
Amazon Keyspaces (for Apache Cassandra)

API Reference

API Version 2022-02-10



Amazon Keyspaces (for Apache Cassandra): API Reference

Copyright © 2022 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
CreateKeyspace	3
Request Syntax	3
Request Parameters	3
Response Syntax	3
Response Elements	4
Errors	4
See Also	4
CreateTable	6
Request Syntax	6
Request Parameters	7
Response Syntax	9
Response Elements	9
Errors	10
See Also	10
DeleteKeyspace	11
Request Syntax	11
Request Parameters	11
Response Elements	11
Errors	11
See Also	12
DeleteTable	13
Request Syntax	13
Request Parameters	13
Response Elements	13
Errors	13
See Also	14
GetKeyspace	15
Request Syntax	15
Request Parameters	15
Response Syntax	15
Response Elements	15
Errors	16
See Also	16
GetTable	17
Request Syntax	17
Request Parameters	17
Response Syntax	17
Response Elements	18
Errors	20
See Also	20
ListKeyspaces	21
Request Syntax	21
Request Parameters	21
Response Syntax	21
Response Elements	21
Errors	22
See Also	22
ListTables	24
Request Syntax	24
Request Parameters	24
Response Syntax	24
Response Elements	25

Errors	25
See Also	26
ListTagsForResource	27
Request Syntax	27
Request Parameters	27
Response Syntax	27
Response Elements	28
Errors	28
See Also	29
RestoreTable	30
Request Syntax	30
Request Parameters	31
Response Syntax	33
Response Elements	33
Errors	33
See Also	34
TagResource	35
Request Syntax	35
Request Parameters	35
Response Elements	35
Errors	35
See Also	36
UntagResource	37
Request Syntax	37
Request Parameters	37
Response Elements	37
Errors	37
See Also	38
UpdateTable	39
Request Syntax	39
Request Parameters	39
Response Syntax	41
Response Elements	41
Errors	41
See Also	42
Data Types	43
CapacitySpecification	44
Contents	44
See Also	44
CapacitySpecificationSummary	46
Contents	46
See Also	46
ClusteringKey	48
Contents	48
See Also	48
ColumnDefinition	49
Contents	49
See Also	49
Comment	50
Contents	50
See Also	50
EncryptionSpecification	51
Contents	51
See Also	51
KeyspaceSummary	53
Contents	53
See Also	53

PartitionKey	54
Contents	54
See Also	54
PointInTimeRecovery	55
Contents	55
See Also	55
PointInTimeRecoverySummary	56
Contents	56
See Also	56
SchemaDefinition	57
Contents	57
See Also	57
StaticColumn	58
Contents	58
See Also	58
TableSummary	59
Contents	59
See Also	59
Tag	60
Contents	60
See Also	60
TimeToLive	61
Contents	61
See Also	61
Common Parameters	62
Common Errors	64

Welcome

Amazon Keyspaces (for Apache Cassandra) is a scalable, highly available, and managed Apache Cassandra-compatible database service. Amazon Keyspaces makes it easy to migrate, run, and scale Cassandra workloads in the AWS Cloud. With just a few clicks on the AWS Management Console or a few lines of code, you can create keyspaces and tables in Amazon Keyspaces, without deploying any infrastructure or installing software.

In addition to supporting Cassandra Query Language (CQL) requests via open-source Cassandra drivers, Amazon Keyspaces supports data definition language (DDL) operations to manage keyspaces and tables using the AWS SDK and AWS CLI. This API reference describes the supported DDL operations in detail.

For the list of all supported CQL APIs, see [Supported Cassandra APIs, operations, and data types in Amazon Keyspaces](#) in the *Amazon Keyspaces Developer Guide*.

To learn how Amazon Keyspaces API actions are tracked in AWS CloudTrail, see [Amazon Keyspaces information in CloudTrail](#) in the *Amazon Keyspaces Developer Guide*.

For more information about AWS APIs, for example how to implement retry logic or how to sign AWS API requests, see [AWS APIs](#) in the *General Reference*.

This document was last published on October 4, 2022.

Actions

The following actions are supported:

- [CreateKeyspace](#) (p. 3)
- [CreateTable](#) (p. 6)
- [DeleteKeyspace](#) (p. 11)
- [DeleteTable](#) (p. 13)
- [GetKeyspace](#) (p. 15)
- [GetTable](#) (p. 17)
- [ListKeyspaces](#) (p. 21)
- [ListTables](#) (p. 24)
- [ListTagsForResource](#) (p. 27)
- [RestoreTable](#) (p. 30)
- [TagResource](#) (p. 35)
- [UntagResource](#) (p. 37)
- [UpdateTable](#) (p. 39)

CreateKeyspace

The `CreateKeyspace` operation adds a new keyspace to your account. In an AWS account, keyspace names must be unique within each Region.

`CreateKeyspace` is an asynchronous operation. You can monitor the creation status of the new keyspace by using the `GetKeyspace` operation.

For more information, see [Creating keyspaces](#) in the *Amazon Keyspaces Developer Guide*.

