a list.

  `AttributeValueList` can contain one or more `AttributeValue` elements of type String, Number, or Binary. These attributes are compared against an existing attribute of an item. If any elements of the input are equal to the item attribute, the expression evaluates to true.
- `BETWEEN` : Greater than or equal to the first value, and less than or equal to the second value.

  `AttributeValueList` must contain two `AttributeValue` elements of the same type, either String, Number, or Binary (not a set type). A target attribute matches if the target value is greater than, or equal to, the first element and less than, or equal to, the second element. If an item contains an `AttributeValue` element of a different type than the one provided in the request, the value does not match. For example, `{"S":"6"}` does not compare to `{"N":"6"}`. Also, `{"N":"6"}` does not compare to `{"NS":["6", "2", "1"]}`

Type: String

Valid Values: `EQ | NE | IN | LE | LT | GE | GT | BETWEEN | NOT_NULL | NULL | CONTAINS | NOT_CONTAINS | BEGINS_WITH`

Required: No

Exists

Causes DynamoDB to evaluate the value before attempting a conditional operation:

- If `Exists` is `true`, DynamoDB will check to see if that attribute value already exists in the table. If it is found, then the operation succeeds. If it is not found, the operation fails with a `ConditionCheckFailedException`.
- If `Exists` is `false`, DynamoDB assumes that the attribute value does not exist in the table. If in fact the value does not exist, then the assumption is valid and the operation succeeds. If the value is found, despite the assumption that it does not exist, the operation fails with a `ConditionCheckFailedException`.

The default setting for `Exists` is `true`. If you supply a `Value` all by itself, DynamoDB assumes the attribute exists: You don't have to set `Exists` to `true`, because it is implied.

DynamoDB returns a `ValidationException` if:

- `Exists` is `true` but there is no `Value` to check. (You expect a value to exist, but don't specify what that value is.)

- `Exists` is `false` but you also provide a `Value`. (You cannot expect an attribute to have a value, while also expecting it not to exist.)

Type: Boolean

Required: No

Value

Represents the data for the expected attribute.

Each attribute value is described as a name-value pair. The name is the data type, and the value is the data itself.

For more information, see Data Types in the *Amazon DynamoDB Developer Guide*.

Type: AttributeValue object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ExportDescription

Service: Amazon DynamoDB

Represents the properties of the exported table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

BilledSizeBytes

The billable size of the table export.

Type: Long

Valid Range: Minimum value of 0.

Required: No

ClientToken

The client token that was provided for the export task. A client token makes calls to `ExportTableToPointInTimeInput` idempotent, meaning that multiple identical calls have the same effect as one single call.

Type: String

Pattern: `^[^\$]+$`

Required: No

EndTime

The time at which the export task completed.

Type: Timestamp

Required: No

ExportArn

The Amazon Resource Name (ARN) of the table export.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

ExportFormat

The format of the exported data. Valid values for `ExportFormat` are `DYNAMODB_JSON` or `ION`.

Type: String

Valid Values: `DYNAMODB_JSON | ION`

Required: No

ExportManifest

The name of the manifest file for the export task.

Type: String

Required: No

ExportStatus

Export can be in one of the following states: IN_PROGRESS, COMPLETED, or FAILED.

Type: String

Valid Values: `IN_PROGRESS | COMPLETED | FAILED`

Required: No

ExportTime

Point in time from which table data was exported.

Type: Timestamp

Required: No

ExportType

The type of export that was performed. Valid values are `FULL_EXPORT` or `INCREMENTAL_EXPORT`.

Type: String

Valid Values: `FULL_EXPORT | INCREMENTAL_EXPORT`

Required: No

FailureCode

Status code for the result of the failed export.

Type: String

Required: No

FailureMessage

Export failure reason description.

Type: String

Required: No

IncrementalExportSpecification

Optional object containing the parameters specific to an incremental export.

Type: IncrementalExportSpecification object

Required: No

ItemCount

The number of items exported.

Type: Long

Valid Range: Minimum value of 0.

Required: No

S3Bucket

The name of the Amazon S3 bucket containing the export.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `^[a-z0-9A-Z]+[\.\-\w]*[a-z0-9A-Z]+$`

Required: No

S3BucketOwner

The ID of the AWS account that owns the bucket containing the export.

Type: String

Pattern: `[0-9]{12}`

Required: No

S3Prefix

The Amazon S3 bucket prefix used as the file name and path of the exported snapshot.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

S3SseAlgorithm

Type of encryption used on the bucket where export data is stored. Valid values for `S3SseAlgorithm` are:

- `AES256` - server-side encryption with Amazon S3 managed keys
- `KMS` - server-side encryption with AWS KMS managed keys

Type: String

Valid Values: `AES256` | `KMS`

Required: No

S3SseKmsKeyId

The ID of the AWS KMS managed key used to encrypt the S3 bucket where export data is stored (if applicable).

Type: String

Length Constraints: Minimum length of 1. Maximum length of 2048.

Required: No

StartTime

The time at which the export task began.

Type: Timestamp

Required: No

TableArn

The Amazon Resource Name (ARN) of the table that was exported.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

TableId

Unique ID of the table that was exported.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ExportSummary

Service: Amazon DynamoDB

Summary information about an export task.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ExportArn

The Amazon Resource Name (ARN) of the export.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

ExportStatus

Export can be in one of the following states: IN_PROGRESS, COMPLETED, or FAILED.

Type: String

Valid Values: `IN_PROGRESS | COMPLETED | FAILED`

Required: No

ExportType

The type of export that was performed. Valid values are `FULL_EXPORT` or `INCREMENTAL_EXPORT`.

Type: String

Valid Values: `FULL_EXPORT | INCREMENTAL_EXPORT`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

FailureException

Service: Amazon DynamoDB

Represents a failure a contributor insights operation.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ExceptionDescription

Description of the failure.

Type: String

Required: No

ExceptionName

Exception name.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Get

Service: Amazon DynamoDB

Specifies an item and related attribute values to retrieve in a `TransactGetItem` object.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Key

A map of attribute names to `AttributeValue` objects that specifies the primary key of the item to retrieve.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

The name of the table from which to retrieve the specified item. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ExpressionAttributeNames

One or more substitution tokens for attribute names in the ProjectionExpression parameter.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ProjectionExpression

A string that identifies one or more attributes of the specified item to retrieve from the table. The attributes in the expression must be separated by commas. If no attribute names are specified, then all attributes of the specified item are returned. If any of the requested attributes are not found, they do not appear in the result.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalSecondaryIndex

Service: Amazon DynamoDB

Represents the properties of a global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index. The name must be unique among all other indexes on this table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

KeySchema

The complete key schema for a global secondary index, which consists of one or more pairs of attribute names and key types:

- `HASH` - partition key
- `RANGE` - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.
> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: Yes

Projection

Represents attributes that are copied (projected) from the table into the global secondary index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

Type: Projection object

Required: Yes

OnDemandThroughput

The maximum number of read and write units for the specified global secondary index. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both. You must use either `OnDemandThroughput` or `ProvisionedThroughput` based on your table's capacity mode.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughput

Represents the provisioned throughput settings for the specified global secondary index. You must use either `OnDemandThroughput` or `ProvisionedThroughput` based on your table's capacity mode.

For current minimum and maximum provisioned throughput values, see Service, Account, and Table Quotas in the *Amazon DynamoDB Developer Guide*.

Type: ProvisionedThroughput object

Required: No

WarmThroughput

Represents the warm throughput value (in read units per second and write units per second) for the specified secondary index. If you use this parameter, you must specify `ReadUnitsPerSecond`, `WriteUnitsPerSecond`, or both.

Type: WarmThroughput object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalSecondaryIndexAutoScalingUpdate

Service: Amazon DynamoDB

Represents the auto scaling settings of a global secondary index for a global table that will be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

ProvisionedWriteCapacityAutoScalingUpdate

Represents the auto scaling settings to be modified for a global table or global secondary index.

Type: AutoScalingSettingsUpdate object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalSecondaryIndexDescription

Service: Amazon DynamoDB

Represents the properties of a global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Backfilling

Indicates whether the index is currently backfilling. *Backfilling* is the process of reading items from the table and determining whether they can be added to the index. (Not all items will qualify: For example, a partition key cannot have any duplicate values.) If an item can be added to the index, DynamoDB will do so. After all items have been processed, the backfilling operation is complete and `Backfilling` is false.

You can delete an index that is being created during the `Backfilling` phase when `IndexStatus` is set to CREATING and `Backfilling` is true. You can't delete the index that is being created when `IndexStatus` is set to CREATING and `Backfilling` is false.

> **Note**
>
> For indexes that were created during a `CreateTable` operation, the `Backfilling` attribute does not appear in the `DescribeTable` output.

Type: Boolean

Required: No

IndexArn

The Amazon Resource Name (ARN) that uniquely identifies the index.

Type: String

Required: No

IndexName

The name of the global secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

IndexSizeBytes

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

Type: Long

Required: No

IndexStatus

The current state of the global secondary index:

- `CREATING` - The index is being created.
- `UPDATING` - The index is being updated.
- `DELETING` - The index is being deleted.
- `ACTIVE` - The index is ready for use.

Type: String

Valid Values: `CREATING | UPDATING | DELETING | ACTIVE`

Required: No

ItemCount

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

Type: Long

Required: No

KeySchema

The complete key schema for a global secondary index, which consists of one or more pairs of attribute names and key types:

- `HASH` - partition key
- `RANGE` - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.
> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: No

OnDemandThroughput

The maximum number of read and write units for the specified global secondary index. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

Projection

Represents attributes that are copied (projected) from the table into the global secondary index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

Type: Projection object

Required: No

ProvisionedThroughput

Represents the provisioned throughput settings for the specified global secondary index.

For current minimum and maximum provisioned throughput values, see Service, Account, and Table Quotas in the *Amazon DynamoDB Developer Guide*.

Type: ProvisionedThroughputDescription object

Required: No

WarmThroughput

Represents the warm throughput value (in read units per second and write units per second) for the specified secondary index.

Type: GlobalSecondaryIndexWarmThroughputDescription object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalSecondaryIndexInfo

Service: Amazon DynamoDB

Represents the properties of a global secondary index for the table when the backup was created.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

KeySchema

The complete key schema for a global secondary index, which consists of one or more pairs of attribute names and key types:

- HASH - partition key
- RANGE - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.
> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: No

OnDemandThroughput

Sets the maximum number of read and write units for the specified on-demand table. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

Projection

Represents attributes that are copied (projected) from the table into the global secondary index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

Type: Projection object

Required: No

ProvisionedThroughput

Represents the provisioned throughput settings for the specified global secondary index.

Type: ProvisionedThroughput object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalSecondaryIndexUpdate

Service: Amazon DynamoDB

Represents one of the following:

- A new global secondary index to be added to an existing table.
- New provisioned throughput parameters for an existing global secondary index.
- An existing global secondary index to be removed from an existing table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Create

The parameters required for creating a global secondary index on an existing table:

- `IndexName`
- `KeySchema`
- `AttributeDefinitions`
- `Projection`
- `ProvisionedThroughput`

Type: CreateGlobalSecondaryIndexAction object

Required: No

Delete

The name of an existing global secondary index to be removed.

Type: DeleteGlobalSecondaryIndexAction object

Required: No

Update

The name of an existing global secondary index, along with new provisioned throughput settings to be applied to that index.

Type: UpdateGlobalSecondaryIndexAction object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalSecondaryIndexWarmThroughputDescription

Service: Amazon DynamoDB

The description of the warm throughput value on a global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ReadUnitsPerSecond

Represents warm throughput read units per second value for a global secondary index.

