
AWS Marketplace Catalog API Reference



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

AWS Marketplace Catalog API	1
Catalog API entities	1
General structure	1
Versioning	1
Identifier	2
Facets	2
Product entity	2
Supported AWS Regions	3
Working with change sets	3
Working with the <code>Details</code> attribute	4
API access control	5
Set up IAM permissions	5
Condition keys	6
Service limits	6
Working with a private marketplace	8
Creating a private marketplace	8
Changing the branding of your private marketplace	10
Enabling or disabling your private marketplace	12
Enabling or disabling user requests	12
Getting a list of products in your private marketplace	13
Adding or removing products from your private marketplace	14
Finding products	15
Working with private marketplaces for AWS Organizations	15
Errors in the private marketplace API	16
Logging AWS Marketplace Catalog API calls with CloudTrail	18
AWS Marketplace Catalog API information in CloudTrail	18
Understanding AWS Marketplace catalog log file entries	19
Discovery API	20
Getting access to the Discovery API	20
Actions	21
CancelChangeSet	22
Request Syntax	22
URI Request Parameters	22
Request Body	22
Response Syntax	22
Response Elements	22
Errors	23
See Also	23
DescribeChangeSet	25
Request Syntax	25
URI Request Parameters	25
Request Body	25
Response Syntax	25
Response Elements	26
Errors	27
See Also	28
DescribeEntity	29
Request Syntax	29
URI Request Parameters	29
Request Body	29
Response Syntax	29
Response Elements	29
Errors	30
See Also	31

ListChangeSets	32
Request Syntax	32
URI Request Parameters	32
Request Body	32
Response Syntax	33
Response Elements	33
Errors	34
See Also	34
ListEntities	35
Request Syntax	35
URI Request Parameters	35
Request Body	35
Response Syntax	36
Response Elements	37
Errors	37
See Also	37
StartChangeSet	39
Request Syntax	39
URI Request Parameters	39
Request Body	39
Response Syntax	40
Response Elements	40
Errors	41
Examples	41
See Also	42
Data Types	43
Change	44
Contents	44
See Also	44
ChangeSetSummaryListItem	45
Contents	45
See Also	46
ChangeSummary	47
Contents	47
See Also	47
Entity	48
Contents	48
See Also	48
EntitySummary	49
Contents	49
See Also	50
ErrorDetail	51
Contents	51
See Also	51
Filter	52
Contents	52
See Also	52
Sort	53
Contents	53
See Also	53
Common Parameters	54
Common Errors	56
Document history	58

AWS Marketplace Catalog API

The AWS Marketplace Catalog API service provides an API interface to manage AWS Marketplace for your AWS Organization or AWS account. For approved sellers, you can programmatically manage your products, including the self-service publishing capabilities on the [AWS Marketplace Management Portal](#). For private marketplace administrators, you can programmatically manage your private marketplace.

Catalog API actions allow you to view and update your existing product programmatically. You can automate your product update process by integrating the AWS Marketplace Catalog API with your AWS Marketplace product build or deployment pipelines. You can also create your own applications on top of the Catalog API to manage your products on AWS Marketplace. You can manage the products that users in your AWS account or AWS organization can see and purchase through your private marketplace.

The AWS Marketplace Catalog API service provides standard AWS API functionality. You can directly use the Rest APIs described here (see [Getting Started with AWS](#) to learn more about AWS application development), or you can use one of the AWS SDKs to access an API that's tailored to the programming language or platform that you're using. For more information, see [AWS SDKs](#).

Catalog API entities

AWS Marketplace entities are containers of data which serve different business purposes. Entities are categorized by types. Each entity type encapsulates data related to a specific business domain (for example, a product or a seller account).

To simplify this paradigm, entities are designed with some level of commonality in their structures. As a result, introducing a new business domain does not require you to learn a completely new structure.

General structure

The general structure of any entity is:

- A named type
- An identifier
- A facet

Versioning

Every named type has a type and version associated with it, for example, *EntityProduct@1.0*. The *type* (*EntityProduct*) represents the classification of the content. The *version* (1.0) represents the structure of *EntityProduct*.

The version gives you details about the structure of the entity.

- Existing entities won't be restructured without changing the version. Additions of optional new fields will result in a minor version update.
- Any feature that fundamentally changes the structure of a type leads to a major version update. Examples include:
 - Removing a field
 - Renaming a field (different name for the same semantic)

- Changing the semantic of an existing field (for example, changing the expected type)
- A major version update can retain a subset of facets from the previous version.
- Users are provided notifications and documentation for new versions.

Identifier

Each entity represents a unique *thing* within a business domain. To identify the unique thing, we use an identifier associating an EntityId with a RevisionId, for example, prod-**ad8EXAMPLE651@12345**. In this example, the EntityId is prod-ad8EXAMPLE651 and the RevisionId is 12345. Every successful change request to the entity will update the revision.

- Each entity is uniquely identified by its EntityId, which is the key to globally distinguish one entity from another.
- Each published revision of an entity has a RevisionId. The revisionId, along with the EntityId, distinguish one published revision from another.
- AWS Marketplace generates EntityIds and RevisionIds.

You can use the `DescribeEntity` action to find the details and the Identifier with the most recent revisionId.

The `RevisionId` is an optional part of requests to `StartChangeSet` (see [Working with change sets \(p. 3\)](#)). If you include a `RevisionId`, then the request to `StartChangeSet` will fail with a `ValidationException` if the `RevisionId` is not the latest revision of the entity. This allows you to implement optimistic locking in your application.

Note

When you include a `RevisionId` that is not the latest revision, the `ValidationException` message includes the latest `RevisionId`.

If you omit the `RevisionId`, the request is performed on the latest revision of the entity automatically.

Warning

Two requests to change the same object could result with one request overwriting the changes of the other request, as the second request rewrites data changed by the first request. Using `RevisionIds` in your requests prevents this issue by not allowing a change to an earlier revision to overwrite the current revision.

Facets

A facet is a logical grouping of attributes. An entity usually includes several facets which represent different aspects of the entity. The attributes within a facet have the following properties.

- Each attribute has a unique name within the scope of the container it belongs to.
- Attributes can be of a simple type (string, integer, or floating number).
- Attributes can be of a complex type (container/structure or array).