Request Syntax

```
{
  "keyspaceName": "string",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 62).

The request accepts the following data in JSON format.

[keyspaceName](#) (p. 3)

The name of the keyspace to be created.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

[tags](#) (p. 3)

A list of key-value pair tags to be attached to the keyspace.

For more information, see [Adding tags and labels to Amazon Keyspaces resources](#) in the *Amazon Keyspaces Developer Guide*.

Type: Array of [Tag](#) (p. 60) objects

Array Members: Minimum number of 1 item. Maximum number of 60 items.

Required: No

Response Syntax

```
{
```



```
"resourceArn": "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.

resourceArn (p. 3)

The unique identifier of the keyspace in the format of an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*):cassandra:.*`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

Amazon Keyspaces could not complete the requested action. This error may occur if you try to perform an action and the same or a different action is already in progress, or if you try to create a resource that already exists.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

CreateTable

The CreateTable operation adds a new table to the specified keyspace. Within a keyspace, table names must be unique.

CreateTable is an asynchronous operation. When the request is received, the status of the table is set to CREATING. You can monitor the creation status of the new table by using the GetTable operation, which returns the current status of the table. You can start using a table when the status is ACTIVE.

For more information, see [Creating tables](#) in the *Amazon Keyspaces Developer Guide*.

Request Syntax

```
{
  "capacitySpecification": {
    "readCapacityUnits": number,
    "throughputMode": "string",
    "writeCapacityUnits": number
  },
  "comment": {
    "message": "string"
  },
  "defaultTimeToLive": number,
  "encryptionSpecification": {
    "kmsKeyIdentifier": "string",
    "type": "string"
  },
  "keyspaceName": "string",
  "pointInTimeRecovery": {
    "status": "string"
  },
  "schemaDefinition": {
    "allColumns": [
      {
        "name": "string",
        "type": "string"
      }
    ],
    "clusteringKeys": [
      {
        "name": "string",
        "orderBy": "string"
      }
    ],
    "partitionKeys": [
      {
        "name": "string"
      }
    ],
    "staticColumns": [
      {
        "name": "string"
      }
    ]
  },
  "tableName": "string",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ],
}
```

```
"ttl": {  
  "status": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

capacitySpecification (p. 6)

Specifies the read/write throughput capacity mode for the table. The options are:

- `throughputMode:PAY_PER_REQUEST` and
- `throughputMode:PROVISIONED`. The provisioned capacity mode requires `readCapacityUnits` and `writeCapacityUnits` as inputs.

The default is `throughput_mode:PAY_PER_REQUEST`.

For more information, see [Read/write capacity modes](#) in the *Amazon Keyspaces Developer Guide*.

Type: [CapacitySpecification \(p. 44\)](#) object

Required: No

comment (p. 6)

This parameter allows to enter a description of the table.

Type: [Comment \(p. 50\)](#) object

Required: No

defaultTimeToLive (p. 6)

The default Time to Live setting in seconds for the table.

For more information, see [Setting the default TTL value for a table](#) in the *Amazon Keyspaces Developer Guide*.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 630720000.

Required: No

encryptionSpecification (p. 6)

Specifies how the encryption key for encryption at rest is managed for the table. You can choose one of the following KMS key (KMS key):

- `type:AWS_OWNED_KMS_KEY` - This key is owned by Amazon Keyspaces.
- `type:CUSTOMER_MANAGED_KMS_KEY` - This key is stored in your account and is created, owned, and managed by you. This option requires the `kms_key_identifier` of the KMS key in Amazon Resource Name (ARN) format as input.

The default is `type:AWS_OWNED_KMS_KEY`.

For more information, see [Encryption at rest](#) in the *Amazon Keyspaces Developer Guide*.

Type: [EncryptionSpecification \(p. 51\)](#) object

Required: No

[keyspaceName \(p. 6\)](#)

The name of the keyspace that the table is going to be created in.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

[pointInTimeRecovery \(p. 6\)](#)

Specifies if `pointInTimeRecovery` is enabled or disabled for the table. The options are:

- `ENABLED`
- `DISABLED`

If it's not specified, the default is `DISABLED`.

For more information, see [Point-in-time recovery](#) in the *Amazon Keyspaces Developer Guide*.

Type: [PointInTimeRecovery \(p. 55\)](#) object

Required: No

[schemaDefinition \(p. 6\)](#)

The `schemaDefinition` consists of the following parameters.

For each column to be created:

- `name` - The name of the column.
- `type` - An Amazon Keyspaces data type. For more information, see [Data types](#) in the *Amazon Keyspaces Developer Guide*.

The primary key of the table consists of the following columns:

- `partitionKeys` - The partition key can be a single column, or it can be a compound value composed of two or more columns. The partition key portion of the primary key is required and determines how Amazon Keyspaces stores your data.
 - `name` - The name of each partition key column.
- `clusteringKeys` - The optional clustering column portion of your primary key determines how the data is clustered and sorted within each partition.
 - `name` - The name of the clustering column.
 - `orderBy` - Sets the ascendant (`ASC`) or descendant (`DESC`) order modifier.

To define a column as static use `staticColumns` - Static columns store values that are shared by all rows in the same partition:

- `name` - The name of the column.
- `type` - An Amazon Keyspaces data type.

Type: [SchemaDefinition \(p. 57\)](#) object

Required: Yes

[tableName \(p. 6\)](#)

The name of the table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

tags (p. 6)

A list of key-value pair tags to be attached to the resource.

For more information, see [Adding tags and labels to Amazon Keyspaces resources](#) in the *Amazon Keyspaces Developer Guide*.

Type: Array of [Tag \(p. 60\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 60 items.

Required: No

ttl (p. 6)

Enables Time to Live custom settings for the table. The options are:

- `status:enabled`
- `status:disabled`

The default is `status:disabled`. After `ttl` is enabled, you can't disable it for the table.

For more information, see [Expiring data by using Amazon Keyspaces Time to Live \(TTL\)](#) in the *Amazon Keyspaces Developer Guide*.

Type: [TimeToLive \(p. 61\)](#) object

Required: No

Response Syntax