Type: Long

Valid Range: Minimum value of 1.

Required: No

Status

Represents the warm throughput status being created or updated on a global secondary index. The status can only be `UPDATING` or `ACTIVE`.

Type: String

Valid Values: `CREATING | UPDATING | DELETING | ACTIVE`

Required: No

WriteUnitsPerSecond

Represents warm throughput write units per second value for a global secondary index.

Type: Long

Valid Range: Minimum value of 1.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalTable

Service: Amazon DynamoDB

Represents the properties of a global table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

GlobalTableName

The global table name.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

ReplicationGroup

The Regions where the global table has replicas.

Type: Array of Replica objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalTableDescription

Service: Amazon DynamoDB

Contains details about the global table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

CreationDateTime

The creation time of the global table.

Type: Timestamp

Required: No

GlobalTableArn

The unique identifier of the global table.

Type: String

Required: No

GlobalTableName

The global table name.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

GlobalTableStatus

The current state of the global table:

- `CREATING` - The global table is being created.

- `UPDATING` - The global table is being updated.
- `DELETING` - The global table is being deleted.
- `ACTIVE` - The global table is ready for use.

Type: String

Valid Values: `CREATING | ACTIVE | DELETING | UPDATING`

Required: No

ReplicationGroup

The Regions where the global table has replicas.

Type: Array of ReplicaDescription objects

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalTableGlobalSecondaryIndexSettingsUpdate

Service: Amazon DynamoDB

Represents the settings of a global secondary index for a global table that will be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index. The name must be unique among all other indexes on this table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

ProvisionedWriteCapacityAutoScalingSettingsUpdate

Auto scaling settings for managing a global secondary index's write capacity units.

Type: AutoScalingSettingsUpdate object

Required: No

ProvisionedWriteCapacityUnits

The maximum number of writes consumed per second before DynamoDB returns a `ThrottlingException`.

Type: Long

Valid Range: Minimum value of 1.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalTableWitnessDescription

Service: Amazon DynamoDB

Represents the properties of a witness Region in a MRSC global table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The name of the AWS Region that serves as a witness for the MRSC global table.

Type: String

Required: No

WitnessStatus

The current status of the witness Region in the MRSC global table.

Type: String

Valid Values: `CREATING | DELETING | ACTIVE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

GlobalTableWitnessGroupUpdate

Service: Amazon DynamoDB

Represents one of the following:

- A new witness to be added to a new global table.
- An existing witness to be removed from an existing global table.

You can configure one witness per MRSC global table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Create

Specifies a witness Region to be added to a new MRSC global table. The witness must be added when creating the MRSC global table.

Type: CreateGlobalTableWitnessGroupMemberAction object

Required: No

Delete

Specifies a witness Region to be removed from an existing global table. Must be done in conjunction with removing a replica. The deletion of both a witness and replica converts the remaining replica to a single-Region DynamoDB table.

Type: DeleteGlobalTableWitnessGroupMemberAction object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ImportSummary

Service: Amazon DynamoDB

Summary information about the source file for the import.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

CloudWatchLogGroupArn

The Amazon Resource Number (ARN) of the Cloudwatch Log Group associated with this import task.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

EndTime

The time at which this import task ended. (Does this include the successful complete creation of the table it was imported to?)

Type: Timestamp

Required: No

ImportArn

The Amazon Resource Number (ARN) corresponding to the import request.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

ImportStatus

The status of the import operation.

Type: String

Valid Values: `IN_PROGRESS | COMPLETED | CANCELLING | CANCELLED | FAILED`

Required: No

InputFormat

The format of the source data. Valid values are `CSV`, `DYNAMODB_JSON` or `ION`.

Type: String

Valid Values: `DYNAMODB_JSON | ION | CSV`

Required: No

S3BucketSource

The path and S3 bucket of the source file that is being imported. This includes the S3Bucket (required), S3KeyPrefix (optional) and S3BucketOwner (optional if the bucket is owned by the requester).

Type: S3BucketSource object

Required: No

StartTime

The time at which this import task began.

Type: Timestamp

Required: No

TableArn

The Amazon Resource Number (ARN) of the table being imported into.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ImportTableDescription

Service: Amazon DynamoDB

Represents the properties of the table being imported into.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ClientToken

The client token that was provided for the import task. Reusing the client token on retry makes a call to `ImportTable` idempotent.

Type: String

Pattern: `^[^\$]+$`

Required: No

CloudWatchLogGroupArn

The Amazon Resource Number (ARN) of the Cloudwatch Log Group associated with the target table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

EndTime

The time at which the creation of the table associated with this import task completed.

Type: Timestamp

Required: No

ErrorCount

The number of errors occurred on importing the source file into the target table.

Type: Long

Valid Range: Minimum value of 0.

Required: No

FailureCode

The error code corresponding to the failure that the import job ran into during execution.

Type: String

Required: No

FailureMessage

The error message corresponding to the failure that the import job ran into during execution.

Type: String

Required: No

ImportArn

The Amazon Resource Number (ARN) corresponding to the import request.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

ImportedItemCount

The number of items successfully imported into the new table.

Type: Long

Valid Range: Minimum value of 0.

Required: No

ImportStatus

The status of the import.

Type: String

Valid Values: `IN_PROGRESS | COMPLETED | CANCELLING | CANCELLED | FAILED`

Required: No

InputCompressionType

The compression options for the data that has been imported into the target table. The values are NONE, GZIP, or ZSTD.

Type: String

Valid Values: `GZIP | ZSTD | NONE`

Required: No

InputFormat

The format of the source data going into the target table.

Type: String

Valid Values: `DYNAMODB_JSON | ION | CSV`

Required: No

InputFormatOptions

The format options for the data that was imported into the target table. There is one value, CsvOption.

Type: InputFormatOptions object

Required: No

ProcessedItemCount

The total number of items processed from the source file.

Type: Long

Valid Range: Minimum value of 0.

Required: No

ProcessedSizeBytes

The total size of data processed from the source file, in Bytes.

Type: Long

Required: No

S3BucketSource

Values for the S3 bucket the source file is imported from. Includes bucket name (required), key prefix (optional) and bucket account owner ID (optional).

Type: S3BucketSource object

Required: No

StartTime

The time when this import task started.

Type: Timestamp

Required: No

TableArn

The Amazon Resource Number (ARN) of the table being imported into.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

TableCreationParameters

The parameters for the new table that is being imported into.

Type: TableCreationParameters object

Required: No

TableId

The table id corresponding to the table created by import table process.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

IncrementalExportSpecification

Service: Amazon DynamoDB

Optional object containing the parameters specific to an incremental export.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ExportFromTime

Time in the past which provides the inclusive start range for the export table's data, counted in seconds from the start of the Unix epoch. The incremental export will reflect the table's state including and after this point in time.

Type: Timestamp

Required: No

ExportToTime

Time in the past which provides the exclusive end range for the export table's data, counted in seconds from the start of the Unix epoch. The incremental export will reflect the table's state just prior to this point in time. If this is not provided, the latest time with data available will be used.

Type: Timestamp

Required: No

ExportViewType

The view type that was chosen for the export. Valid values are `NEW_AND_OLD_IMAGES` and `NEW_IMAGES`. The default value is `NEW_AND_OLD_IMAGES`.

Type: String

Valid Values: `NEW_IMAGE | NEW_AND_OLD_IMAGES`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

InputFormatOptions

Service: Amazon DynamoDB

The format options for the data that was imported into the target table. There is one value, CsvOption.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Csv

The options for imported source files in CSV format. The values are Delimiter and HeaderList.

Type: CsvOptions object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ItemCollectionMetrics

Service: Amazon DynamoDB

Information about item collections, if any, that were affected by the operation. `ItemCollectionMetrics` is only returned if the request asked for it. If the table does not have any local secondary indexes, this information is not returned in the response.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ItemCollectionKey

The partition key value of the item collection. This value is the same as the partition key value of the item.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

SizeEstimateRangeGB

An estimate of item collection size, in gigabytes. This value is a two-element array containing a lower bound and an upper bound for the estimate. The estimate includes the size of all the items in the table, plus the size of all attributes projected into all of the local secondary indexes on that table. Use this estimate to measure whether a local secondary index is approaching its size limit.

The estimate is subject to change over time; therefore, do not rely on the precision or accuracy of the estimate.

Type: Array of doubles

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ItemResponse

Service: Amazon DynamoDB

Details for the requested item.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Item

Map of attribute data consisting of the data type and attribute value.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KeysAndAttributes

Service: Amazon DynamoDB

Represents a set of primary keys and, for each key, the attributes to retrieve from the table.

For each primary key, you must provide *all* of the key attributes. For example, with a simple primary key, you only need to provide the partition key. For a composite primary key, you must provide *both* the partition key and the sort key.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Keys

The primary key attribute values that define the items and the attributes associated with the items.

Type: Array of string to AttributeValue object maps

Array Members: Minimum number of 1 item. Maximum number of 100 items.

Key Length Constraints: Maximum length of 65535.

Required: Yes

AttributesToGet

This is a legacy parameter. Use `ProjectionExpression` instead. For more information, see Legacy Conditional Parameters in the *Amazon DynamoDB Developer Guide*.

Type: Array of strings

Array Members: Minimum number of 1 item.

Length Constraints: Maximum length of 65535.

Required: No

ConsistentRead

The consistency of a read operation. If set to `true`, then a strongly consistent read is used; otherwise, an eventually consistent read is used.

Type: Boolean

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression. The following are some use cases for using `ExpressionAttributeNames`:

- To access an attribute whose name conflicts with a DynamoDB reserved word.
- To create a placeholder for repeating occurrences of an attribute name in an expression.
- To prevent special characters in an attribute name from being misinterpreted in an expression.

Use the **#** character in an expression to dereference an attribute name. For example, consider the following attribute name:

- `Percentile`

The name of this attribute conflicts with a reserved word, so it cannot be used directly in an expression. (For the complete list of reserved words, see Reserved Words in the *Amazon DynamoDB Developer Guide*). To work around this, you could specify the following for `ExpressionAttributeNames`:

- `{"#P":"Percentile"}`

You could then use this substitution in an expression, as in this example:

- `#P = :val`

> **Note**
>
> Tokens that begin with the **:** character are *expression attribute values*, which are placeholders for the actual value at runtime.

For more information on expression attribute names, see Accessing Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ProjectionExpression

A string that identifies one or more attributes to retrieve from the table. These attributes can include scalars, sets, or elements of a JSON document. The attributes in the `ProjectionExpression` must be separated by commas.

If no attribute names are specified, then all attributes will be returned. If any of the requested attributes are not found, they will not appear in the result.

For more information, see Accessing Item Attributes in the *Amazon DynamoDB Developer Guide*.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KeySchemaElement

Service: Amazon DynamoDB

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A `KeySchemaElement` represents exactly one attribute of the primary key. For example, a simple primary key would be represented by one `KeySchemaElement` (for the partition key). A composite primary key would require one `KeySchemaElement` for the partition key, and another `KeySchemaElement` for the sort key.

A `KeySchemaElement` must be a scalar, top-level attribute (not a nested attribute). The data type must be one of String, Number, or Binary. The attribute cannot be nested within a List or a Map.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AttributeName

The name of a key attribute.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

KeyType

The role that this key attribute will assume:

- `HASH` - partition key
- `RANGE` - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.

> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Type: String

Valid Values: `HASH` | `RANGE`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KinesisDataStreamDestination

Service: Amazon DynamoDB

Describes a Kinesis data stream destination.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ApproximateCreationDateTimePrecision

The precision of the Kinesis data stream timestamp. The values are either `MILLISECOND` or `MICROSECOND`.