Product entity

A software product you own and list on AWS Marketplace is represented by a product entity. Product entities have different types. Regardless of type, product entities have some common facets in addition to product type specific facets. The example below is an example of common facets: `Description`, `PromotionalResources`, `RegionAvailability`, and `SupportInformation`. `Details` is an example of a product type specific facet.

```
{
  "Details": "{\\\"Description\\\":{}, \\\"PromotionalResources\\\":{}, \\\"RegionAvailability\\\":{},
  \\\"SupportInformation\\\":{}}",
  "EntityArn": "arn:aws:aws-marketplace:us-east-1:0123456789012:AWSMarketplace/Entity-
Type/9EXAMPLE-0123-4567-8901-74eEXAMPLE47",
  "EntityIdentifier": "9EXAMPLE-0123-4567-8901-74eEXAMPLE47@23",
  "EntityType": "Entity-Type@1.0",
  "LastModifiedDate": "2019-07-31T21:59:39Z"
}
```

The Details facet is a string that includes JSON. For more information, see [Working with the Details attribute \(p. 4\)](#).

Supported AWS Regions

You can access the AWS Marketplace Catalog API from the US East (N. Virginia) AWS Region with the following endpoint.

```
catalog.marketplace.us-east-1.amazonaws.com
```

Working with change sets

When using the Catalog API, you perform many actions by creating change requests. You do this by creating and working with change sets. A change set is a list of change entities, each of which represents a request for a change in AWS Marketplace. You can make changes to the products that you provide (as a seller), or to the private marketplace that you maintain (as a buyer).

Note

Although you can request multiple changes in a single change set, you cannot request conflicting changes. You will receive a `ResourceInUseException` error in this case. This is true even across multiple change sets. If you want to modify the same entity twice, the safest way is to make a request, wait for it to complete, and then request the next change. For information about using `RevisionIds` in this case, see [Identifier \(p. 2\)](#).

There are four actions that allow you to work with change sets:

- `StartChangeSet` – Requests a set of changes. The changes are added to a queue and processed.
- `DescribeChangeSet` – Gets the details of a set of changes, including the status of the request. The statuses include:
 - `PREPARING` – Getting ready to apply the changes.
 - `APPLYING` – In the process of making the requested changes.
 - `SUCCEEDED` – Request was completed successfully.
 - `CANCELED` – Request was canceled by the user.
 - `FAILED` – Request was completed unsuccessfully. Further details are available in the response.
- `ListChangeSets` – Gets a list of the change sets that are currently in process.
- `CancelChangeSet` – Requests a change set be canceled. Changes can only be canceled while in the `PREPARING` status.

A typical workflow is to request a change with `StartChangeSet`, and then use the returned `ChangeSetId` to poll the `DescribeChangeSet` action until the change is complete.

Note

When polling or working with change sets programmatically, you must follow the [Service limits \(p. 6\)](#).

After your change is complete, you can use `ListEntities` to find the entity that you created or modified (and its associated `EntityID`). You can then use `DescribeEntity` with the `EntityID` to get details about it.

For more information about working with Change Requests in the console for sellers, see [Creating a change request](#) in the *AWS Marketplace Seller Guide*.

Working with the Details attribute

The `Details` attribute of the `StartChangeSet` operation is a string value. Its contents are JSON objects. To put a JSON object into a string attribute, you must convert the object to strings by escaping all JSON control characters, and removing linebreaks.

For example, if you are using the `StartChangeSet` operation with `UpdateProcurementPolicy` to disable requests from users in your private marketplace, you would make a request like the following.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateProcurementPolicy",
      "Details": "<string>",
      "Entity": {
        "Type": "Experience@1.0",
        "Identifier" : "exp-1234example@5"
      }
    }
  ]
}
```

In this case, the JSON object that you use for the `Details` attribute would look like the following

```
{
  "Configuration": {
    "PolicyResourceRequests": "Deny"
  }
}
```

After converting this JSON object to a single line string, it would look like the following.

```
"{\\"Configuration\\" : {\\"PolicyResourceRequests\\" : \\"Deny\\"}}"
```

With this string, you can create the full change set request, as follows.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
```



```
{
  "ChangeType": "UpdateProcurementPolicy",
  "Details": "{\"Configuration\" : {\"PolicyResourceRequests\" : \"Deny\"}}",
  "Entity": {
    "Type": "Experience@1.0",
    "Identifier" : "exp-1234example@5"
  }
}
```

Generally, examples in this API reference will show the JSON object already converted to a string. In some cases, more complicated samples with new lines are included for understandability.

API access control

Before you can use the AWS Marketplace Catalog API, your account must have access to the functionality you want to call through the API.

You must create AWS Identity and Access Management (IAM) users, roles, and policies before you can use the AWS Marketplace Catalog API.

Use [AWS Identity and Access Management](#) to create IAM roles and assign policies that grant limited permissions to end users. The policies define the actions the role can take on your product entities through the AWS Marketplace Catalog API. For example, you can define roles such as engineering, marketing, and pricing. A user in your organization who has been added to the engineering role might be granted permissions to initiate a change request to publish a new version but cannot list all change sets.

Prerequisite for AWS Marketplace sellers: To sell products on AWS Marketplace, your seller account must be added to the approved list. Otherwise, your requests will fail. For more details about becoming an AWS Marketplace seller, see [Getting started as a seller](#) in the *AWS Marketplace Seller Guide*.

Set up IAM permissions

You can use policies that are managed by AWS to grant IAM permissions to your users.

To manage a private marketplace, you can use the `AWSPrivateMarketplaceAdminFullAccess` IAM managed policy which has full access to create and edit the private marketplace for your account or AWS organization.

To work with products that you sell on AWS Marketplace, you can use the `AWSMarketplaceSellerFullAccess` IAM managed policy which has full access to the AWS Marketplace Catalog API in addition to its other permissions. You can grant read-only access for the Catalog API with the `AWSMarketplaceSellerProductsReadOnly` policy.

For more details about these policies, their permissions, and other IAM managed policies, sign into the IAM console at <https://console.aws.amazon.com/iam/>, choose **Policies**, and enter *marketplace* in the **Search** field.

You can also create your own policies or limit the scope of managed policies to a subset of functionality available in the AWS Marketplace Catalog API. The following is a list of the actions that you can use in your IAM policies for scoping permissions to the AWS Marketplace Catalog API:

- `aws-marketplace:ListChangeSets`
- `aws-marketplace:DescribeChangeSet`

- `aws-marketplace:StartChangeSet`
- `aws-marketplace:CancelChangeSet`
- `aws-marketplace:ListEntities`
- `aws-marketplace:DescribeEntity`

For more information about using policies in AWS Marketplace, see the following topics:

- For buyers who want to manage a private marketplace, see [Creating a private marketplace IT administrator](#), and [Controlling access to AWS Marketplace subscriptions](#).
- For sellers who want to manage the products they sell, see [Controlling access to AWS Marketplace Management Portal](#).