```
{
  "resourceArn": "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.

resourceArn (p. 9)

The unique identifier of the table in the format of an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*):cassandra:.*`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

Amazon Keyspaces could not complete the requested action. This error may occur if you try to perform an action and the same or a different action is already in progress, or if you try to create a resource that already exists.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteKeyspace

The `DeleteKeyspace` operation deletes a keyspace and all of its tables.

Request Syntax

```
{  
  "keyspaceName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

keyspaceName (p. 11)

The name of the keyspace to be deleted.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

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 \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

Amazon Keyspaces could not complete the requested action. This error may occur if you try to perform an action and the same or a different action is already in progress, or if you try to create a resource that already exists.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

DeleteTable

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

Request Syntax

```
{
  "keyspaceName": "string",
  "tableName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

keyspaceName (p. 13)

The name of the keyspace of the to be deleted table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

tableName (p. 13)

The name of the table to be deleted.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

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 \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

Amazon Keyspaces could not complete the requested action. This error may occur if you try to perform an action and the same or a different action is already in progress, or if you try to create a resource that already exists.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetKeyspace

Returns the name and the Amazon Resource Name (ARN) of the specified table.

Request Syntax

```
{  
  "keyspaceName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 62).

The request accepts the following data in JSON format.

keyspaceName (p. 15)

The name of the keyspace.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_]{1,47}

Required: Yes

Response Syntax

```
{  
  "keyspaceName": "string",  
  "resourceArn": "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.

keyspaceName (p. 15)

The name of the keyspace.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_]{1,47}

resourceArn (p. 15)

The ARN of the keyspace.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*):cassandra:.*.*`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

GetTable

Returns information about the table, including the table's name and current status, the keyspace name, configuration settings, and metadata.

To read table metadata using `GetTable`, `Select` action permissions for the table and system tables are required to complete the operation.

Request Syntax

```
{
  "keyspaceName": "string",
  "tableName": "string"
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

keyspaceName (p. 17)

The name of the keyspace that the table is stored in.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

tableName (p. 17)

The name of the table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

Response Syntax

```
{
  "capacitySpecification": {
    "lastUpdateToPayPerRequestTimestamp": number,
    "readCapacityUnits": number,
    "throughputMode": "string",
    "writeCapacityUnits": number
  },
  "comment": {
    "message": "string"
  }
}
```

```

    },
    "creationTimestamp": number,
    "defaultTimeToLive": number,
    "encryptionSpecification": {
      "kmsKeyIdentifier": "string",
      "type": "string"
    },
    },
    "keyspaceName": "string",
    "pointInTimeRecovery": {
      "earliestRestorableTimestamp": number,
      "status": "string"
    },
    },
    "resourceArn": "string",
    "schemaDefinition": {
      "allColumns": [
        {
          "name": "string",
          "type": "string"
        }
      ],
      "clusteringKeys": [
        {
          "name": "string",
          "orderBy": "string"
        }
      ],
      "partitionKeys": [
        {
          "name": "string"
        }
      ],
      "staticColumns": [
        {
          "name": "string"
        }
      ]
    },
    },
    "status": "string",
    "tableName": "string",
    "ttl": {
      "status": "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.

capacitySpecification (p. 17)

The read/write throughput capacity mode for a table. The options are:

- throughputMode:PAY_PER_REQUEST and
- throughputMode:PROVISIONED.

Type: [CapacitySpecificationSummary \(p. 46\)](#) object

comment (p. 17)

The the description of the specified table.

Type: [Comment \(p. 50\)](#) object

[creationTimestamp \(p. 17\)](#)

The creation timestamp of the specified table.

Type: Timestamp

[defaultTimeToLive \(p. 17\)](#)

The default Time to Live settings of the specified table.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 630720000.

[encryptionSpecification \(p. 17\)](#)

The encryption settings of the specified table.

Type: [EncryptionSpecification \(p. 51\)](#) object

[keyspaceName \(p. 17\)](#)

The name of the keyspace that the specified table is stored in.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_]{1,47}

[pointInTimeRecovery \(p. 17\)](#)

The point-in-time recovery status of the specified table.

Type: [PointInTimeRecoverySummary \(p. 56\)](#) object

[resourceArn \(p. 17\)](#)

The Amazon Resource Name (ARN) of the specified table.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: arn:(aws[a-zA-Z0-9-]*):cassandra:.*

[schemaDefinition \(p. 17\)](#)

The schema definition of the specified table.

Type: [SchemaDefinition \(p. 57\)](#) object

[status \(p. 17\)](#)

The current status of the specified table.

Type: String

Valid Values: ACTIVE | CREATING | UPDATING | DELETING | DELETED | RESTORING | INACCESSIBLE_ENCRYPTION_CREDENTIALS

[tableName \(p. 17\)](#)

The name of the specified table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

[ttl \(p. 17\)](#)

The custom Time to Live settings of the specified table.