Type: String

Valid Values: `MILLISECOND | MICROSECOND`

Required: No

DestinationStatus

The current status of replication.

Type: String

Valid Values: `ENABLING | ACTIVE | DISABLING | DISABLED | ENABLE_FAILED | UPDATING`

Required: No

DestinationStatusDescription

The human-readable string that corresponds to the replica status.

Type: String

Required: No

StreamArn

The ARN for a specific Kinesis data stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LocalSecondaryIndex

Service: Amazon DynamoDB

Represents the properties of a local secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the local secondary index. The name must be unique among all other indexes on this table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

KeySchema

The complete key schema for the local secondary index, consisting of one or more pairs of attribute names and key types:

- `HASH` - partition key
- `RANGE` - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.
> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: Yes

Projection

Represents attributes that are copied (projected) from the table into the local secondary index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

Type: Projection object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LocalSecondaryIndexDescription

Service: Amazon DynamoDB

Represents the properties of a local secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexArn

The Amazon Resource Name (ARN) that uniquely identifies the index.

Type: String

Required: No

IndexName

Represents the name of the local secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

IndexSizeBytes

The total size of the specified index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

Type: Long

Required: No

ItemCount

The number of items in the specified index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

Type: Long

Required: No

KeySchema

The complete key schema for the local secondary index, consisting of one or more pairs of attribute names and key types:

- HASH - partition key
- RANGE - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.
> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: No

Projection

Represents attributes that are copied (projected) from the table into the global secondary index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

Type: Projection object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

LocalSecondaryIndexInfo

Service: Amazon DynamoDB

Represents the properties of a local secondary index for the table when the backup was created.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

Represents the name of the local secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

KeySchema

The complete key schema for a local secondary index, which consists of one or more pairs of attribute names and key types:

- HASH - partition key
- RANGE - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.
> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: No

Projection

Represents attributes that are copied (projected) from the table into the global secondary index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

Type: Projection object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

OnDemandThroughput

Service: Amazon DynamoDB

Sets the maximum number of read and write units for the specified on-demand table. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

MaxReadRequestUnits

Maximum number of read request units for the specified table.

To specify a maximum `OnDemandThroughput` on your table, set the value of `MaxReadRequestUnits` as greater than or equal to 1. To remove the maximum `OnDemandThroughput` that is currently set on your table, set the value of `MaxReadRequestUnits` to -1.

Type: Long

Required: No

MaxWriteRequestUnits

Maximum number of write request units for the specified table.

To specify a maximum `OnDemandThroughput` on your table, set the value of `MaxWriteRequestUnits` as greater than or equal to 1. To remove the maximum `OnDemandThroughput` that is currently set on your table, set the value of `MaxWriteRequestUnits` to -1.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

OnDemandThroughputOverride

Service: Amazon DynamoDB

Overrides the on-demand throughput settings for this replica table. If you don't specify a value for this parameter, it uses the source table's on-demand throughput settings.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

MaxReadRequestUnits

Maximum number of read request units for the specified replica table.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ParameterizedStatement

Service: Amazon DynamoDB

Represents a PartiQL statement that uses parameters.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Statement

A PartiQL statement that uses parameters.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 8192.

Required: Yes

Parameters

The parameter values.

Type: Array of AttributeValue objects

Array Members: Minimum number of 1 item.

Required: No

ReturnValuesOnConditionCheckFailure

An optional parameter that returns the item attributes for a PartiQL `ParameterizedStatement` operation that failed a condition check.

There is no additional cost associated with requesting a return value aside from the small network and processing overhead of receiving a larger response. No read capacity units are consumed.

Type: String

Valid Values: `ALL_OLD` | `NONE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PointInTimeRecoveryDescription

Service: Amazon DynamoDB

The description of the point in time settings applied to the table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

EarliestRestorableDateTime

Specifies the earliest point in time you can restore your table to. You can restore your table to any point in time during the last 35 days.

Type: Timestamp

Required: No

LatestRestorableDateTime

`LatestRestorableDateTime` is typically 5 minutes before the current time.

Type: Timestamp

Required: No

PointInTimeRecoveryStatus

The current state of point in time recovery:

- `ENABLED` - Point in time recovery is enabled.
- `DISABLED` - Point in time recovery is disabled.

Type: String

Valid Values: `ENABLED | DISABLED`

Required: No

RecoveryPeriodInDays

The number of preceding days for which continuous backups are taken and maintained. Your table data is only recoverable to any point-in-time from within the configured recovery period. This parameter is optional.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 35.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PointInTimeRecoverySpecification

Service: Amazon DynamoDB

Represents the settings used to enable point in time recovery.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

PointInTimeRecoveryEnabled

Indicates whether point in time recovery is enabled (true) or disabled (false) on the table.

Type: Boolean

Required: Yes

RecoveryPeriodInDays

The number of preceding days for which continuous backups are taken and maintained. Your table data is only recoverable to any point-in-time from within the configured recovery period. This parameter is optional. If no value is provided, the value will default to 35.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 35.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Projection

Service: Amazon DynamoDB

Represents attributes that are copied (projected) from the table into an index. These are in addition to the primary key attributes and index key attributes, which are automatically projected.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

NonKeyAttributes

Represents the non-key attribute names which will be projected into the index.

For global and local secondary indexes, the total count of `NonKeyAttributes` summed across all of the secondary indexes, must not exceed 100. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total. This limit only applies when you specify the ProjectionType of `INCLUDE`. You still can specify the ProjectionType of `ALL` to project all attributes from the source table, even if the table has more than 100 attributes.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

ProjectionType

The set of attributes that are projected into the index:

- `KEYS_ONLY` - Only the index and primary keys are projected into the index.
- `INCLUDE` - In addition to the attributes described in `KEYS_ONLY`, the secondary index will include other non-key attributes that you specify.
- `ALL` - All of the table attributes are projected into the index.

When using the DynamoDB console, `ALL` is selected by default.

Type: String

Valid Values: `ALL | KEYS_ONLY | INCLUDE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProvisionedThroughput

Service: Amazon DynamoDB

Represents the provisioned throughput settings for the specified global secondary index. You must use `ProvisionedThroughput` or `OnDemandThroughput` based on your table's capacity mode.

For current minimum and maximum provisioned throughput values, see Service, Account, and Table Quotas in the *Amazon DynamoDB Developer Guide*.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ReadCapacityUnits

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a `ThrottlingException`. For more information, see Specifying Read and Write Requirements in the *Amazon DynamoDB Developer Guide*.

If read/write capacity mode is `PAY_PER_REQUEST` the value is set to 0.

Type: Long

Valid Range: Minimum value of 1.

Required: Yes

WriteCapacityUnits

The maximum number of writes consumed per second before DynamoDB returns a `ThrottlingException`. For more information, see Specifying Read and Write Requirements in the *Amazon DynamoDB Developer Guide*.

If read/write capacity mode is `PAY_PER_REQUEST` the value is set to 0.

Type: Long

Valid Range: Minimum value of 1.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProvisionedThroughputDescription

Service: Amazon DynamoDB

Represents the provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

LastDecreaseDateTime

The date and time of the last provisioned throughput decrease for this table.

Type: Timestamp

Required: No

LastIncreaseDateTime

The date and time of the last provisioned throughput increase for this table.

Type: Timestamp

Required: No

NumberOfDecreasesToday

The number of provisioned throughput decreases for this table during this UTC calendar day. For current maximums on provisioned throughput decreases, see Service, Account, and Table Quotas in the *Amazon DynamoDB Developer Guide*.

Type: Long

Valid Range: Minimum value of 1.

Required: No

ReadCapacityUnits

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a `ThrottlingException`. Eventually consistent reads require less effort than strongly

consistent reads, so a setting of 50 `ReadCapacityUnits` per second provides 100 eventually consistent `ReadCapacityUnits` per second.

Type: Long

Valid Range: Minimum value of 0.

Required: No

WriteCapacityUnits

The maximum number of writes consumed per second before DynamoDB returns a `ThrottlingException`.

Type: Long

Valid Range: Minimum value of 0.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ProvisionedThroughputOverride

Service: Amazon DynamoDB

Replica-specific provisioned throughput settings. If not specified, uses the source table's provisioned throughput settings.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ReadCapacityUnits

Replica-specific read capacity units. If not specified, uses the source table's read capacity settings.

Type: Long

Valid Range: Minimum value of 1.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Put

Service: Amazon DynamoDB

Represents a request to perform a `PutItem` operation.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Item

A map of attribute name to attribute values, representing the primary key of the item to be written by `PutItem`. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema. If any attributes are present in the item that are part of an index key schema for the table, their types must match the index key schema.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

Name of the table in which to write the item. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

ConditionExpression

A condition that must be satisfied in order for a conditional update to succeed.

Type: String

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression.

Type: String to AttributeValue object map

Required: No

ReturnValuesOnConditionCheckFailure

Use `ReturnValuesOnConditionCheckFailure` to get the item attributes if the `Put` condition fails. For `ReturnValuesOnConditionCheckFailure`, the valid values are: NONE and ALL_OLD.

Type: String

Valid Values: `ALL_OLD | NONE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

PutRequest

Service: Amazon DynamoDB

Represents a request to perform a `PutItem` operation on an item.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Item

A map of attribute name to attribute values, representing the primary key of an item to be processed by `PutItem`. All of the table's primary key attributes must be specified, and their data types must match those of the table's key schema. If any attributes are present in the item that are part of an index key schema for the table, their types must match the index key schema.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Replica

Service: Amazon DynamoDB

Represents the properties of a replica.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region where the replica needs to be created.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaAutoScalingDescription

Service: Amazon DynamoDB

Represents the auto scaling settings of the replica.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

GlobalSecondaryIndexes

Replica-specific global secondary index auto scaling settings.

Type: Array of ReplicaGlobalSecondaryIndexAutoScalingDescription objects

Required: No

RegionName

The Region where the replica exists.

Type: String

Required: No

ReplicaProvisionedReadCapacityAutoScalingSettings

Represents the auto scaling settings for a global table or global secondary index.

Type: AutoScalingSettingsDescription object

Required: No

ReplicaProvisionedWriteCapacityAutoScalingSettings

Represents the auto scaling settings for a global table or global secondary index.

Type: AutoScalingSettingsDescription object

Required: No

ReplicaStatus

The current state of the replica:

- `CREATING` - The replica is being created.
- `UPDATING` - The replica is being updated.
- `DELETING` - The replica is being deleted.
- `ACTIVE` - The replica is ready for use.

Type: String

Valid Values: `CREATING | CREATION_FAILED | UPDATING | DELETING | ACTIVE | REGION_DISABLED | INACCESSIBLE_ENCRYPTION_CREDENTIALS | ARCHIVING | ARCHIVED | REPLICATION_NOT_AUTHORIZED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaAutoScalingUpdate

Service: Amazon DynamoDB

Represents the auto scaling settings of a replica that will be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region where the replica exists.

Type: String

Required: Yes

ReplicaGlobalSecondaryIndexUpdates

Represents the auto scaling settings of global secondary indexes that will be modified.

Type: Array of ReplicaGlobalSecondaryIndexAutoScalingUpdate objects

Required: No

ReplicaProvisionedReadCapacityAutoScalingUpdate

Represents the auto scaling settings to be modified for a global table or global secondary index.

Type: AutoScalingSettingsUpdate object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaDescription

Service: Amazon DynamoDB

Contains the details of the replica.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

GlobalSecondaryIndexes

Replica-specific global secondary index settings.

Type: Array of ReplicaGlobalSecondaryIndexDescription objects

Required: No

KMSMasterKeyId

The AWS KMS key of the replica that will be used for AWS KMS encryption.