Condition keys

The AWS Marketplace Catalog API also supports condition keys for the `StartChangeSet` action, allowing you to tune IAM policies for each change type. For example, if an IAM user has the policy attached to their user, then they can only perform `StartChangeSet` when the change type name is `ExampleChangeTypeName`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "aws-marketplace:StartChangeSet",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "catalog:ChangeType": [ "ExampleChangeTypeName" ]
        }
      }
    }
  ]
}
```

Service limits

The AWS Marketplace Catalog API has the following limits:

Request Limits

API Operations	Request Rate
ListEntities	10 per second
DescribeEntity	10 per second
StartChangeSet	5 per second
ListChangeSets	5 per second
DescribeChangeSet	10 per second
CancelChangeSet	5 per second

Account Limits

Limit	Description
Maximum number of open <code>StartChangeSet</code> requests per account	250

Request History Retention Limits

Description	Limit
Retention period for change requests. This applies after the end time of each change request.	60 days

Working with a private marketplace

You can use the Catalog API to manage a *private marketplace* for your AWS account or [organization](#).

For more information about private marketplaces, see [Private marketplaces](#) in the *AWS Marketplace Buyer Guide*.

The following topics describe how to use the Catalog API to perform actions on your private marketplace.

Topics

- [Creating a private marketplace \(p. 8\)](#)
- [Changing the branding of your private marketplace \(p. 10\)](#)
- [Enabling or disabling your private marketplace \(p. 12\)](#)
- [Enabling or disabling user requests \(p. 12\)](#)
- [Getting a list of products in your private marketplace \(p. 13\)](#)
- [Adding or removing products from your private marketplace \(p. 14\)](#)
- [Finding products \(p. 15\)](#)
- [Working with private marketplaces for AWS Organizations \(p. 15\)](#)
- [Errors in the private marketplace API \(p. 16\)](#)

Creating a private marketplace

In the AWS Marketplace Catalog API, three entities represent a private marketplace. At the highest level is an `Experience` entity, which contains two child entities. A `ProcurementPolicy` represents the products that have been allowed and denied in your private marketplace, You can also create a `BrandingSettings` entity to define how your private marketplace looks to your users.

At a minimum, creating a private marketplace requires creating the `Experience` and the `ProcurementPolicy` entities.

Note

If your account is part of an organization in AWS Organizations see [Working with private marketplaces for AWS Organizations \(p. 15\)](#).

To create the `Experience` entity, use the `StartChangeSet` action with the `CreateExperience` value for the `ChangeType` parameter to request that the experience be created by AWS Marketplace. See the following example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateExperience",
      "Details": "{\"Name\": \"ExamplePrivateMarketplace\"}"
    }
  ]
}
```

```
    "Entity": {
      "Type": "Experience@1.0"
    }
  ],
  "ChangeSetName": "Create Private Marketplace Example"
}
```

In this action, `Entity` is a template for the entity that you want to create. It is assigned an `EntityId` when it is created. `ChangeSetName` identifies the change to help you find it later.

The response looks like the following.

```
{
  "ChangeSetArn": "arn:...:AWSMarketplace/ChangeSet/abcd1234example5678frjzkz",
  "ChangeSetId": "abcd1234example5678frjzkz"
}
```

The response includes a `ChangeSetId` that you can use to get the status of your change request as it is processed with `DescribeChangeSet`. You can also use `ListEntities` to find your `Experience` entity without the `ChangeSetId`. For more information about change sets, see [Working with change sets \(p. 3\)](#).

A new `Experience` entity doesn't have a procurement policy that controls what products are available in the private marketplace. It is also created without any settings for branding, which means that it has the defaults. See [Changing the branding of your private marketplace \(p. 10\)](#) for more information about branding settings.

You must create a `ProcurementPolicy` entity. A new `Experience` entity defaults to disabled, so you can create the procurement policy before enabling it.

Note

An `Experience` entity with *no* procurement policy (null) allows all products to be seen in your private marketplace. An `Experience` entity with an *empty* procurement policy has no products available to users.

To allow and deny products in your private marketplace, you must create the procurement policy. To do this, you again call `StartChangeSet`, but this time with the `ChangeType` of `CreateProcurementPolicy`. This example creates an empty procurement policy.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "CreateProcurementPolicy",
      "Details": "{\"Name\": \"ExampleProcurementPolicy\"}",
      "Entity": {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example@1"
      }
    }
  ]
}
```

The `Entity` you provide in this action is the `Experience` entity that you want the procurement policy created within, so you must include the identifier for the entity that you created earlier. You can use `ListEntities` to find the `Experience` entity. `DescribeChangeSet` with the change set identifier from the `CreateExperience` action will also return the identifier for you.

Note

This example shows the identifier with a revision of 1. For more information about revisions for identifiers, see the [Identifier \(p. 2\)](#) topic

You can again use `DescribeChangeSet` on the `CreateProcurementPolicy` change to follow the processing of your request.

Note

The names you give the `Experience` and `ProcurementPolicy` objects do not appear in AWS Marketplace. The names are only for your ease of finding the entities in the API.

After you have created the procurement policy, your private marketplace will show up in the AWS Management Console (you can go to the [private marketplace page](#) to see it). After you have completed these steps, your private marketplace will be disabled, have default branding, and have an empty procurement policy. You can update the branding and add any products that you want in it, and then enable your private marketplace.

The following sections describing managing your private marketplace with the AWS Marketplace Catalog API.

Changing the branding of your private marketplace

You can customize the look of your private marketplace for your users. Without customization, your private marketplace will have the default branding settings, which are described in the list below. Aspects of branding that you can change in a private marketplace include the following:

- **Title** – The name displayed for your private marketplace. This is the same as the **Name** field in the private marketplace **Profile settings** screen. If you set the **Title** to **Example**, then the text displayed is **Example Private Marketplace**. The default is no title, which is displayed as **Private Marketplace**.
- **Information** – The paragraph displayed under the name in your private marketplace. This is the same as the **Description** field in **Profile settings**. The default is to display a general description of private marketplaces.
- **ThemeColor** – The color displayed in the banner of your private marketplace. This is a color in RGB hexadecimal format. This value is the same as the **Theme color** field in **Profile settings**. The default value is `#232F3E`.
- **LogoUrl** – This is a URL that points to an image file to be used as the logo on your private marketplace. The URL must be publicly available (for example, a signed Amazon S3 URL). The file must be either a `.png` or `.svg` file and be under 500kb. If necessary, the image file will be resized down to a maximum height of 30 pixels and a maximum width of 100 pixels. This is the same value as the **Logo Select** in **Profile Settings**. The default is to not show a logo.

To set these values, you must first create a `BrandingSettings` entity with the `CreateBrandingSettings` change type. You can then request an `UpdateBrandingSettings` change to set or change the branding. You only need to create a `BrandingSettings` object once. To create this object, call `StartChangeSet` with the `CreateBrandingSettings` change type, as shown in the following example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
```

```
    "ChangeType": "CreateBrandingSettings",
    "Details": "{ \"Name\": \"ExampleBrandingSettingsName\"}",
    "Entity": {
      "Type": "Experience@1.0",
      "Identifier" : "exp-1234example@2"
    }
  }
]
}
```