Type: [TimeToLive \(p. 61\)](#) object

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListKeyspaces

Returns a list of keyspaces.

Request Syntax

```
{  
  "maxResults": number,  
  "nextToken": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

maxResults (p. 21)

The total number of keyspaces to return in the output. If the total number of keyspaces available is more than the value specified, a `NextToken` is provided in the output. To resume pagination, provide the `NextToken` value as an argument of a subsequent API invocation.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

nextToken (p. 21)

The pagination token. To resume pagination, provide the `NextToken` value as argument of a subsequent API invocation.

Type: String

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

Required: No

Response Syntax

```
{  
  "keyspaces": [  
    {  
      "keyspaceName": "string",  
      "resourceArn": "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.

keyspaces (p. 21)

A list of keyspaces.

Type: Array of [KeyspaceSummary \(p. 53\)](#) objects

nextToken (p. 21)

A token to specify where to start paginating. This is the `NextToken` from a previously truncated response.

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 \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListTables

Returns a list of tables for a specified keyspace.

Request Syntax

```
{  
  "keyspaceName": "string",  
  "maxResults": number,  
  "nextToken": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 62).

The request accepts the following data in JSON format.

keyspaceName (p. 24)

The name of the keyspace.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

maxResults (p. 24)

The total number of tables to return in the output. If the total number of tables available is more than the value specified, a `NextToken` is provided in the output. To resume pagination, provide the `NextToken` value as an argument of a subsequent API invocation.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

nextToken (p. 24)

The pagination token. To resume pagination, provide the `NextToken` value as an argument of a subsequent API invocation.

Type: String

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

Required: No

Response Syntax

```
{  
  "nextToken": "string",  
}
```

```
"tables": [  
  {  
    "keyspaceName": "string",  
    "resourceArn": "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.

nextToken (p. 24)

A token to specify where to start paginating. This is the `NextToken` from a previously truncated response.

Type: String

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

tables (p. 24)

A list of tables.

Type: Array of [TableSummary](#) (p. 59) objects

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 64).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerError

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

ListTagsForResource

Returns a list of all tags associated with the specified Amazon Keyspaces resource.

Request Syntax

```
{  
  "maxResults": number,  
  "nextToken": "string",  
  "resourceArn": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

maxResults (p. 27)

The total number of tags to return in the output. If the total number of tags available is more than the value specified, a `NextToken` is provided in the output. To resume pagination, provide the `NextToken` value as an argument of a subsequent API invocation.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 1000.

Required: No

nextToken (p. 27)

The pagination token. To resume pagination, provide the `NextToken` value as argument of a subsequent API invocation.

Type: String

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

Required: No

resourceArn (p. 27)

The Amazon Resource Name (ARN) of the Amazon Keyspaces resource.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*):cassandra:.*`

Required: Yes

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 (p. 27)

A token to specify where to start paginating. This is the `NextToken` from a previously truncated response.

Type: String

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

tags (p. 27)

A list of tags.

Type: Array of [Tag](#) (p. 60) objects

Array Members: Minimum number of 1 item. Maximum number of 60 items.

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 64).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

RestoreTable

Restores the specified table to the specified point in time within the `earliest_restorable_timestamp` and the current time. For more information about restore points, see [Time window for PITR continuous backups](#) in the *Amazon Keyspaces Developer Guide*.

Any number of users can execute up to 4 concurrent restores (any type of restore) in a given account.

When you restore using point in time recovery, Amazon Keyspaces restores your source table's schema and data to the state based on the selected timestamp (`day:hour:minute:second`) to a new table. The Time to Live (TTL) settings are also restored to the state based on the selected timestamp.

In addition to the table's schema, data, and TTL settings, `RestoreTable` restores the capacity mode, encryption, and point-in-time recovery settings from the source table. Unlike the table's schema data and TTL settings, which are restored based on the selected timestamp, these settings are always restored based on the table's settings as of the current time or when the table was deleted.

You can also overwrite these settings during restore:

- Read/write capacity mode
- Provisioned throughput capacity settings
- Point-in-time (PITR) settings
- Tags

For more information, see [PITR restore settings](#) in the *Amazon Keyspaces Developer Guide*.

The following settings are not restored, and you must configure them manually for the new table.

- Automatic scaling policies (for tables that use provisioned capacity mode)
- AWS Identity and Access Management (IAM) policies
- Amazon CloudWatch metrics and alarms

Request Syntax

```
{
  "capacitySpecificationOverride": {
    "readCapacityUnits": number,
    "throughputMode": "string",
    "writeCapacityUnits": number
  },
  "encryptionSpecificationOverride": {
    "kmsKeyIdentifier": "string",
    "type": "string"
  },
  "pointInTimeRecoveryOverride": {
    "status": "string"
  },
  "restoreTimestamp": number,
  "sourceKeyspaceName": "string",
  "sourceTableName": "string",
  "tagsOverride": [
    {
      "key": "string",
      "value": "string"
    }
  ],
}
```

```
"targetKeyspaceName": "string",  
"targetTableName": "string"  
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters](#) (p. 62).