Type: String

Required: No

OnDemandThroughputOverride

Overrides the maximum on-demand throughput settings for the specified replica table.

Type: OnDemandThroughputOverride object

Required: No

ProvisionedThroughputOverride

Replica-specific provisioned throughput. If not described, uses the source table's provisioned throughput settings.

Type: ProvisionedThroughputOverride object

Required: No

RegionName

The name of the Region.

Type: String

Required: No

ReplicaInaccessibleDateTime

The time at which the replica was first detected as inaccessible. To determine cause of inaccessibility check the `ReplicaStatus` property.

Type: Timestamp

Required: No

ReplicaStatus

The current state of the replica:

- `CREATING` - The replica is being created.
- `UPDATING` - The replica is being updated.
- `DELETING` - The replica is being deleted.
- `ACTIVE` - The replica is ready for use.
- `REGION_DISABLED` - The replica is inaccessible because the AWS Region has been disabled.

> **Note**
>
> If the AWS Region remains inaccessible for more than 20 hours, DynamoDB will remove this replica from the replication group. The replica will not be deleted and replication will stop from and to this region.

- `INACCESSIBLE_ENCRYPTION_CREDENTIALS` - The AWS KMS key used to encrypt the table is inaccessible.

> **Note**
>
> If the AWS KMS key remains inaccessible for more than 20 hours, DynamoDB will remove this replica from the replication group. The replica will not be deleted and replication will stop from and to this region.

Type: String

Valid Values: `CREATING | CREATION_FAILED | UPDATING | DELETING | ACTIVE | REGION_DISABLED | INACCESSIBLE_ENCRYPTION_CREDENTIALS | ARCHIVING | ARCHIVED | REPLICATION_NOT_AUTHORIZED`

Required: No

ReplicaStatusDescription

Detailed information about the replica status.

Type: String

Required: No

ReplicaStatusPercentProgress

Specifies the progress of a Create, Update, or Delete action on the replica as a percentage.

Type: String

Required: No

ReplicaTableClassSummary

Contains details of the table class.

Type: TableClassSummary object

Required: No

WarmThroughput

Represents the warm throughput value for this replica.

Type: TableWarmThroughputDescription object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaGlobalSecondaryIndex

Service: Amazon DynamoDB

Represents the properties of a replica global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

OnDemandThroughputOverride

Overrides the maximum on-demand throughput settings for the specified global secondary index in the specified replica table.

Type: OnDemandThroughputOverride object

Required: No

ProvisionedThroughputOverride

Replica table GSI-specific provisioned throughput. If not specified, uses the source table GSI's read capacity settings.

Type: ProvisionedThroughputOverride object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaGlobalSecondaryIndexAutoScalingDescription

Service: Amazon DynamoDB

Represents the auto scaling configuration for a replica global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

IndexStatus

The current state of the replica global secondary index:

- `CREATING` - The index is being created.
- `UPDATING` - The table/index configuration is being updated. The table/index remains available for data operations when `UPDATING`
- `DELETING` - The index is being deleted.
- `ACTIVE` - The index is ready for use.

Type: String

Valid Values: `CREATING | UPDATING | DELETING | ACTIVE`

Required: No

ProvisionedReadCapacityAutoScalingSettings

Represents the auto scaling settings for a global table or global secondary index.

Type: AutoScalingSettingsDescription object

Required: No

ProvisionedWriteCapacityAutoScalingSettings

Represents the auto scaling settings for a global table or global secondary index.

Type: AutoScalingSettingsDescription object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaGlobalSecondaryIndexAutoScalingUpdate

Service: Amazon DynamoDB

Represents the auto scaling settings of a global secondary index for a replica that will be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

ProvisionedReadCapacityAutoScalingUpdate

Represents the auto scaling settings to be modified for a global table or global secondary index.

Type: AutoScalingSettingsUpdate object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaGlobalSecondaryIndexDescription

Service: Amazon DynamoDB

Represents the properties of a replica global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

OnDemandThroughputOverride

Overrides the maximum on-demand throughput for the specified global secondary index in the specified replica table.

Type: OnDemandThroughputOverride object

Required: No

ProvisionedThroughputOverride

If not described, uses the source table GSI's read capacity settings.

Type: ProvisionedThroughputOverride object

Required: No

WarmThroughput

Represents the warm throughput of the global secondary index for this replica.

Type: GlobalSecondaryIndexWarmThroughputDescription object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaGlobalSecondaryIndexSettingsDescription

Service: Amazon DynamoDB

Represents the properties of a global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index. The name must be unique among all other indexes on this table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

IndexStatus

The current status of the global secondary index:

- `CREATING` - The global secondary index is being created.
- `UPDATING` - The global secondary index is being updated.
- `DELETING` - The global secondary index is being deleted.
- `ACTIVE` - The global secondary index is ready for use.

Type: String

Valid Values: `CREATING | UPDATING | DELETING | ACTIVE`

Required: No

ProvisionedReadCapacityAutoScalingSettings

Auto scaling settings for a global secondary index replica's read capacity units.

Type: AutoScalingSettingsDescription object

Required: No

ProvisionedReadCapacityUnits

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a `ThrottlingException`.

Type: Long

Valid Range: Minimum value of 1.

Required: No

ProvisionedWriteCapacityAutoScalingSettings

Auto scaling settings for a global secondary index replica's write capacity units.

Type: AutoScalingSettingsDescription object

Required: No

ProvisionedWriteCapacityUnits

The maximum number of writes consumed per second before DynamoDB returns a `ThrottlingException`.

Type: Long

Valid Range: Minimum value of 1.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaGlobalSecondaryIndexSettingsUpdate

Service: Amazon DynamoDB

Represents the settings of a global secondary index for a global table that will be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index. The name must be unique among all other indexes on this table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

ProvisionedReadCapacityAutoScalingSettingsUpdate

Auto scaling settings for managing a global secondary index replica's read capacity units.

Type: AutoScalingSettingsUpdate object

Required: No

ProvisionedReadCapacityUnits

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a `ThrottlingException`.

Type: Long

Valid Range: Minimum value of 1.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaSettingsDescription

Service: Amazon DynamoDB

Represents the properties of a replica.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region name of the replica.

Type: String

Required: Yes

ReplicaBillingModeSummary

The read/write capacity mode of the replica.

Type: BillingModeSummary object

Required: No

ReplicaGlobalSecondaryIndexSettings

Replica global secondary index settings for the global table.

Type: Array of ReplicaGlobalSecondaryIndexSettingsDescription objects

Required: No

ReplicaProvisionedReadCapacityAutoScalingSettings

Auto scaling settings for a global table replica's read capacity units.

Type: AutoScalingSettingsDescription object

Required: No

ReplicaProvisionedReadCapacityUnits

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a `ThrottlingException`. For more information, see Specifying Read and Write Requirements in the *Amazon DynamoDB Developer Guide*.

Type: Long

Valid Range: Minimum value of 0.

Required: No

ReplicaProvisionedWriteCapacityAutoScalingSettings

Auto scaling settings for a global table replica's write capacity units.

Type: AutoScalingSettingsDescription object

Required: No

ReplicaProvisionedWriteCapacityUnits

The maximum number of writes consumed per second before DynamoDB returns a `ThrottlingException`. For more information, see Specifying Read and Write Requirements in the *Amazon DynamoDB Developer Guide*.

Type: Long

Valid Range: Minimum value of 0.

Required: No

ReplicaStatus

The current state of the Region:

- `CREATING` - The Region is being created.
- `UPDATING` - The Region is being updated.
- `DELETING` - The Region is being deleted.
- `ACTIVE` - The Region is ready for use.

Type: String

Valid Values: `CREATING | CREATION_FAILED | UPDATING | DELETING | ACTIVE | REGION_DISABLED | INACCESSIBLE_ENCRYPTION_CREDENTIALS | ARCHIVING | ARCHIVED | REPLICATION_NOT_AUTHORIZED`

Required: No

ReplicaTableClassSummary

Contains details of the table class.

Type: TableClassSummary object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaSettingsUpdate

Service: Amazon DynamoDB

Represents the settings for a global table in a Region that will be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region of the replica to be added.

Type: String

Required: Yes

ReplicaGlobalSecondaryIndexSettingsUpdate

Represents the settings of a global secondary index for a global table that will be modified.

Type: Array of ReplicaGlobalSecondaryIndexSettingsUpdate objects

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Required: No

ReplicaProvisionedReadCapacityAutoScalingSettingsUpdate

Auto scaling settings for managing a global table replica's read capacity units.

Type: AutoScalingSettingsUpdate object

Required: No

ReplicaProvisionedReadCapacityUnits

The maximum number of strongly consistent reads consumed per second before DynamoDB returns a `ThrottlingException`. For more information, see Specifying Read and Write Requirements in the *Amazon DynamoDB Developer Guide*.

Type: Long

Valid Range: Minimum value of 1.

Required: No

ReplicaTableClass

Replica-specific table class. If not specified, uses the source table's table class.

Type: String

Valid Values: `STANDARD | STANDARD_INFREQUENT_ACCESS`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicationGroupUpdate

Service: Amazon DynamoDB

Represents one of the following:

- A new replica to be added to an existing regional table or global table. This request invokes the `CreateTableReplica` action in the destination Region.
- New parameters for an existing replica. This request invokes the `UpdateTable` action in the destination Region.
- An existing replica to be deleted. The request invokes the `DeleteTableReplica` action in the destination Region, deleting the replica and all if its items in the destination Region.

> **Note**
>
> When you manually remove a table or global table replica, you do not automatically remove any associated scalable targets, scaling policies, or CloudWatch alarms.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Create

The parameters required for creating a replica for the table.

Type: CreateReplicationGroupMemberAction object

Required: No

Delete

The parameters required for deleting a replica for the table.

Type: DeleteReplicationGroupMemberAction object

Required: No

Update

The parameters required for updating a replica for the table.

Type: UpdateReplicationGroupMemberAction object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ReplicaUpdate

Service: Amazon DynamoDB

Represents one of the following:

- A new replica to be added to an existing global table.
- New parameters for an existing replica.
- An existing replica to be removed from an existing global table.

Contents

> **Note**
> In the following list, the required parameters are described first.

Create

The parameters required for creating a replica on an existing global table.

Type: CreateReplicaAction object

Required: No

Delete

The name of the existing replica to be removed.

Type: DeleteReplicaAction object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

RestoreSummary

Service: Amazon DynamoDB

Contains details for the restore.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RestoreDateTime

Point in time or source backup time.

Type: Timestamp

Required: Yes

RestoreInProgress

Indicates if a restore is in progress or not.

Type: Boolean

Required: Yes

SourceBackupArn

The Amazon Resource Name (ARN) of the backup from which the table was restored.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

SourceTableArn

The ARN of the source table of the backup that is being restored.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

S3BucketSource

Service: Amazon DynamoDB

The S3 bucket that is being imported from.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

S3Bucket

The S3 bucket that is being imported from.

Type: String

Length Constraints: Maximum length of 255.

Pattern: `^[a-z0-9A-Z]+[\.\-\w]*[a-z0-9A-Z]+$`

Required: Yes

S3BucketOwner

The account number of the S3 bucket that is being imported from. If the bucket is owned by the requester this is optional.

Type: String

Pattern: `[0-9]{12}`

Required: No

S3KeyPrefix

The key prefix shared by all S3 Objects that are being imported.

Type: String

Length Constraints: Maximum length of 1024.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SourceTableDetails

Service: Amazon DynamoDB

Contains the details of the table when the backup was created.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

KeySchema

Schema of the table.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: Yes

ProvisionedThroughput

Read IOPs and Write IOPS on the table when the backup was created.