This examples modifies the Experience entity by adding the BrandingSettings object to it. The revision of the entity identifier has incremented to 2. For more information about revisions for identifiers, see the [Identifier \(p. 2\)](#) topic.

Note

You can specify all the details of the branding settings in the call to create the branding settings entity. The details facet is the same for CreateBrandingSettings and UpdateBrandingSettings.

You modify the settings by calling StartChangeSet with the UpdateBrandingSettings change type. The settings are part of the Configuration of the Details object.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateBrandingSettings",
      "Details": "{
        \"Name\": \"ExampleBrandingSettingsName\",
        \"Description\": \"Example description\",
        \"Configuration\": {
          \"Title\": \"ExampleName\",
          \"Information\": \"Example description.\",
          \"ThemeColor\" : \"#0e7f74\",
          \"LogoUrl\" : \"https://example.com/path/mylogo.png\"
        }
      }",
      "Entity": {
        "Type": "Experience@1.0",
        "Identifier" : "exp-1234example@3"
      }
    }
  ]
}
```

Note

This example has line wraps added for readability and will not work as-is. The Details attribute is a string, and should be converted from a JSON object to a string in your call to StartChangeSet. In this case, the string would be "{ \"Name\": \"ExampleBrandingSettingsName\", \"Description\": \"Example description\", \"Configuration\": { \"Title\": \"ExampleName\", \"Information\": \"Example description.\", \"ThemeColor\" : \"#0e7f74\", \"LogoUrl\" : \"https://example.com/path/mylogo.png\" } }". For more information, see [Working with the Details attribute \(p. 4\)](#).

Note

The URL for the logo is used to make a copy during the update change. After the change is complete, if you remove or change the URL at that path, it will not affect your private marketplace unless you again request UpdateBrandingSettings.

Enabling or disabling your private marketplace

When your private marketplace is enabled (and has a procurement policy), users in your account or organization can only purchase products that you have allowed. When your private marketplace is disabled, users can purchase products across the full AWS Marketplace catalog.

To enable your private marketplace, use the `StartChangeRequest` action with the `UpdateExperience` change type.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateExperience",
      "Details": "{\"Status\": \"Enabled\"}",
      "Entity": {
        "Type": "Experience@1.0",
        "Identifier": "exp-1234example@4"
      }
    }
  ]
}
```

Similarly, you can use the same action and `ChangeType`, but change the `Status` in `Details` to `Disabled` to disable your private marketplace.

Note

Disabling your private marketplace keeps your list of both allowed and denied products, as well as your customizations, such as branding. When your private marketplace is disabled, your users no longer see the private marketplace. All restrictions are removed, and your users are able to procure any products in the public AWS Marketplace.

Enabling or disabling user requests

Users in your organization can view the full public AWS Marketplace, but they can only subscribe to the products that you have allowed. By default, they can request that a product that is not in the private marketplace be added to it. These requests show up in the private marketplace administrator page (<https://aws.amazon.com/marketplace/privatemarketplace/admin/>), where you can decide whether to accept or deny the request (and whether to block further requests for the same product). You cannot see the requests, nor respond to them, using the Catalog API.

You can enable or disable the ability for users to create requests for your private marketplace. Use `StartChangeSet` with the `UpdateProcurementPolicy` change type. The ability to make requests is disabled in the following example.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "UpdateProcurementPolicy",
      "Details": "{\"Configuration\" : {\"PolicyResourceRequests\" : \"Deny\"}}",
    }
  ]
}
```



```
    "Entity": {
      "Type": "Experience@1.0",
      "Identifier" : "exp-1234example@5"
    }
  ]
}
```

To enable the change request capability for users, use `Allow` instead of `Deny` in `PolicyResourceRequests`.

To learn how to get the current status of this setting, see the next section, [Getting a list of products in your private marketplace \(p. 13\)](#).

Getting a list of products in your private marketplace

The products shown (and disallowed to be shown) in your private marketplace are part of the procurement policy in your Experience entity. To get the details about the procurement policies in your private marketplace, you first get the procurement policy identifier, and then call `DescribeEntity` with that identifier.

To get the procurement policy identifier, list the procurement policy for your account.

```
POST /ListEntities HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "ProcurementPolicy"
}
```

Following is an example response.

```
{
  "EntitySummaryList": [
    {
      "EntityArn": "arn:<...>AWSMarketplace/ProcurementPolicy/
procpolicy-123example456",
      "EntityId": "procpolicy-123example456",
      "EntityType": "ProcurementPolicy",
      "LastModifiedDate": "2020-10-01T12:00:00Z",
      "Name": "ExampleProcurementPolicy",
      "Visibility": null
    }
  ],
  "NextToken": "<token>"
}
```

You can use the returned `EntityId` to get the details.

```
GET /DescribeEntity?catalog=AWSMarketplace&entityId=procpolicy-123example456
```

This returns the full details of the policy, including both allowed and denied products. Following is an example response (new lines added for readability).