The request accepts the following data in JSON format.

capacitySpecificationOverride (p. 30)

Specifies the read/write throughput capacity mode for the target table. The options are:

- `throughputMode:PAY_PER_REQUEST` and
- `throughputMode:PROVISIONED`. The provisioned capacity mode requires `readCapacityUnits` and `writeCapacityUnits` as inputs.

The default is `throughput_mode:PAY_PER_REQUEST`.

For more information, see [Read/write capacity modes](#) in the *Amazon Keyspaces Developer Guide*.

Type: [CapacitySpecification](#) (p. 44) object

Required: No

encryptionSpecificationOverride (p. 30)

Specifies the encryption settings for the target table. You can choose one of the following KMS key (KMS key):

- `type:AWS_OWNED_KMS_KEY` - This key is owned by Amazon Keyspaces.
- `type:CUSTOMER_MANAGED_KMS_KEY` - This key is stored in your account and is created, owned, and managed by you. This option requires the `kms_key_identifier` of the KMS key in Amazon Resource Name (ARN) format as input.

The default is `type:AWS_OWNED_KMS_KEY`.

For more information, see [Encryption at rest](#) in the *Amazon Keyspaces Developer Guide*.

Type: [EncryptionSpecification](#) (p. 51) object

Required: No

pointInTimeRecoveryOverride (p. 30)

Specifies the `pointInTimeRecovery` settings for the target table. The options are:

- `ENABLED`
- `DISABLED`

If it's not specified, the default is `DISABLED`.

For more information, see [Point-in-time recovery](#) in the *Amazon Keyspaces Developer Guide*.

Type: [PointInTimeRecovery](#) (p. 55) object

Required: No

restoreTimestamp (p. 30)

The restore timestamp in ISO 8601 format.

Type: Timestamp

Required: No

sourceKeyspaceName (p. 30)

The keyspace name of the source table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_]{1,47}

Required: Yes

sourceTableName (p. 30)

The name of the source table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_]{1,47}

Required: Yes

tagsOverride (p. 30)

A list of key-value pair tags to be attached to the restored table.

For more information, see [Adding tags and labels to Amazon Keyspaces resources](#) in the *Amazon Keyspaces Developer Guide*.

Type: Array of [Tag \(p. 60\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 60 items.

Required: No

targetKeyspaceName (p. 30)

The name of the target keyspace.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_]{1,47}

Required: Yes

targetTableName (p. 30)

The name of the target table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: [a-zA-Z0-9][a-zA-Z0-9_]{1,47}

Required: Yes

Response Syntax

```
{  
  "restoredTableARN": "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.

restoredTableARN (p. 33)

The Amazon Resource Name (ARN) of the restored table.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*):cassandra:.*.*`

Errors

For information about the errors that are common to all actions, see [Common Errors](#) (p. 64).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

Amazon Keyspaces could not complete the requested action. This error may occur if you try to perform an action and the same or a different action is already in progress, or if you try to create a resource that already exists.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

TagResource

Associates a set of tags with a Amazon Keyspaces resource. You can then activate these user-defined tags so that they appear on the Cost Management Console for cost allocation tracking. For more information, see [Adding tags and labels to Amazon Keyspaces resources](#) in the *Amazon Keyspaces Developer Guide*.

For IAM policy examples that show how to control access to Amazon Keyspaces resources based on tags, see [Amazon Keyspaces resource access based on tags](#) in the *Amazon Keyspaces Developer Guide*.

Request Syntax

```
{
  "resourceArn": "string",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

[resourceArn \(p. 35\)](#)

The Amazon Resource Name (ARN) of the Amazon Keyspaces resource to which to add tags.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*) :cassandra: .+.*`

Required: Yes

[tags \(p. 35\)](#)

The tags to be assigned to the Amazon Keyspaces resource.

Type: Array of [Tag \(p. 60\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 60 items.

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 \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UntagResource

Removes the association of tags from a Amazon Keyspaces resource.

Request Syntax

```
{
  "resourceArn": "string",
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ]
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

resourceArn (p. 37)

The Amazon Keyspaces resource that the tags will be removed from. This value is an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*) :cassandra:.*`

Required: Yes

tags (p. 37)

A list of existing tags to be removed from the Amazon Keyspaces resource.

Type: Array of [Tag \(p. 60\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 60 items.

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 \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

Amazon Keyspaces could not complete the requested action. This error may occur if you try to perform an action and the same or a different action is already in progress, or if you try to create a resource that already exists.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

UpdateTable

Adds new columns to the table or updates one of the table's settings, for example capacity mode, encryption, point-in-time recovery, or ttl settings. Note that you can only update one specific table setting per update operation.

Request Syntax