Type: ProvisionedThroughput object

Required: Yes

TableCreationDateTime

Time when the source table was created.

Type: Timestamp

Required: Yes

TableId

Unique identifier for the table for which the backup was created.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: Yes

TableName

The name of the table for which the backup was created.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

BillingMode

Controls how you are charged for read and write throughput and how you manage capacity. This setting can be changed later.

- `PROVISIONED` - Sets the read/write capacity mode to `PROVISIONED`. We recommend using `PROVISIONED` for predictable workloads.
- `PAY_PER_REQUEST` - Sets the read/write capacity mode to `PAY_PER_REQUEST`. We recommend using `PAY_PER_REQUEST` for unpredictable workloads.

Type: String

Valid Values: `PROVISIONED | PAY_PER_REQUEST`

Required: No

ItemCount

Number of items in the table. Note that this is an approximate value.

Type: Long

Valid Range: Minimum value of 0.

Required: No

OnDemandThroughput

Sets the maximum number of read and write units for the specified on-demand table. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

TableArn

ARN of the table for which backup was created.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: No

TableSizeBytes

Size of the table in bytes. Note that this is an approximate value.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SourceTableFeatureDetails

Service: Amazon DynamoDB

Contains the details of the features enabled on the table when the backup was created. For example, LSIs, GSIs, streams, TTL.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

GlobalSecondaryIndexes

Represents the GSI properties for the table when the backup was created. It includes the IndexName, KeySchema, Projection, and ProvisionedThroughput for the GSIs on the table at the time of backup.

Type: Array of GlobalSecondaryIndexInfo objects

Required: No

LocalSecondaryIndexes

Represents the LSI properties for the table when the backup was created. It includes the IndexName, KeySchema and Projection for the LSIs on the table at the time of backup.

Type: Array of LocalSecondaryIndexInfo objects

Required: No

SSEDescription

The description of the server-side encryption status on the table when the backup was created.

Type: SSEDescription object

Required: No

StreamDescription

Stream settings on the table when the backup was created.

Type: StreamSpecification object

Required: No

TimeToLiveDescription

Time to Live settings on the table when the backup was created.

Type: TimeToLiveDescription object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SSEDescription

Service: Amazon DynamoDB

The description of the server-side encryption status on the specified table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

InaccessibleEncryptionDateTime

Indicates the time, in UNIX epoch date format, when DynamoDB detected that the table's AWS KMS key was inaccessible. This attribute will automatically be cleared when DynamoDB detects that the table's AWS KMS key is accessible again. DynamoDB will initiate the table archival process when table's AWS KMS key remains inaccessible for more than seven days from this date.

Type: Timestamp

Required: No

KMSMasterKeyArn

The AWS KMS key ARN used for the AWS KMS encryption.

Type: String

Required: No

SSEType

Server-side encryption type. The only supported value is:

- `KMS` - Server-side encryption that uses AWS Key Management Service. The key is stored in your account and is managed by AWS KMS (AWS KMS charges apply).

Type: String

Valid Values: `AES256` | `KMS`

Required: No

Status

Represents the current state of server-side encryption. The only supported values are:

- `ENABLED` - Server-side encryption is enabled.
- `UPDATING` - Server-side encryption is being updated.

Type: String

Valid Values: `ENABLING | ENABLED | DISABLING | DISABLED | UPDATING`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SSESpecification

Service: Amazon DynamoDB

Represents the settings used to enable server-side encryption.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Enabled

Indicates whether server-side encryption is done using an AWS managed key or an AWS owned key. If enabled (true), server-side encryption type is set to `KMS` and an AWS managed key is used (AWS KMS charges apply). If disabled (false) or not specified, server-side encryption is set to AWS owned key.

Type: Boolean

Required: No

KMSMasterKeyId

The AWS KMS key that should be used for the AWS KMS encryption. To specify a key, use its key ID, Amazon Resource Name (ARN), alias name, or alias ARN. Note that you should only provide this parameter if the key is different from the default DynamoDB key `alias/aws/dynamodb`.

Type: String

Required: No

SSEType

Server-side encryption type. The only supported value is:

- `KMS` - Server-side encryption that uses AWS Key Management Service. The key is stored in your account and is managed by AWS KMS (AWS KMS charges apply).

Type: String

Valid Values: `AES256` | `KMS`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamSpecification

Service: Amazon DynamoDB

Represents the DynamoDB Streams configuration for a table in DynamoDB.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

StreamEnabled

Indicates whether DynamoDB Streams is enabled (true) or disabled (false) on the table.

Type: Boolean

Required: Yes

StreamViewType

When an item in the table is modified, `StreamViewType` determines what information is written to the stream for this table. Valid values for `StreamViewType` are:

- `KEYS_ONLY` - Only the key attributes of the modified item are written to the stream.
- `NEW_IMAGE` - The entire item, as it appears after it was modified, is written to the stream.
- `OLD_IMAGE` - The entire item, as it appeared before it was modified, is written to the stream.
- `NEW_AND_OLD_IMAGES` - Both the new and the old item images of the item are written to the stream.

Type: String

Valid Values: `NEW_IMAGE | OLD_IMAGE | NEW_AND_OLD_IMAGES | KEYS_ONLY`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TableAutoScalingDescription

Service: Amazon DynamoDB

Represents the auto scaling configuration for a global table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Replicas

Represents replicas of the global table.

Type: Array of ReplicaAutoScalingDescription objects

Required: No

TableName

The name of the table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

TableStatus

The current state of the table:

- `CREATING` - The table is being created.
- `UPDATING` - The table is being updated.
- `DELETING` - The table is being deleted.
- `ACTIVE` - The table is ready for use.

Type: String

Valid Values: `CREATING | UPDATING | DELETING | ACTIVE | INACCESSIBLE_ENCRYPTION_CREDENTIALS | ARCHIVING | ARCHIVED | REPLICATION_NOT_AUTHORIZED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TableClassSummary

Service: Amazon DynamoDB

Contains details of the table class.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

LastUpdateDateTime

The date and time at which the table class was last updated.

Type: Timestamp

Required: No

TableClass

The table class of the specified table. Valid values are `STANDARD` and `STANDARD_INFREQUENT_ACCESS`.

Type: String

Valid Values: `STANDARD` | `STANDARD_INFREQUENT_ACCESS`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TableCreationParameters

Service: Amazon DynamoDB

The parameters for the table created as part of the import operation.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AttributeDefinitions

The attributes of the table created as part of the import operation.

Type: Array of AttributeDefinition objects

Required: Yes

KeySchema

The primary key and option sort key of the table created as part of the import operation.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: Yes

TableName

The name of the table created as part of the import operation.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

BillingMode

The billing mode for provisioning the table created as part of the import operation.

Type: String

Valid Values: `PROVISIONED | PAY_PER_REQUEST`

Required: No

GlobalSecondaryIndexes

The Global Secondary Indexes (GSI) of the table to be created as part of the import operation.

Type: Array of GlobalSecondaryIndex objects

Required: No

OnDemandThroughput

Sets the maximum number of read and write units for the specified on-demand table. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughput

Represents the provisioned throughput settings for the specified global secondary index. You must use `ProvisionedThroughput` or `OnDemandThroughput` based on your table's capacity mode.

For current minimum and maximum provisioned throughput values, see Service, Account, and Table Quotas in the *Amazon DynamoDB Developer Guide*.

Type: ProvisionedThroughput object

Required: No

SSESpecification

Represents the settings used to enable server-side encryption.

Type: SSESpecification object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TableDescription

Service: Amazon DynamoDB

Represents the properties of a table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ArchivalSummary

Contains information about the table archive.

Type: ArchivalSummary object

Required: No

AttributeDefinitions

An array of `AttributeDefinition` objects. Each of these objects describes one attribute in the table and index key schema.

Each `AttributeDefinition` object in this array is composed of:

- `AttributeName` - The name of the attribute.
- `AttributeType` - The data type for the attribute.

Type: Array of AttributeDefinition objects

Required: No

BillingModeSummary

Contains the details for the read/write capacity mode.

Type: BillingModeSummary object

Required: No

CreationDateTime

The date and time when the table was created, in UNIX epoch time format.

Type: Timestamp

Required: No

DeletionProtectionEnabled

Indicates whether deletion protection is enabled (true) or disabled (false) on the table.

Type: Boolean

Required: No

GlobalSecondaryIndexes

The global secondary indexes, if any, on the table. Each index is scoped to a given partition key value. Each element is composed of:

- `Backfilling` - If true, then the index is currently in the backfilling phase. Backfilling occurs only when a new global secondary index is added to the table. It is the process by which DynamoDB populates the new index with data from the table. (This attribute does not appear for indexes that were created during a `CreateTable` operation.)

  You can delete an index that is being created during the `Backfilling` phase when `IndexStatus` is set to CREATING and `Backfilling` is true. You can't delete the index that is being created when `IndexStatus` is set to CREATING and `Backfilling` is false. (This attribute does not appear for indexes that were created during a `CreateTable` operation.)
- `IndexName` - The name of the global secondary index.
- `IndexSizeBytes` - The total size of the global secondary index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- `IndexStatus` - The current status of the global secondary index:
  - `CREATING` - The index is being created.
  - `UPDATING` - The index is being updated.
  - `DELETING` - The index is being deleted.
  - `ACTIVE` - The index is ready for use.
- `ItemCount` - The number of items in the global secondary index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

- `KeySchema` - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same partition key as the table.
- `Projection` - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
  - `ProjectionType` - One of the following:
    - `KEYS_ONLY` - Only the index and primary keys are projected into the index.
    - `INCLUDE` - In addition to the attributes described in `KEYS_ONLY`, the secondary index will include other non-key attributes that you specify.
    - `ALL` - All of the table attributes are projected into the index.
  - `NonKeyAttributes` - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in `NonKeyAttributes`, summed across all of the secondary indexes, must not exceed 100. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total. This limit only applies when you specify the ProjectionType of `INCLUDE`. You still can specify the ProjectionType of `ALL` to project all attributes from the source table, even if the table has more than 100 attributes.
- `ProvisionedThroughput` - The provisioned throughput settings for the global secondary index, consisting of read and write capacity units, along with data about increases and decreases.

If the table is in the `DELETING` state, no information about indexes will be returned.

Type: Array of GlobalSecondaryIndexDescription objects

Required: No

GlobalTableVersion

Represents the version of global tables in use, if the table is replicated across AWS Regions.

Type: String

Required: No

GlobalTableWitnesses

The witness Region and its current status in the MRSC global table. Only one witness Region can be configured per MRSC global table.

Type: Array of GlobalTableWitnessDescription objects

Required: No

ItemCount

The number of items in the specified table. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

Type: Long

Required: No

KeySchema

The primary key structure for the table. Each `KeySchemaElement` consists of:

- `AttributeName` - The name of the attribute.
- `KeyType` - The role of the attribute:
  - `HASH` - partition key
  - `RANGE` - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.
> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

For more information about primary keys, see Primary Key in the *Amazon DynamoDB Developer Guide*.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: No

LatestStreamArn

The Amazon Resource Name (ARN) that uniquely identifies the latest stream for this table.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

LatestStreamLabel

A timestamp, in ISO 8601 format, for this stream.