```
{
  "Details": "{\\"Name\\":\\"ExampleProcurementPolicy\\",
    \\"Statements\\":[
      {\\"Effect\\":\\"Allow\\",\\"Resources\\":[
        {\\"Type\\":\\"Product\\",\\"Ids\\":[\"example1-1234-abcd-5678-90abcdef1234\\"]},
        {\\"Type\\":\\"Product\\",\\"Ids\\":[\"example2-2345-bcde-6789-01bcdea2345\\"]}],
      {\\"Effect\\":\\"Deny\\",\\"Resources\\":[
        {\\"Type\\":\\"Product\\",\\"Ids\\":[\"example3-3456-
cdef-7890-12defabc5678\\"]}],
      \\"Configuration\\":{\\"PolicyResourceRequests\\":\\"Allow\\"}},
  "EntityArn": "arn:<...>AWSMarketplace/ProcurementPolicy/procpolicy-123example456",
  "EntityIdentifier": "procpolicy-123example456@4",
  "EntityType": "ProcurementPolicy@1.0",
  "LastModifiedDate": "2020-10-01T12:00:00Z"
}
```

In this sample, the procurement policy has two allowed products and one denied product. The policy allows user resource requests.

Adding or removing products from your private marketplace

By default, your private marketplace does not have any approved products in it. Adding and removing products are done with change requests, but each have their own change type. To add a product, use the `AllowProductProcurement` change type. To remove a product, use the `DenyProductProcurement` change type. Following is an example of using the `AllowProductProcurement` change type with the `StartChangeSet` action to add a product to your private marketplace.

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "AWSMarketplace",
  "ChangeSet": [
    {
      "ChangeType": "AllowProductProcurement",
      "Details": "{\\"Products\\":[{\\"Ids\\":[\"example-1234-abcd-5678-90abcde1234\\"],
\\Notes\\":\\"Useful products\\"}]}",
      "Entity": {
        "Identifier": "exp-1234example@6",
        "Type": "Experience@1.0"
      }
    }
  ]
}
```

You add the product to the `Experience` entity for your private marketplace by using `AllowProductProcurement`. The syntax to remove a product from your marketplace is identical, with the exception that you use the `DenyProductProcurement` `ChangeType` instead of `AllowProductProcurement`.

Note

The list of products in the `Details` of your change is an array of `Ids`, so you can add (or remove) multiple products with one call by including a list of product identifiers. The limit is 50 products in a single request.

The `Notes` field for the list of `Ids` is not required, but you can use it to record why a decision to allow or deny a set of products was made.

Finding products

By getting the details of your procurement policy, you can find the product IDs for the products that are already in your private marketplace. However, the AWS Marketplace Catalog API does not provide a way to find the product IDs for other products. There are two ways to get product IDs to use with the Catalog API service.

- **Public marketplace** – After you find a product in the public marketplace, choose **Continue to Subscribe** to see a details page about the product (it will not subscribe you to the product). The URL will include the product ID as a parameter. For example, in the URL `https://aws.amazon.com/marketplace/fulfillment?productId=ab1234cd-1234-abcd-5678-90abcdef1234&ref_=dtl_psb_continue,ab1234cd-1234-abcd-5678-90abcdef1234` is the product ID.
- **AWS Marketplace Discovery API** – Programmatically, you can access the full list of products in the AWS Marketplace by using the Discovery API. The Discovery API is a private API. You must request access to be able to use it. For more information, see [Getting access to the Discovery API](#) (p. 20).

Working with private marketplaces for AWS Organizations

If your account is part of an organization in AWS Organizations, then the only private marketplace that can exist is the private marketplace for the entire organization. That private marketplace (and its associated `Experience`, `ProcurementPolicy`, and `BrandingSettings` objects) applies to all users in the organization.

Whether you are working with a private marketplace for your account or your organization, you use the same API. However, there are a few differences when working within your organization.

First, you must create the `Experience` of your private marketplace from within the management account. After it is created, you can edit and update it from any account, as long as you have the correct permissions.

Second, when listing objects in the shared private marketplace, you must specifically request them with the `SharedWithMe` filter. This applies to both `ListEntities` and `ListChangeSets` actions.

For example, when you call `ListEntities` from an account that isn't the management account, it only returns entities created by or owned by that account, by default. To list `Experience` objects in your own account, call `ListEntities` as shown in the following example.

```
POST /ListEntities HTTP/1.1
Content-Type: application/json

{
  "Catalog": "AWSMarketplace",
  "EntityType": "Experience"
}
```

However, to list the entities that have been shared with you, you must add a `FilterList` with a `Scope` of `SharedWithMe` to tell AWS Marketplace to search outside of your own account.

```
POST /ListEntities HTTP/1.1
Content-Type: application/json

{"Catalog": "AWSMarketplace",
```

```
"EntityType": "Experience",
"FilterList":
  [{
    "Name": "Scope",
    "ValueList":
      ["SharedWithMe"]
  ]}]}
```

In this case, only entities outside of your account (the ones for your organization) are returned.

Similarly, to call `ListChangeSets`, you must set the scope.

```
POST /ListChangeSets HTTP/1.1
Content-Type: application/json

{"Catalog": "AWSMarketplace",
"FilterList":
  [{
    "Name": "Scope",
    "ValueList":
      ["SharedWithMe"]
  ]}]}
```

This returns change sets that apply to the shared private marketplace for your organization.

Errors in the private marketplace API

The following errors are specific to the private marketplace actions in the AWS Marketplace Catalog API.

Change type	Error code	Error message	Description
Errors returned directly by the StartChangeSet action			
All	422	Document not valid JSON format	Invalid JSON input used, check your syntax.
AllowProductProcurement DenyProductProcurement	422	Values in Ids array must be unique	You can't include the same product multiple times in a single change request.
AllowProductProcurement DenyProductProcurement	422	Cumulative number of values in Ids array must be less than or equal to 50	You can allow or deny up to 50 products in a single change request.
CreateExperience	422	Entity of type 'Experience' with Id: '[Id]' already exists.	You can only have a single Experience entity for a private marketplace.
Errors found by calling the DescribeChangeSet action			
CreateBrandingSettings, UpdateBrandingSettings	INVALID_URL	Image could not be fetched from the input URL	You must specify a valid, reachable URL for the logo field in BrandingSettings.

Change type	Error code	Error message	Description
CreateBrandingSettings, UpdateBrandingSettings	INVALID_IMAGE	Image verification for type, content, or file size failed. Only .png and .svg file types with sizes less than or equal to 500KB are supported.	Your image file must match the logo requirements for branding settings.
AllowProductProcurement DenyProductProcurement	ENTITY_NOT_FOUND	Procurement policy missing from Experience	You must create a <code>ProcurementPolicy</code> before allowing or denying products.
CreateProcurementPolicy	ENTITY_ALREADY_EXISTS	Procurement policy exists for Experience	You can only have a single procurement policy for a private marketplace.
UpdateProcurementPolicy	ENTITY_NOT_FOUND	Procurement policy missing from Experience	You must create a <code>ProcurementPolicy</code> before updating the procurement policy.
CreateBrandingSettings	ENTITY_ALREADY_EXISTS	Branding settings exists for Experience	You can only have a single branding settings for a private marketplace.
UpdateBrandingSettings	ENTITY_NOT_FOUND	Branding settings missing from Experience	You must create a <code>BrandingSettings</code> entity before updating the branding settings.