```
{
  "addColumnns": [
    {
      "name": "string",
      "type": "string"
    }
  ],
  "capacitySpecification": {
    "readCapacityUnits": number,
    "throughputMode": "string",
    "writeCapacityUnits": number
  },
  "defaultTimeToLive": number,
  "encryptionSpecification": {
    "kmsKeyIdentifier": "string",
    "type": "string"
  },
  "keyspaceName": "string",
  "pointInTimeRecovery": {
    "status": "string"
  },
  "tableName": "string",
  "ttl": {
    "status": "string"
  }
}
```

Request Parameters

For information about the parameters that are common to all actions, see [Common Parameters \(p. 62\)](#).

The request accepts the following data in JSON format.

addColumnns (p. 39)

For each column to be added to the specified table:

- *name* - The name of the column.
- *type* - An Amazon Keyspaces data type. For more information, see [Data types](#) in the *Amazon Keyspaces Developer Guide*.

Type: Array of [ColumnDefinition \(p. 49\)](#) objects

Array Members: Minimum number of 1 item.

Required: No

capacitySpecification (p. 39)

Modifies the read/write throughput capacity mode for the table. The options are:

- `throughputMode:PAY_PER_REQUEST` and

- `throughputMode:PROVISIONED`. The provisioned capacity mode requires `readCapacityUnits` and `writeCapacityUnits` as inputs.

The default is `throughput_mode:PAY_PER_REQUEST`.

For more information, see [Read/write capacity modes](#) in the *Amazon Keyspaces Developer Guide*.

Type: [CapacitySpecification \(p. 44\)](#) object

Required: No

defaultTimeToLive (p. 39)

The default Time to Live setting in seconds for the table.

For more information, see [Setting the default TTL value for a table](#) in the *Amazon Keyspaces Developer Guide*.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 630720000.

Required: No

encryptionSpecification (p. 39)

Modifies the encryption settings of the table. You can choose one of the following KMS key (KMS key):

- `type:AWS_OWNED_KMS_KEY` - This key is owned by Amazon Keyspaces.
- `type:CUSTOMER_MANAGED_KMS_KEY` - This key is stored in your account and is created, owned, and managed by you. This option requires the `kms_key_identifier` of the KMS key in Amazon Resource Name (ARN) format as input.

The default is `AWS_OWNED_KMS_KEY`.

For more information, see [Encryption at rest](#) in the *Amazon Keyspaces Developer Guide*.

Type: [EncryptionSpecification \(p. 51\)](#) object

Required: No

keyspaceName (p. 39)

The name of the keyspace the specified table is stored in.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

pointInTimeRecovery (p. 39)

Modifies the `pointInTimeRecovery` settings of the table. The options are:

- `ENABLED`
- `DISABLED`

If it's not specified, the default is `DISABLED`.

For more information, see [Point-in-time recovery](#) in the *Amazon Keyspaces Developer Guide*.

Type: [PointInTimeRecovery \(p. 55\)](#) object

Required: No

tableName (p. 39)

The name of the table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

ttl (p. 39)

Modifies Time to Live custom settings for the table. The options are:

- `status:enabled`
- `status:disabled`

The default is `status:disabled`. After `ttl` is enabled, you can't disable it for the table.

For more information, see [Expiring data by using Amazon Keyspaces Time to Live \(TTL\)](#) in the *Amazon Keyspaces Developer Guide*.

Type: [TimeToLive \(p. 61\)](#) object

Required: No

Response Syntax

```
{  
  "resourceArn": "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.

resourceArn (p. 41)

The Amazon Resource Name (ARN) of the modified table.

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*):cassandra:.*.*`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 64\)](#).

AccessDeniedException

You do not have sufficient access to perform this action.

HTTP Status Code: 400

ConflictException

Amazon Keyspaces could not complete the requested action. This error may occur if you try to perform an action and the same or a different action is already in progress, or if you try to create a resource that already exists.

HTTP Status Code: 400

InternalServerErrorException

Amazon Keyspaces was unable to fully process this request because of an internal server error.

HTTP Status Code: 500

ResourceNotFoundException

The operation tried to access a keyspace or table that doesn't exist. The resource might not be specified correctly, or its status might not be `ACTIVE`.

HTTP Status Code: 400

ServiceQuotaExceededException

The operation exceeded the service quota for this resource. For more information on service quotas, see [Quotas](#) in the *Amazon Keyspaces Developer Guide*.

HTTP Status Code: 400

ValidationException

The operation failed due to an invalid or malformed request.

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](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

Data Types

The Amazon Keyspaces API contains several data types that various actions use. This section describes each data type in detail.

Note

The order of each element in a data type structure is not guaranteed. Applications should not assume a particular order.

The following data types are supported:

- [CapacitySpecification](#) (p. 44)
- [CapacitySpecificationSummary](#) (p. 46)
- [ClusteringKey](#) (p. 48)
- [ColumnDefinition](#) (p. 49)
- [Comment](#) (p. 50)
- [EncryptionSpecification](#) (p. 51)
- [KeyspaceSummary](#) (p. 53)
- [PartitionKey](#) (p. 54)
- [PointInTimeRecovery](#) (p. 55)
- [PointInTimeRecoverySummary](#) (p. 56)
- [SchemaDefinition](#) (p. 57)
- [StaticColumn](#) (p. 58)
- [TableSummary](#) (p. 59)
- [Tag](#) (p. 60)
- [TimeToLive](#) (p. 61)

CapacitySpecification

Amazon Keyspaces has two read/write capacity modes for processing reads and writes on your tables:

- On-demand (default)
- Provisioned

The read/write capacity mode that you choose controls how you are charged for read and write throughput and how table throughput capacity is managed.

For more information, see [Read/write capacity modes](#) in the *Amazon Keyspaces Developer Guide*.

Contents

readCapacityUnits

The throughput capacity specified for read operations defined in `read capacity units (RCUs)`.

Type: Long

Valid Range: Minimum value of 1.