Note that `LatestStreamLabel` is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- AWS customer ID
- Table name
- `StreamLabel`

Type: String

Required: No

LocalSecondaryIndexes

Represents one or more local secondary indexes on the table. Each index is scoped to a given partition key value. Tables with one or more local secondary indexes are subject to an item collection size limit, where the amount of data within a given item collection cannot exceed 10 GB. Each element is composed of:

- `IndexName` - The name of the local secondary index.
- `KeySchema` - Specifies the complete index key schema. The attribute names in the key schema must be between 1 and 255 characters (inclusive). The key schema must begin with the same partition key as the table.
- `Projection` - Specifies attributes that are copied (projected) from the table into the index. These are in addition to the primary key attributes and index key attributes, which are automatically projected. Each attribute specification is composed of:
  - `ProjectionType` - One of the following:
    - `KEYS_ONLY` - Only the index and primary keys are projected into the index.
    - `INCLUDE` - Only the specified table attributes are projected into the index. The list of projected attributes is in `NonKeyAttributes`.
    - `ALL` - All of the table attributes are projected into the index.

- `NonKeyAttributes` - A list of one or more non-key attribute names that are projected into the secondary index. The total count of attributes provided in `NonKeyAttributes`, summed across all of the secondary indexes, must not exceed 100. If you project the same attribute into two different indexes, this counts as two distinct attributes when determining the total. This limit only applies when you specify the ProjectionType of `INCLUDE`. You still can specify the ProjectionType of `ALL` to project all attributes from the source table, even if the table has more than 100 attributes.
- `IndexSizeBytes` - Represents the total size of the index, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.
- `ItemCount` - Represents the number of items in the index. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

If the table is in the `DELETING` state, no information about indexes will be returned.

Type: Array of LocalSecondaryIndexDescription objects

Required: No

MultiRegionConsistency

Indicates one of the following consistency modes for a global table:

- `EVENTUAL`: Indicates that the global table is configured for multi-Region eventual consistency (MREC).
- `STRONG`: Indicates that the global table is configured for multi-Region strong consistency (MRSC).

If you don't specify this field, the global table consistency mode defaults to `EVENTUAL`. For more information about global tables consistency modes, see Consistency modes in DynamoDB developer guide.

Type: String

Valid Values: `EVENTUAL | STRONG`

Required: No

OnDemandThroughput

The maximum number of read and write units for the specified on-demand table. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughput

The provisioned throughput settings for the table, consisting of read and write capacity units, along with data about increases and decreases.

Type: ProvisionedThroughputDescription object

Required: No

Replicas

Represents replicas of the table.

Type: Array of ReplicaDescription objects

Required: No

RestoreSummary

Contains details for the restore.

Type: RestoreSummary object

Required: No

SSEDescription

The description of the server-side encryption status on the specified table.

Type: SSEDescription object

Required: No

StreamSpecification

The current DynamoDB Streams configuration for the table.

Type: StreamSpecification object

Required: No

TableArn

The Amazon Resource Name (ARN) that uniquely identifies the table.

Type: String

Required: No

TableClassSummary

Contains details of the table class.

Type: TableClassSummary object

Required: No

TableId

Unique identifier for the table for which the backup was created.

Type: String

Pattern: `[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}`

Required: No

TableName

The name of the table.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

TableSizeBytes

The total size of the specified table, in bytes. DynamoDB updates this value approximately every six hours. Recent changes might not be reflected in this value.

Type: Long

Required: No

TableStatus

The current state of the table:

- `CREATING` - The table is being created.
- `UPDATING` - The table/index configuration is being updated. The table/index remains available for data operations when `UPDATING`.
- `DELETING` - The table is being deleted.
- `ACTIVE` - The table is ready for use.
- `INACCESSIBLE_ENCRYPTION_CREDENTIALS` - The AWS KMS key used to encrypt the table in inaccessible. Table operations may fail due to failure to use the AWS KMS key. DynamoDB will initiate the table archival process when a table's AWS KMS key remains inaccessible for more than seven days.
- `ARCHIVING` - The table is being archived. Operations are not allowed until archival is complete.
- `ARCHIVED` - The table has been archived. See the ArchivalReason for more information.

Type: String

Valid Values: `CREATING | UPDATING | DELETING | ACTIVE | INACCESSIBLE_ENCRYPTION_CREDENTIALS | ARCHIVING | ARCHIVED | REPLICATION_NOT_AUTHORIZED`

Required: No

WarmThroughput

Describes the warm throughput value of the base table.

Type: TableWarmThroughputDescription object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TableWarmThroughputDescription

Service: Amazon DynamoDB

Represents the warm throughput value (in read units per second and write units per second) of the table. Warm throughput is applicable for DynamoDB Standard-IA tables and specifies the minimum provisioned capacity maintained for immediate data access.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ReadUnitsPerSecond

Represents the base table's warm throughput value in read units per second.

Type: Long

Valid Range: Minimum value of 1.

Required: No

Status

Represents warm throughput value of the base table.

Type: String

Valid Values: `CREATING | UPDATING | DELETING | ACTIVE | INACCESSIBLE_ENCRYPTION_CREDENTIALS | ARCHIVING | ARCHIVED | REPLICATION_NOT_AUTHORIZED`

Required: No

WriteUnitsPerSecond

Represents the base table's warm throughput value in write units per second.

Type: Long

Valid Range: Minimum value of 1.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Tag

Service: Amazon DynamoDB

Describes a tag. A tag is a key-value pair. You can add up to 50 tags to a single DynamoDB table.

AWS-assigned tag names and values are automatically assigned the `aws:` prefix, which the user cannot assign. AWS-assigned tag names do not count towards the tag limit of 50. User-assigned tag names have the prefix `user:` in the Cost Allocation Report. You cannot backdate the application of a tag.

For an overview on tagging DynamoDB resources, see Tagging for DynamoDB in the *Amazon DynamoDB Developer Guide*.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Key

The key of the tag. Tag keys are case sensitive. Each DynamoDB table can only have up to one tag with the same key. If you try to add an existing tag (same key), the existing tag value will be updated to the new value.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 128.

Required: Yes

Value

The value of the tag. Tag values are case-sensitive and can be null.

Type: String

Length Constraints: Minimum length of 0. Maximum length of 256.

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ThrottlingReason

Service: Amazon DynamoDB

Represents the specific reason why a DynamoDB request was throttled and the ARN of the impacted resource. This helps identify exactly what resource is being throttled, what type of operation caused it, and why the throttling occurred.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

reason

The reason for throttling. The throttling reason follows a specific format: `ResourceType+OperationType+LimitType`:

- Resource Type (What is being throttled): Table or Index
- Operation Type (What kind of operation): Read or Write
- Limit Type (Why the throttling occurred):
  - `ProvisionedThroughputExceeded`: The request rate is exceeding the provisioned throughput capacity (read or write capacity units) configured for a table or a global secondary index (GSI) in provisioned capacity mode.
  - `AccountLimitExceeded`: The request rate has caused a table or global secondary index (GSI) in on-demand mode to exceed the per-table account-level service quotas for read/write throughput in the current AWS Region.
  - `KeyRangeThroughputExceeded`: The request rate directed at a specific partition key value has exceeded the internal partition-level throughput limits, indicating uneven access patterns across the table's or GSI's key space.
  - `MaxOnDemandThroughputExceeded`: The request rate has exceeded the configured maximum throughput limits set for a table or index in on-demand capacity mode.

Examples of complete throttling reasons:

- TableReadProvisionedThroughputExceeded
- IndexWriteAccountLimitExceeded

This helps identify exactly what resource is being throttled, what type of operation caused it, and why the throttling occurred.

Type: String

Required: No

resource

The Amazon Resource Name (ARN) of the DynamoDB table or index that experienced the throttling event.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TimeToLiveDescription

Service: Amazon DynamoDB

The description of the Time to Live (TTL) status on the specified table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AttributeName

The name of the TTL attribute for items in the table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: No

TimeToLiveStatus

The TTL status for the table.

Type: String

Valid Values: `ENABLING | DISABLING | ENABLED | DISABLED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TimeToLiveSpecification

Service: Amazon DynamoDB

Represents the settings used to enable or disable Time to Live (TTL) for the specified table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AttributeName

The name of the TTL attribute used to store the expiration time for items in the table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

Enabled

Indicates whether TTL is to be enabled (true) or disabled (false) on the table.

Type: Boolean

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TransactGetItem

Service: Amazon DynamoDB

Specifies an item to be retrieved as part of the transaction.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Get

Contains the primary key that identifies the item to get, together with the name of the table that contains the item, and optionally the specific attributes of the item to retrieve.

Type: Get object

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

TransactWriteItem

Service: Amazon DynamoDB

A list of requests that can perform update, put, delete, or check operations on multiple items in one or more tables atomically.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ConditionCheck

A request to perform a check item operation.

Type: ConditionCheck object

Required: No

Delete

A request to perform a `DeleteItem` operation.

Type: Delete object

Required: No

Put

A request to perform a `PutItem` operation.

Type: Put object

Required: No

Update

A request to perform an `UpdateItem` operation.

Type: Update object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Update

Service: Amazon DynamoDB

Represents a request to perform an `UpdateItem` operation.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Key

The primary key of the item to be updated. Each element consists of an attribute name and a value for that attribute.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: Yes

TableName

Name of the table for the `UpdateItem` request. You can also provide the Amazon Resource Name (ARN) of the table in this parameter.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 1024.

Required: Yes

UpdateExpression

An expression that defines one or more attributes to be updated, the action to be performed on them, and new value(s) for them.

Type: String

Required: Yes

ConditionExpression

A condition that must be satisfied in order for a conditional update to succeed.

Type: String

Required: No

ExpressionAttributeNames

One or more substitution tokens for attribute names in an expression.

Type: String to string map

Value Length Constraints: Maximum length of 65535.

Required: No

ExpressionAttributeValues

One or more values that can be substituted in an expression.

Type: String to AttributeValue object map

Required: No

ReturnValuesOnConditionCheckFailure

Use `ReturnValuesOnConditionCheckFailure` to get the item attributes if the `Update` condition fails. For `ReturnValuesOnConditionCheckFailure`, the valid values are: NONE and ALL_OLD.

Type: String

Valid Values: `ALL_OLD | NONE`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

UpdateGlobalSecondaryIndexAction

Service: Amazon DynamoDB

Represents the new provisioned throughput settings to be applied to a global secondary index.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

IndexName

The name of the global secondary index to be updated.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: Yes

OnDemandThroughput

Updates the maximum number of read and write units for the specified global secondary index. If you use this parameter, you must specify `MaxReadRequestUnits`, `MaxWriteRequestUnits`, or both.

Type: OnDemandThroughput object

Required: No

ProvisionedThroughput

Represents the provisioned throughput settings for the specified global secondary index.

For current minimum and maximum provisioned throughput values, see Service, Account, and Table Quotas in the *Amazon DynamoDB Developer Guide*.

Type: ProvisionedThroughput object

Required: No

WarmThroughput

Represents the warm throughput value of the new provisioned throughput settings to be applied to a global secondary index.

Type: WarmThroughput object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UpdateKinesisStreamingConfiguration

Service: Amazon DynamoDB

Enables updating the configuration for Kinesis Streaming.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ApproximateCreationDateTimePrecision

Enables updating the precision of Kinesis data stream timestamp.

Type: String

Valid Values: `MILLISECOND | MICROSECOND`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

UpdateReplicationGroupMemberAction

Service: Amazon DynamoDB

Represents a replica to be modified.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

RegionName

The Region where the replica exists.

Type: String

Required: Yes

GlobalSecondaryIndexes

Replica-specific global secondary index settings.

Type: Array of ReplicaGlobalSecondaryIndex objects

Array Members: Minimum number of 1 item.

Required: No

KMSMasterKeyId

The AWS KMS key of the replica that should be used for AWS KMS encryption. To specify a key, use its key ID, Amazon Resource Name (ARN), alias name, or alias ARN. Note that you should only provide this parameter if the key is different from the default DynamoDB KMS key `alias/aws/dynamodb`.