Logging AWS Marketplace Catalog API calls with CloudTrail

The AWS Marketplace Catalog API is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures all calls to the Catalog API as events, including calls from the AWS Marketplace Management Portal.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request, the IP address from which the request was made, who made the request, when it was made, and additional details.

AWS Marketplace Catalog API information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in the AWS Marketplace Catalog API, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#) in the *AWS CloudTrail User Guide*.

For an ongoing record of events in your AWS account, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#)
- [Receiving CloudTrail Log Files from Multiple Accounts](#)

All AWS Marketplace Catalog API actions are logged by CloudTrail and are documented in this API Reference. For example, calls to the `StartChangeSet`, `DescribeChangeSet`, and `ListChangeSets` API actions generate entries in the CloudTrail log files. Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see [CloudTrail userIdentity Element](#) in the *AWS CloudTrail User Guide*.

Understanding AWS Marketplace catalog log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

Note

These examples have been formatted for improved readability. In a CloudTrail log file, all entries and events are concatenated into a single line. In addition, this example has been limited to a single AWS Marketplace Catalog API entry. In a real CloudTrail log file, you see entries and events from multiple AWS services.

The following example shows a AWS Marketplace Catalog API log entry that demonstrates the `ListEntities` action:

```
[
  {
    "eventVersion": "1.05",
    "userIdentity": {
      "type": "IAMUser",
      "principalId": "ABCDEF_GHIJKLMNOP12345",
      "arn": "arn:aws:iam::123456789010:user/CloudTrailTestUser",
      "accountId": "123456789010",
      "accessKeyId": "ABCDEFGHIJKLMN0P1234",
      "userName": "CloudTrailTestUser"
    },
    "eventTime": "2019-10-17T21:49:23Z",
    "eventSource": "marketplacecatalog.amazonaws.com",
    "eventName": "ListEntities",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "127.0.0.1",
    "userAgent": "PostmanRuntime/7.18.0",
    "requestParameters": {
      "catalog": "AWSMarketplace",
      "entityType": "EntityProduct",
      "sort": {
        "sortBy": "LastUpdateTimeInMillis",
        "sortOrder": "DESC"
      },
      "maxResults": 20
    },
    "responseElements": null,
    "requestID": "fEXAMPLE-cb3e-4e21-86fd-6b3EXAMPLEd1",
    "eventID": "7EXAMPLE-97d6-4139-91e3-01aEXAMPLE48",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "recipientAccountId": "123456789010"
  }
]
```

Discovery API

Using the AWS Marketplace Catalog API service gives you access to manage the products that you create as a seller in AWS Marketplace, and to manage your private marketplace. However, you must use the AWS Marketplace Discovery API in the following scenarios:

- Get product information from AWS Marketplace
- Populate your private marketplace with products
- Create a custom view of AWS Marketplace for your customers where you show offerings from other sellers and offer value-added functionality

With the Discovery API, you can create browse and search functionality for the full AWS Marketplace catalog of products, with links to AWS Marketplace so customers can buy products. Or, you can access the full list of AWS Marketplace products to help populate your private marketplace.

Note

The AWS Marketplace Discovery API is a private API. You must request access to be able to use it. For more information, see the next section, [Getting access to the Discovery API \(p. 20\)](#).

Getting access to the Discovery API

The AWS Marketplace Discovery API is a private API, only available to select customers. Calls to the Discovery API require an Integration ID that is provided to you when you are approved for access.

To request access to the Discovery API, or get answers to other questions about the Discovery API, reach out to your existing AWS Marketplace contact. If you don't have a contact, or don't know who your contact is, you can reach out to the [AWS Marketplace Seller Operations team](#).

After you have been approved for access, you will receive your Integration ID, as well as documentation for how to use the Discovery API.

Actions

The following actions are supported:

- [CancelChangeSet \(p. 22\)](#)
- [DescribeChangeSet \(p. 25\)](#)
- [DescribeEntity \(p. 29\)](#)
- [ListChangeSets \(p. 32\)](#)
- [ListEntities \(p. 35\)](#)
- [StartChangeSet \(p. 39\)](#)

CancelChangeSet

Used to cancel an open change request. Must be sent before the status of the request changes to `APPLYING`, the final stage of completing your change request. You can describe a change during the 60-day request history retention period for API calls.

Request Syntax

```
PATCH /CancelChangeSet?catalog=Catalog&changeSetId=ChangeSetId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

Catalog (p. 22)

Required. The catalog related to the request. Fixed value: `AWSMarketplace`.

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

ChangeSetId (p. 22)

Required. The unique identifier of the `StartChangeSet` request that you want to cancel.

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

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

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ChangeSetArn": "string",
  "ChangeSetId": "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.

[ChangeSetArn \(p. 22\)](#)

The ARN associated with the change set referenced in this request.

Type: String

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

Pattern: `^[a-zA-Z0-9:*/-]+$`

[ChangeSetId \(p. 22\)](#)

The unique identifier for the change set referenced in this request.

Type: String

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

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

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 56\)](#).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceInUseException

The resource is currently in use.

HTTP Status Code: 423

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

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](#)

DescribeChangeSet

Provides information about a given change set.

Request Syntax

```
GET /DescribeChangeSet?catalog=Catalog&changeSetId=ChangeSetId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

Catalog (p. 25)

Required. The catalog related to the request. Fixed value: `AWSMarketplace`

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

ChangeSetId (p. 25)

Required. The unique identifier for the `StartChangeSet` request that you want to describe the details for.

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

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

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ChangeSet": [
    {
      "ChangeType": "string",
      "Details": "string",
      "Entity": {
        "Identifier": "string",
        "Type": "string"
      },
      "ErrorDetailList": [
        {
          "ErrorCode": "string",
          "ErrorMessage": "string"
        }
      ]
    }
  ]
}
```

```
    ]
  }
],
"ChangeSetArn": "string",
"ChangeSetId": "string",
"ChangeSetName": "string",
"EndTime": "string",
"FailureCode": "string",
"FailureDescription": "string",
"StartTime": "string",
"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.