Required: No

throughputMode

The read/write throughput capacity mode for a table. The options are:

- `throughputMode:PAY_PER_REQUEST` and
- `throughputMode:PROVISIONED`. The provisioned capacity mode requires `readCapacityUnits` and `writeCapacityUnits` as inputs.

The default is `throughput_mode:PAY_PER_REQUEST`.

For more information, see [Read/write capacity modes](#) in the *Amazon Keyspaces Developer Guide*.

Type: String

Valid Values: `PAY_PER_REQUEST` | `PROVISIONED`

Required: Yes

writeCapacityUnits

The throughput capacity specified for write operations defined in `write capacity units (WCUs)`.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

CapacitySpecificationSummary

The read/write throughput capacity mode for a table. The options are:

- `throughputMode:PAY_PER_REQUEST` and
- `throughputMode:PROVISIONED`.

For more information, see [Read/write capacity modes](#) in the *Amazon Keyspaces Developer Guide*.

Contents

lastUpdateToPayPerRequestTimestamp

The timestamp of the last operation that changed the provisioned throughput capacity of a table.

Type: Timestamp

Required: No

readCapacityUnits

The throughput capacity specified for read operations defined in `read capacity units (RCUs)`.

Type: Long

Valid Range: Minimum value of 1.

Required: No

throughputMode

The read/write throughput capacity mode for a table. The options are:

- `throughputMode:PAY_PER_REQUEST` and
- `throughputMode:PROVISIONED`. The provisioned capacity mode requires `readCapacityUnits` and `writeCapacityUnits` as inputs.

The default is `throughput_mode:PAY_PER_REQUEST`.

For more information, see [Read/write capacity modes](#) in the *Amazon Keyspaces Developer Guide*.

Type: String

Valid Values: `PAY_PER_REQUEST` | `PROVISIONED`

Required: Yes

writeCapacityUnits

The throughput capacity specified for write operations defined in `write capacity units (WCUs)`.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ClusteringKey

The optional clustering column portion of your primary key determines how the data is clustered and sorted within each partition.

Contents

name

The name(s) of the clustering column(s).

Type: String

Required: Yes

orderBy

Sets the ascendant (ASC) or descendant (DESC) order modifier.

Type: String

Valid Values: ASC | DESC

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ColumnDefinition

The names and data types of regular columns.

Contents

name

The name of the column.

Type: String

Required: Yes

type

The data type of the column. For a list of available data types, see [Data types](#) in the *Amazon Keyspaces Developer Guide*.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Comment

An optional comment that describes the table.

Contents

message

An optional description of the 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

EncryptionSpecification

Amazon Keyspaces encrypts and decrypts the table data at rest transparently and integrates with AWS Key Management Service for storing and managing the encryption key. You can choose one of the following AWS KMS keys (KMS keys):

- **AWS owned key** - This is the default encryption type. The key is owned by Amazon Keyspaces (no additional charge).
- **Customer managed key** - This key is stored in your account and is created, owned, and managed by you. You have full control over the customer managed key (AWS KMS charges apply).

For more information about encryption at rest in Amazon Keyspaces, see [Encryption at rest](#) in the *Amazon Keyspaces Developer Guide*.

For more information about AWS KMS, see [AWS KMS management service concepts](#) in the *AWS Key Management Service Developer Guide*.

Contents

kmsKeyIdIdentifier

The Amazon Resource Name (ARN) of the customer managed KMS key, for example `kms_key_identifier:ARN`.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 5096.

Required: No

type

The encryption option specified for the table. You can choose one of the following KMS keys (KMS keys):

- `type:AWS_OWNED_KMS_KEY` - This key is owned by Amazon Keyspaces.
- `type:CUSTOMER_MANAGED_KMS_KEY` - This key is stored in your account and is created, owned, and managed by you. This option requires the `kms_key_identifier` of the KMS key in Amazon Resource Name (ARN) format as input.

The default is `type:AWS_OWNED_KMS_KEY`.

For more information, see [Encryption at rest](#) in the *Amazon Keyspaces Developer Guide*.

Type: String

Valid Values: `CUSTOMER_MANAGED_KMS_KEY` | `AWS_OWNED_KMS_KEY`

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 Go](#)

- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

KeyspaceSummary

Represents the properties of a keyspace.

Contents

keyspaceName

The name of the keyspace.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

resourceArn

The unique identifier of the keyspace in the format of an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*):cassandra:.*.*`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PartitionKey

The partition key portion of the primary key is required and determines how Amazon Keyspaces stores the data. The partition key can be a single column, or it can be a compound value composed of two or more columns.

Contents

name

The name(s) of the partition key column(s).

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PointInTimeRecovery

Point-in-time recovery (PITR) helps protect your Amazon Keyspaces tables from accidental write or delete operations by providing you continuous backups of your table data.

For more information, see [Point-in-time recovery](#) in the *Amazon Keyspaces Developer Guide*.

Contents

status

The options are:

- `ENABLED`
- `DISABLED`

Type: String

Valid Values: `ENABLED` | `DISABLED`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

PointInTimeRecoverySummary

The point-in-time recovery status of the specified table.

Contents

earliestRestorableTimestamp

Specifies the earliest possible restore point of the table in ISO 8601 format.

Type: Timestamp

Required: No

status

Shows if point-in-time recovery is enabled or disabled for the specified table.