Type: String

Required: No

OnDemandThroughputOverride

Overrides the maximum on-demand throughput for the replica table.

Type: OnDemandThroughputOverride object

Required: No

ProvisionedThroughputOverride

Replica-specific provisioned throughput. If not specified, uses the source table's provisioned throughput settings.

Type: ProvisionedThroughputOverride object

Required: No

TableClassOverride

Replica-specific table class. If not specified, uses the source table's table class.

Type: String

Valid Values: `STANDARD | STANDARD_INFREQUENT_ACCESS`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

WarmThroughput

Service: Amazon DynamoDB

Provides visibility into the number of read and write operations your table or secondary index can instantaneously support. The settings can be modified using the `UpdateTable` operation to meet the throughput requirements of an upcoming peak event.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ReadUnitsPerSecond

Represents the number of read operations your base table can instantaneously support.

Type: Long

Required: No

WriteUnitsPerSecond

Represents the number of write operations your base table can instantaneously support.

Type: Long

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

WriteRequest

Service: Amazon DynamoDB

Represents an operation to perform - either `DeleteItem` or `PutItem`. You can only request one of these operations, not both, in a single `WriteRequest`. If you do need to perform both of these operations, you need to provide two separate `WriteRequest` objects.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

DeleteRequest

A request to perform a `DeleteItem` operation.

Type: DeleteRequest object

Required: No

PutRequest

A request to perform a `PutItem` operation.

Type: PutRequest object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DynamoDB Accelerator

The following data types are supported by DynamoDB Accelerator:

- Cluster
- Endpoint
- Event
- Node
- NodeTypeSpecificValue
- NotificationConfiguration
- Parameter
- ParameterGroup
- ParameterGroupStatus
- ParameterNameValue
- SecurityGroupMembership
- SSEDescription
- SSESpecification
- Subnet
- SubnetGroup
- Tag

Cluster

Service: DynamoDB Accelerator

Contains all of the attributes of a specific DAX cluster.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ActiveNodes

The number of nodes in the cluster that are active (i.e., capable of serving requests).

Type: Integer

Required: No

ClusterArn

The Amazon Resource Name (ARN) that uniquely identifies the cluster.

Type: String

Required: No

ClusterDiscoveryEndpoint

The endpoint for this DAX cluster, consisting of a DNS name, a port number, and a URL. Applications should use the URL to configure the DAX client to find their cluster.

Type: Endpoint object

Required: No

ClusterEndpointEncryptionType

The type of encryption supported by the cluster's endpoint. Values are:

- `NONE` for no encryption

  `TLS` for Transport Layer Security

Type: String

Valid Values: `NONE | TLS`

Required: No

ClusterName

The name of the DAX cluster.

Type: String

Required: No

Description

The description of the cluster.

Type: String

Required: No

IamRoleArn

A valid Amazon Resource Name (ARN) that identifies an IAM role. At runtime, DAX will assume this role and use the role's permissions to access DynamoDB on your behalf.

Type: String

Required: No

NodeIdsToRemove

A list of nodes to be removed from the cluster.

Type: Array of strings

Required: No

Nodes

A list of nodes that are currently in the cluster.

Type: Array of Node objects

Required: No

NodeType

The node type for the nodes in the cluster. (All nodes in a DAX cluster are of the same type.)

Type: String

Required: No

NotificationConfiguration

Describes a notification topic and its status. Notification topics are used for publishing DAX events to subscribers using Amazon Simple Notification Service (SNS).

Type: NotificationConfiguration object

Required: No

ParameterGroup

The parameter group being used by nodes in the cluster.

Type: ParameterGroupStatus object

Required: No

PreferredMaintenanceWindow

A range of time when maintenance of DAX cluster software will be performed. For example: `sun:01:00-sun:09:00`. Cluster maintenance normally takes less than 30 minutes, and is performed automatically within the maintenance window.

Type: String

Required: No

SecurityGroups

A list of security groups, and the status of each, for the nodes in the cluster.

Type: Array of SecurityGroupMembership objects

Required: No

SSEDescription

The description of the server-side encryption status on the specified DAX cluster.

Type: SSEDescription object

Required: No

Status

The current status of the cluster.

Type: String

Required: No

SubnetGroup

The subnet group where the DAX cluster is running.

Type: String

Required: No

TotalNodes

The total number of nodes in the cluster.

Type: Integer

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Endpoint

Service: DynamoDB Accelerator

Represents the information required for client programs to connect to the endpoint for a DAX cluster.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Address

The DNS hostname of the endpoint.

Type: String

Required: No

Port

The port number that applications should use to connect to the endpoint.

Type: Integer

Required: No

URL

The URL that applications should use to connect to the endpoint. The default ports are 8111 for the "dax" protocol and 9111 for the "daxs" protocol.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Event

Service: DynamoDB Accelerator

Represents a single occurrence of something interesting within the system. Some examples of events are creating a DAX cluster, adding or removing a node, or rebooting a node.

Contents

> **Note**
> In the following list, the required parameters are described first.

Date

The date and time when the event occurred.

Type: Timestamp

Required: No

Message

A user-defined message associated with the event.

Type: String

Required: No

SourceName

The source of the event. For example, if the event occurred at the node level, the source would be the node ID.

Type: String

Required: No

SourceType

Specifies the origin of this event - a cluster, a parameter group, a node ID, etc.

Type: String

Valid Values: `CLUSTER` | `PARAMETER_GROUP` | `SUBNET_GROUP`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Node

Service: DynamoDB Accelerator

Represents an individual node within a DAX cluster.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AvailabilityZone

The Availability Zone (AZ) in which the node has been deployed.

Type: String

Required: No

Endpoint

The endpoint for the node, consisting of a DNS name and a port number. Client applications can connect directly to a node endpoint, if desired (as an alternative to allowing DAX client software to intelligently route requests and responses to nodes in the DAX cluster.

Type: Endpoint object

Required: No

NodeCreateTime

The date and time (in UNIX epoch format) when the node was launched.

Type: Timestamp

Required: No

NodeId

A system-generated identifier for the node.

Type: String

Required: No

NodeStatus

The current status of the node. For example: `available`.

Type: String

Required: No

ParameterGroupStatus

The status of the parameter group associated with this node. For example, `in-sync`.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NodeTypeSpecificValue

Service: DynamoDB Accelerator

Represents a parameter value that is applicable to a particular node type.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

NodeType

A node type to which the parameter value applies.

Type: String

Required: No

Value

The parameter value for this node type.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

NotificationConfiguration

Service: DynamoDB Accelerator

Describes a notification topic and its status. Notification topics are used for publishing DAX events to subscribers using Amazon Simple Notification Service (SNS).

Contents

> **Note**
>
> In the following list, the required parameters are described first.

TopicArn

The Amazon Resource Name (ARN) that identifies the topic.

Type: String

Required: No

TopicStatus

The current state of the topic. A value of “active” means that notifications will be sent to the topic. A value of “inactive” means that notifications will not be sent to the topic.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Parameter

Service: DynamoDB Accelerator

Describes an individual setting that controls some aspect of DAX behavior.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AllowedValues

A range of values within which the parameter can be set.

Type: String

Required: No

ChangeType

The conditions under which changes to this parameter can be applied. For example, `requires-reboot` indicates that a new value for this parameter will only take effect if a node is rebooted.

Type: String

Valid Values: `IMMEDIATE | REQUIRES_REBOOT`

Required: No

DataType

The data type of the parameter. For example, `integer`:

Type: String

Required: No

Description

A description of the parameter

Type: String

Required: No

IsModifiable

Whether the customer is allowed to modify the parameter.

Type: String

Valid Values: `TRUE | FALSE | CONDITIONAL`

Required: No

NodeTypeSpecificValues

A list of node types, and specific parameter values for each node.

Type: Array of NodeTypeSpecificValue objects

Required: No

ParameterName

The name of the parameter.

Type: String

Required: No

ParameterType

Determines whether the parameter can be applied to any nodes, or only nodes of a particular type.

Type: String

Valid Values: `DEFAULT | NODE_TYPE_SPECIFIC`

Required: No

ParameterValue

The value for the parameter.

Type: String

Required: No

Source

How the parameter is defined. For example, `system` denotes a system-defined parameter.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ParameterGroup

Service: DynamoDB Accelerator

A named set of parameters that are applied to all of the nodes in a DAX cluster.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Description

A description of the parameter group.

Type: String

Required: No

ParameterGroupName

The name of the parameter group.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ParameterGroupStatus

Service: DynamoDB Accelerator

The status of a parameter group.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

NodeIdsToReboot

The node IDs of one or more nodes to be rebooted.

Type: Array of strings

Required: No

ParameterApplyStatus

The status of parameter updates.

Type: String

Required: No

ParameterGroupName

The name of the parameter group.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

ParameterNameValue

Service: DynamoDB Accelerator

An individual DAX parameter.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ParameterName

The name of the parameter.

Type: String

Required: No

ParameterValue

The value of the parameter.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SecurityGroupMembership

Service: DynamoDB Accelerator

An individual VPC security group and its status.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

SecurityGroupIdentifier

The unique ID for this security group.

Type: String

Required: No

Status

The status of this security group.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SSEDescription

Service: DynamoDB Accelerator

The description of the server-side encryption status on the specified DAX cluster.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Status

The current state of server-side encryption:

- `ENABLING` - Server-side encryption is being enabled.
- `ENABLED` - Server-side encryption is enabled.
- `DISABLING` - Server-side encryption is being disabled.
- `DISABLED` - Server-side encryption is disabled.

Type: String

Valid Values: `ENABLING | ENABLED | DISABLING | DISABLED`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SSESpecification

Service: DynamoDB Accelerator

Represents the settings used to enable server-side encryption.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Enabled

Indicates whether server-side encryption is enabled (true) or disabled (false) on the cluster.

Type: Boolean

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Subnet

Service: DynamoDB Accelerator

Represents the subnet associated with a DAX cluster. This parameter refers to subnets defined in Amazon Virtual Private Cloud (Amazon VPC) and used with DAX.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

SubnetAvailabilityZone

The Availability Zone (AZ) for the subnet.

Type: String

Required: No

SubnetIdentifier

The system-assigned identifier for the subnet.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SubnetGroup

Service: DynamoDB Accelerator

Represents the output of one of the following actions:

- *CreateSubnetGroup*
- *ModifySubnetGroup*

Contents

> **Note**
> In the following list, the required parameters are described first.

Description

The description of the subnet group.

Type: String

Required: No

SubnetGroupName

The name of the subnet group.

Type: String

Required: No

Subnets

A list of subnets associated with the subnet group.

Type: Array of Subnet objects

Required: No

VpcId

The Amazon Virtual Private Cloud identifier (VPC ID) of the subnet group.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Tag

Service: DynamoDB Accelerator

A description of a tag. Every tag is a key-value pair. You can add up to 50 tags to a single DAX cluster.

AWS-assigned tag names and values are automatically assigned the `aws:` prefix, which the user cannot assign. AWS-assigned tag names do not count towards the tag limit of 50. User-assigned tag names have the prefix `user:`.

You cannot backdate the application of a tag.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

Key

The key for the tag. Tag keys are case sensitive. Every DAX cluster can only have one tag with the same key. If you try to add an existing tag (same key), the existing tag value will be updated to the new value.

Type: String

Required: No

Value

The value of the tag. Tag values are case-sensitive and can be null.