ChangeSet (p. 25)

An array of ChangeSummary objects.

Type: Array of [ChangeSummary \(p. 47\)](#) objects

ChangeSetArn (p. 25)

The ARN associated with the unique identifier for the change set referenced in this request.

Type: String

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

Pattern: `^[a-zA-Z0-9:*/-]+$`

ChangeSetId (p. 25)

Required. The unique identifier for the change set referenced in this request.

Type: String

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

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

ChangeSetName (p. 25)

The optional name provided in the `StartChangeSet` request. If you do not provide a name, one is set by default.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[\\w\\s+=.:@-]+$`

EndTime (p. 25)

The date and time, in ISO 8601 format (2018-02-27T13:45:22Z), the request transitioned to a terminal state. The change cannot transition to a different state. Null if the request is not in a terminal state.

Type: String

Length Constraints: Fixed length of 20.

Pattern: `^([\d]{4})\-(1[0-2]|0[1-9])\-(3[01]|0[1-9]|12)[\d]T(2[0-3]|01)[\d]:([0-5][\d]):([0-5][\d])Z$`

FailureCode (p. 25)

Returned if the change set is in `FAILED` status. Can be either `CLIENT_ERROR`, which means that there are issues with the request (see the `ErrorDetailList`), or `SERVER_FAULT`, which means that there is a problem in the system, and you should retry your request.

Type: String

Valid Values: `CLIENT_ERROR` | `SERVER_FAULT`

FailureDescription (p. 25)

Returned if there is a failure on the change set, but that failure is not related to any of the changes in the request.

Type: String

StartTime (p. 25)

The date and time, in ISO 8601 format (2018-02-27T13:45:22Z), the request started.

Type: String

Length Constraints: Fixed length of 20.

Pattern: `^([\d]{4})\-(1[0-2]|0[1-9])\-(3[01]|0[1-9]|12)[\d]T(2[0-3]|01)[\d]:([0-5][\d]):([0-5][\d])Z$`

Status (p. 25)

The status of the change request.

Type: String

Valid Values: `PREPARING` | `APPLYING` | `SUCCEEDED` | `CANCELLED` | `FAILED`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 56\)](#).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

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](#)

DescribeEntity

Returns the metadata and content of the entity.

Request Syntax

```
GET /DescribeEntity?catalog=Catalog&entityId=EntityId HTTP/1.1
```

URI Request Parameters

The request uses the following URI parameters.

Catalog (p. 29)

Required. The catalog related to the request. Fixed value: `AWSMarketplace`

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

EntityId (p. 29)

Required. The unique ID of the entity to describe.

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

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

Required: Yes

Request Body

The request does not have a request body.

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "Details": "string",
  "EntityArn": "string",
  "EntityIdentifier": "string",
  "EntityType": "string",
  "LastModifiedDate": "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.

Details (p. 29)

This stringified JSON object includes the details of the entity.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 16384.

Pattern: `^\s*\{[\s\S]*\}\s*$`

EntityArn (p. 29)

The ARN associated to the unique identifier for the change set referenced in this request.

Type: String

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

Pattern: `^[a-zA-Z0-9:*/-]+$`

EntityIdentifier (p. 29)

The identifier of the entity, in the format of `EntityId@RevisionId`.

Type: String

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

Pattern: `^[w\-\@]+$`

EntityType (p. 29)

The named type of the entity, in the format of `EntityType@Version`.

Type: String

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

Pattern: `^[a-zA-Z]+$`

LastModifiedDate (p. 29)

The last modified date of the entity, in ISO 8601 format (2018-02-27T13:45:22Z).

Type: String

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 56\)](#).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ResourceNotSupportedException

Currently, the specified resource is not supported.

HTTP Status Code: 415

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

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](#)

ListChangeSets

Returns the list of change sets owned by the account being used to make the call. You can filter this list by providing any combination of `entityId`, `ChangeSetName`, and `status`. If you provide more than one filter, the API operation applies a logical AND between the filters.

You can describe a change during the 60-day request history retention period for API calls.

Request Syntax

```
POST /ListChangeSets HTTP/1.1
Content-type: application/json

{
  "Catalog": "string",
  "FilterList": [
    {
      "Name": "string",
      "ValueList": [ "string" ]
    }
  ],
  "MaxResults": number,
  "NextToken": "string",
  "Sort": {
    "SortBy": "string",
    "SortOrder": "string"
  }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

Catalog (p. 32)

The catalog related to the request. Fixed value: `AWSMarketplace`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

FilterList (p. 32)

An array of filter objects.

Type: Array of [Filter \(p. 52\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 8 items.

Required: No

MaxResults (p. 32)

The maximum number of results returned by a single call. This value must be provided in the next call to retrieve the next set of results. By default, this value is 20.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 20.

Required: No

NextToken (p. 32)

The token value retrieved from a previous call to access the next page of results.

Type: String

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

Pattern: `^[\w+=. :@\-\/]$`

Required: No

Sort (p. 32)

An object that contains two attributes, `SortBy` and `SortOrder`.

Type: [Sort \(p. 53\)](#) object

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ChangeSetSummaryList": [
    {
      "ChangeSetArn": "string",
      "ChangeSetId": "string",
      "ChangeSetName": "string",
      "EndTime": "string",
      "EntityIdList": [ "string" ],
      "FailureCode": "string",
      "StartTime": "string",
      "Status": "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.

ChangeSetSummaryList (p. 33)

Array of `ChangeSetSummaryListItem` objects.

Type: Array of [ChangeSetSummaryListItem \(p. 45\)](#) objects

NextToken (p. 33)

The value of the next token, if it exists. Null if there are no more results.

Type: String

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

Pattern: `^\w+\. :@-\[/]$`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 56\)](#).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

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](#)

ListEntities

Provides the list of entities of a given type.

Request Syntax

```
POST /ListEntities HTTP/1.1
Content-type: application/json

{
  "Catalog": "string",
  "EntityType": "string",
  "FilterList": [
    {
      "Name": "string",
      "ValueList": [ "string" ]
    }
  ],
  "MaxResults": number,
  "NextToken": "string",
  "Sort": {
    "SortBy": "string",
    "SortOrder": "string"
  }
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

Catalog (p. 35)

The catalog related to the request. Fixed value: `AWSMarketplace`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

EntityType (p. 35)

The type of entities to retrieve.