Type: String

Valid Values: `ENABLED` | `DISABLED`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

SchemaDefinition

Describes the schema of the table.

Contents

allColumns

The regular columns of the table.

Type: Array of [ColumnDefinition \(p. 49\)](#) objects

Array Members: Minimum number of 1 item.

Required: Yes

clusteringKeys

The columns that are part of the clustering key of the table.

Type: Array of [ClusteringKey \(p. 48\)](#) objects

Required: No

partitionKeys

The columns that are part of the partition key of the table .

Type: Array of [PartitionKey \(p. 54\)](#) objects

Array Members: Minimum number of 1 item.

Required: Yes

staticColumns

The columns that have been defined as `STATIC`. Static columns store values that are shared by all rows in the same partition.

Type: Array of [StaticColumn \(p. 58\)](#) 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

StaticColumn

The static columns of the table. Static columns store values that are shared by all rows in the same partition.

Contents

name

The name of the static column.

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TableSummary

Returns the name of the specified table, the keyspace it is stored in, and the unique identifier in the format of an Amazon Resource Name (ARN).

Contents

keyspaceName

The name of the keyspace that the table is stored in.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

Required: Yes

resourceArn

The unique identifier of the table in the format of an Amazon Resource Name (ARN).

Type: String

Length Constraints: Minimum length of 20. Maximum length of 1000.

Pattern: `arn:(aws[a-zA-Z0-9-]*):cassandra:.+.*`

Required: Yes

tableName

The name of the table.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 48.

Pattern: `[a-zA-Z0-9][a-zA-Z0-9_]{1,47}`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Tag

Describes a tag. A tag is a key-value pair. You can add up to 50 tags to a single Amazon Keyspaces resource.

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 more information, see [Adding tags and labels to Amazon Keyspaces resources](#) in the *Amazon Keyspaces Developer Guide*.

Contents

key

The key of the tag. Tag keys are case sensitive. Each Amazon Keyspaces resource 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 1. 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

TimeToLive

Enable custom Time to Live (TTL) settings for rows and columns without setting a TTL default for the specified table.

For more information, see [Enabling TTL on tables](#) in the *Amazon Keyspaces Developer Guide*.

Contents

status

Shows how to enable custom Time to Live (TTL) settings for the specified table.

Type: String

Valid Values: `ENABLED`

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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

Common Parameters

The following list contains the parameters that all actions use for signing Signature Version 4 requests with a query string. Any action-specific parameters are listed in the topic for that action. For more information about Signature Version 4, see [Signature Version 4 Signing Process](#) in the *Amazon Web Services General Reference*.

Action

The action to be performed.

Type: string

Required: Yes

Version

The API version that the request is written for, expressed in the format YYYY-MM-DD.

Type: string

Required: Yes

X-Amz-Algorithm

The hash algorithm that you used to create the request signature.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Valid Values: `AWS4-HMAC-SHA256`

Required: Conditional

X-Amz-Credential

The credential scope value, which is a string that includes your access key, the date, the region you are targeting, the service you are requesting, and a termination string ("aws4_request"). The value is expressed in the following format: `access_key/YYYYMMDD/region/service/aws4_request`.

For more information, see [Task 2: Create a String to Sign for Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-Date

The date that is used to create the signature. The format must be ISO 8601 basic format (YYYYMMDD'THHMMSS'Z'). For example, the following date time is a valid X-Amz-Date value: `20120325T120000Z`.

Condition: X-Amz-Date is optional for all requests; it can be used to override the date used for signing requests. If the Date header is specified in the ISO 8601 basic format, X-Amz-Date is

not required. When X-Amz-Date is used, it always overrides the value of the Date header. For more information, see [Handling Dates in Signature Version 4](#) in the *Amazon Web Services General Reference*.

Type: string

Required: Conditional

X-Amz-Security-Token

The temporary security token that was obtained through a call to AWS Security Token Service (AWS STS). For a list of services that support temporary security credentials from AWS Security Token Service, go to [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Condition: If you're using temporary security credentials from the AWS Security Token Service, you must include the security token.

Type: string

Required: Conditional

X-Amz-Signature

Specifies the hex-encoded signature that was calculated from the string to sign and the derived signing key.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

X-Amz-SignedHeaders

Specifies all the HTTP headers that were included as part of the canonical request. For more information about specifying signed headers, see [Task 1: Create a Canonical Request For Signature Version 4](#) in the *Amazon Web Services General Reference*.

Condition: Specify this parameter when you include authentication information in a query string instead of in the HTTP authorization header.

Type: string

Required: Conditional

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

InvalidParameterCombination

Parameters that must not be used together were used together.

HTTP Status Code: 400

InvalidParameterValue

An invalid or out-of-range value was supplied for the input parameter.

HTTP Status Code: 400

InvalidQueryParameter

The AWS query string is malformed or does not adhere to AWS standards.

HTTP Status Code: 400

MalformedQueryString

The query string contains a syntax error.

HTTP Status Code: 404

MissingAction

The request is missing an action or a required parameter.

HTTP Status Code: 400

MissingAuthenticationToken

The request must contain either a valid (registered) AWS access key ID or X.509 certificate.

HTTP Status Code: 403

MissingParameter

A required parameter for the specified action is not supplied.

HTTP Status Code: 400

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