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Amazon DynamoDB Streams

The following data types are supported by Amazon DynamoDB Streams:

- AttributeValue
- Identity
- KeySchemaElement
- Record
- SequenceNumberRange
- Shard
- ShardFilter
- Stream
- StreamDescription
- StreamRecord

AttributeValue

Service: Amazon DynamoDB Streams

Represents the data for an attribute.

Each attribute value is described as a name-value pair. The name is the data type, and the value is the data itself.

For more information, see Data Types in the *Amazon DynamoDB Developer Guide*.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

B

An attribute of type Binary. For example:

```
"B": "dGhpcyB0ZXh0IGlzIGJhc2U2NC1lbmNvZGVk"
```

Type: Base64-encoded binary data object

Required: No

BOOL

An attribute of type Boolean. For example:

```
"BOOL": true
```

Type: Boolean

Required: No

BS

An attribute of type Binary Set. For example:

```
"BS": ["U3Vubnk=", "UmFpbnk=", "U25vd3k="]
```

Type: Array of Base64-encoded binary data objects

Required: No

L

An attribute of type List. For example:

```
"L": [ {"S": "Cookies"} , {"S": "Coffee"}, {"N": "3.14159"}]
```

Type: Array of AttributeValue objects

Required: No

M

An attribute of type Map. For example:

```
"M": {"Name": {"S": "Joe"}, "Age": {"N": "35"}}
```

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

N

An attribute of type Number. For example:

```
"N": "123.45"
```

Numbers are sent across the network to DynamoDB as strings, to maximize compatibility across languages and libraries. However, DynamoDB treats them as number type attributes for mathematical operations.

Type: String

Required: No

NS

An attribute of type Number Set. For example:

```
"NS": ["42.2", "-19", "7.5", "3.14"]
```

Numbers are sent across the network to DynamoDB as strings, to maximize compatibility across languages and libraries. However, DynamoDB treats them as number type attributes for mathematical operations.

Type: Array of strings

Required: No

NULL

An attribute of type Null. For example:

```
"NULL": true
```

Type: Boolean

Required: No

S

An attribute of type String. For example:

```
"S": "Hello"
```

Type: String

Required: No

SS

An attribute of type String Set. For example:

```
"SS": ["Giraffe", "Hippo" ,"Zebra"]
```

Type: Array of strings

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

Identity

Service: Amazon DynamoDB Streams

Contains details about the type of identity that made the request.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

PrincipalId

A unique identifier for the entity that made the call. For Time To Live, the principalId is "dynamodb.amazonaws.com".

Type: String

Required: No

Type

The type of the identity. For Time To Live, the type is "Service".

Type: String

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

KeySchemaElement

Service: Amazon DynamoDB Streams

Represents *a single element* of a key schema. A key schema specifies the attributes that make up the primary key of a table, or the key attributes of an index.

A `KeySchemaElement` represents exactly one attribute of the primary key. For example, a simple primary key would be represented by one `KeySchemaElement` (for the partition key). A composite primary key would require one `KeySchemaElement` for the partition key, and another `KeySchemaElement` for the sort key.

A `KeySchemaElement` must be a scalar, top-level attribute (not a nested attribute). The data type must be one of String, Number, or Binary. The attribute cannot be nested within a List or a Map.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

AttributeName

The name of a key attribute.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 255.

Required: Yes

KeyType

The role that this key attribute will assume:

- `HASH` - partition key
- `RANGE` - sort key

> **Note**
>
> The partition key of an item is also known as its *hash attribute*. The term "hash attribute" derives from DynamoDB's usage of an internal hash function to evenly distribute data items across partitions, based on their partition key values.

> The sort key of an item is also known as its *range attribute*. The term "range attribute" derives from the way DynamoDB stores items with the same partition key physically close together, in sorted order by the sort key value.

Type: String

Valid Values: `HASH` | `RANGE`

Required: Yes

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Record

Service: Amazon DynamoDB Streams

A description of a unique event within a stream.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

awsRegion

The region in which the `GetRecords` request was received.

Type: String

Required: No

dynamodb

The main body of the stream record, containing all of the DynamoDB-specific fields.

Type: StreamRecord object

Required: No

eventID

A globally unique identifier for the event that was recorded in this stream record.

Type: String

Required: No

eventName

The type of data modification that was performed on the DynamoDB table:

- `INSERT` - a new item was added to the table.
- `MODIFY` - one or more of an existing item's attributes were modified.
- `REMOVE` - the item was deleted from the table

Type: String

Valid Values: `INSERT | MODIFY | REMOVE`

Required: No

eventSource

The AWS service from which the stream record originated. For DynamoDB Streams, this is `aws:dynamodb`.

Type: String

Required: No

eventVersion

The version number of the stream record format. This number is updated whenever the structure of `Record` is modified.

Client applications must not assume that `eventVersion` will remain at a particular value, as this number is subject to change at any time. In general, `eventVersion` will only increase as the low-level DynamoDB Streams API evolves.

Type: String

Required: No

userIdentity

Items that are deleted by the Time to Live process after expiration have the following fields:

- Records[].userIdentity.type

  "Service"
- Records[].userIdentity.principalId

  "dynamodb.amazonaws.com"

Type: Identity object

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

SequenceNumberRange

Service: Amazon DynamoDB Streams

The beginning and ending sequence numbers for the stream records contained within a shard.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

EndingSequenceNumber

The last sequence number for the stream records contained within a shard. String contains numeric characters only.

Type: String

Length Constraints: Minimum length of 21. Maximum length of 40.

Required: No

StartingSequenceNumber

The first sequence number for the stream records contained within a shard. String contains numeric characters only.

Type: String

Length Constraints: Minimum length of 21. Maximum length of 40.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

Shard

Service: Amazon DynamoDB Streams

A uniquely identified group of stream records within a stream.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ParentShardId

The shard ID of the current shard's parent.

Type: String

Length Constraints: Minimum length of 28. Maximum length of 65.

Required: No

SequenceNumberRange

The range of possible sequence numbers for the shard.

Type: SequenceNumberRange object

Required: No

ShardId

The system-generated identifier for this shard.

Type: String

Length Constraints: Minimum length of 28. Maximum length of 65.

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ShardFilter

Service: Amazon DynamoDB Streams

This optional field contains the filter definition for the `DescribeStream` API.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ShardId

Contains the `shardId` of the parent shard for which you are requesting child shards.

Sample request:

Type: String

Length Constraints: Minimum length of 28. Maximum length of 65.

Required: No

Type

Contains the type of filter to be applied on the `DescribeStream` API. Currently, the only value this parameter accepts is `CHILD_SHARDS`.

Type: String

Valid Values: `CHILD_SHARDS`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++

- AWS SDK for Java V2
- AWS SDK for Ruby V3

Stream

Service: Amazon DynamoDB Streams

Represents all of the data describing a particular stream.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

StreamArn

The Amazon Resource Name (ARN) for the stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

StreamLabel

A timestamp, in ISO 8601 format, for this stream.

Note that `LatestStreamLabel` is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name
- the `StreamLabel`

Type: String

Required: No

TableName

The DynamoDB table with which the stream is associated.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

StreamDescription

Service: Amazon DynamoDB Streams

Represents all of the data describing a particular stream.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

CreationRequestDateTime

The date and time when the request to create this stream was issued.

Type: Timestamp

Required: No

KeySchema

The key attribute(s) of the stream's DynamoDB table.

Type: Array of KeySchemaElement objects

Array Members: Minimum number of 1 item. Maximum number of 2 items.

Required: No

LastEvaluatedShardId

The shard ID of the item where the operation stopped, inclusive of the previous result set. Use this value to start a new operation, excluding this value in the new request.

If `LastEvaluatedShardId` is empty, then the "last page" of results has been processed and there is currently no more data to be retrieved.

If `LastEvaluatedShardId` is not empty, it does not necessarily mean that there is more data in the result set. The only way to know when you have reached the end of the result set is when `LastEvaluatedShardId` is empty.

Type: String

Length Constraints: Minimum length of 28. Maximum length of 65.

Required: No

Shards

The shards that comprise the stream.

Type: Array of Shard objects

Required: No

StreamArn

The Amazon Resource Name (ARN) for the stream.

Type: String

Length Constraints: Minimum length of 37. Maximum length of 1024.

Required: No

StreamLabel

A timestamp, in ISO 8601 format, for this stream.

Note that `LatestStreamLabel` is not a unique identifier for the stream, because it is possible that a stream from another table might have the same timestamp. However, the combination of the following three elements is guaranteed to be unique:

- the AWS customer ID.
- the table name
- the `StreamLabel`

Type: String

Required: No

StreamStatus

Indicates the current status of the stream:

- `ENABLING` - Streams is currently being enabled on the DynamoDB table.
- `ENABLED` - the stream is enabled.
- `DISABLING` - Streams is currently being disabled on the DynamoDB table.

- `DISABLED` - the stream is disabled.

Type: String

Valid Values: `ENABLING | ENABLED | DISABLING | DISABLED`

Required: No

StreamViewType

Indicates the format of the records within this stream:

- `KEYS_ONLY` - only the key attributes of items that were modified in the DynamoDB table.
- `NEW_IMAGE` - entire items from the table, as they appeared after they were modified.
- `OLD_IMAGE` - entire items from the table, as they appeared before they were modified.
- `NEW_AND_OLD_IMAGES` - both the new and the old images of the items from the table.

Type: String

Valid Values: `NEW_IMAGE | OLD_IMAGE | NEW_AND_OLD_IMAGES | KEYS_ONLY`

Required: No

TableName

The DynamoDB table with which the stream is associated.

Type: String

Length Constraints: Minimum length of 3. Maximum length of 255.

Pattern: `[a-zA-Z0-9_.-]+`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2

- AWS SDK for Ruby V3

StreamRecord

Service: Amazon DynamoDB Streams

A description of a single data modification that was performed on an item in a DynamoDB table.

Contents

> **Note**
>
> In the following list, the required parameters are described first.

ApproximateCreationDateTime

The approximate date and time when the stream record was created, in ISO 8601 format and rounded down to the closest second.

Type: Timestamp

Required: No

Keys

The primary key attribute(s) for the DynamoDB item that was modified.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

NewImage

The item in the DynamoDB table as it appeared after it was modified.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

OldImage

The item in the DynamoDB table as it appeared before it was modified.

Type: String to AttributeValue object map

Key Length Constraints: Maximum length of 65535.

Required: No

SequenceNumber

The sequence number of the stream record.

Type: String

Length Constraints: Minimum length of 21. Maximum length of 40.

Required: No

SizeBytes

The size of the stream record, in bytes.

Type: Long

Valid Range: Minimum value of 1.

Required: No

StreamViewType

The type of data from the modified DynamoDB item that was captured in this stream record:

- `KEYS_ONLY` - only the key attributes of the modified item.
- `NEW_IMAGE` - the entire item, as it appeared after it was modified.
- `OLD_IMAGE` - the entire item, as it appeared before it was modified.
- `NEW_AND_OLD_IMAGES` - both the new and the old item images of the item.

Type: String

Valid Values: `NEW_IMAGE | OLD_IMAGE | NEW_AND_OLD_IMAGES | KEYS_ONLY`

Required: No

See Also

For more information about using this API in one of the language-specific AWS SDKs, see the following:

- AWS SDK for C++
- AWS SDK for Java V2
- AWS SDK for Ruby V3

Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

IncompleteSignature

The request signature does not conform to AWS standards.

HTTP Status Code: 400

InternalFailure

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

InvalidAction

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

InvalidClientTokenId

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

NotAuthorized

You do not have permission to perform this action.

HTTP Status Code: 400

OptInRequired

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

RequestExpired

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

ServiceUnavailable

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

ThrottlingException

The request was denied due to request throttling.

HTTP Status Code: 400

ValidationError

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400