Type: String

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

Pattern: `^[a-zA-Z]+$`

Required: Yes

FilterList (p. 35)

An array of filter objects. Each filter object contains two attributes, `filterName` and `filterValues`.

Type: Array of [Filter \(p. 52\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 8 items.

Required: No

MaxResults (p. 35)

Specifies the upper limit of the elements on a single page. If a value isn't provided, the default value is 20.

Type: Integer

Valid Range: Minimum value of 1. Maximum value of 20.

Required: No

NextToken (p. 35)

The value of the next token, if it exists. Null if there are no more results.

Type: String

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

Pattern: `^[\w+=. :@-\/]$`

Required: No

Sort (p. 35)

An object that contains two attributes, `SortBy` and `SortOrder`.

Type: [Sort \(p. 53\)](#) object

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "EntitySummaryList": [
    {
      "EntityArn": "string",
      "EntityId": "string",
      "EntityType": "string",
      "LastModifiedDate": "string",
      "Name": "string",
      "Visibility": "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.

EntitySummaryList (p. 36)

Array of `EntitySummary` object.

Type: Array of `EntitySummary (p. 49)` objects

NextToken (p. 36)

The value of the next token if it exists. Null if there is no more result.

Type: String

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

Pattern: `^[\w+=.:@\-\\/]#$`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 56\)](#).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

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](#)

StartChangeSet

This operation allows you to request changes for your entities. Within a single ChangeSet, you cannot start the same change type against the same entity multiple times. Additionally, when a ChangeSet is running, all the entities targeted by the different changes are locked until the ChangeSet has completed (either succeeded, cancelled, or failed). If you try to start a ChangeSet containing a change against an entity that is already locked, you will receive a `ResourceInUseException`.

For example, you cannot start the ChangeSet described in the [example](#) later in this topic, because it contains two changes to execute the same change type (`AddRevisions`) against the same entity (`entity-id@1`).

For more information about working with change sets, see [Working with change sets](#).

Request Syntax

```
POST /StartChangeSet HTTP/1.1
Content-type: application/json

{
  "Catalog": "string",
  "ChangeSet": [
    {
      "ChangeType": "string",
      "Details": "string",
      "Entity": {
        "Identifier": "string",
        "Type": "string"
      }
    }
  ],
  "ChangeSetName": "string",
  "ClientRequestToken": "string"
}
```

URI Request Parameters

The request does not use any URI parameters.

Request Body

The request accepts the following data in JSON format.

Catalog (p. 39)

The catalog related to the request. Fixed value: `AWSMarketplace`

Type: String

Length Constraints: Minimum length of 1. Maximum length of 64.

Pattern: `^[a-zA-Z]+$`

Required: Yes

ChangeSet (p. 39)

Array of change object.

Type: Array of [Change \(p. 44\)](#) objects

Array Members: Minimum number of 1 item. Maximum number of 20 items.

Required: Yes

[ChangeSetName \(p. 39\)](#)

Optional case sensitive string of up to 100 ASCII characters. The change set name can be used to filter the list of change sets.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[\w\s+=. :@-]+$`

Required: No

[ClientRequestToken \(p. 39\)](#)

A unique token to identify the request to ensure idempotency.

Type: String

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

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

Required: No

Response Syntax

```
HTTP/1.1 200
Content-type: application/json

{
  "ChangeSetArn": "string",
  "ChangeSetId": "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.

[ChangeSetArn \(p. 40\)](#)

The ARN associated to the unique identifier generated for the request.

Type: String

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

Pattern: `^[a-zA-Z0-9:*/-]+$`

[ChangeSetId \(p. 40\)](#)

Unique identifier generated for the request.

Type: String

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

Pattern: `^\w[-]+`

Errors

For information about the errors that are common to all actions, see [Common Errors \(p. 56\)](#).

AccessDeniedException

Access is denied.

HTTP Status Code: 403

InternalServiceException

There was an internal service exception.

HTTP Status Code: 500

ResourceInUseException

The resource is currently in use.

HTTP Status Code: 423

ResourceNotFoundException

The specified resource wasn't found.

HTTP Status Code: 404

ServiceQuotaExceededException

The maximum number of open requests per account has been exceeded.

HTTP Status Code: 402

ThrottlingException

Too many requests.

HTTP Status Code: 429

ValidationException

An error occurred during validation.

HTTP Status Code: 422

Examples

Example

You cannot start this ChangeSet because it contains two changes to execute the same change type (AddRevisions) against the same entity (entity-id@1).

```
{  
  "Catalog": "AWSMarketplace",
```

```
"ChangeSetName": "Adding revisions to my test Data Product",
"ChangeSet": [
  {
    "ChangeType": "AddRevisions",
    "Entity": {
      "Identifier": "entity-id@1",
      "Type": "DataProduct@1.0"
    },
    "Details": "{\"DataSetArn\": \"data-set-arn\", \"RevisionArns\": [\"revision-arn\", \"revision-arn-2\"] }"
  },
  {
    "ChangeType": "AddRevisions",
    "Entity": {
      "Identifier": "entity-id@1",
      "Type": "DataProduct@1.0"
    },
    "Details": "{\"DataSetArn\": \"data-set-arn\", \"RevisionArns\": [\"revision-arn3\"] }"
  }
]
}
```

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 AWS Marketplace Catalog Service 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:

- [Change](#) (p. 44)
- [ChangeSetSummaryListItem](#) (p. 45)
- [ChangeSummary](#) (p. 47)
- [Entity](#) (p. 48)
- [EntitySummary](#) (p. 49)
- [ErrorDetail](#) (p. 51)
- [Filter](#) (p. 52)
- [Sort](#) (p. 53)

Change

An object that contains the `ChangeType`, `Details`, and `Entity`.

Contents

ChangeType

Change types are single string values that describe your intention for the change. Each change type is unique for each `EntityType` provided in the change's scope.

Type: String

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

Pattern: `^[A-Z][\w]*$`

Required: Yes

Details

This object contains details specific to the change type of the requested change.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 16384.

Pattern: `^\s*\{[\s\S]*\}\s*$`

Required: Yes

Entity

The entity to be changed.

Type: [Entity \(p. 48\)](#) 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 Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for Ruby V3](#)

ChangeSetSummaryListItem

A summary of a change set returned in a list of change sets when the `ListChangeSets` action is called.

Contents

ChangeSetArn

The ARN associated with the unique identifier for the change set referenced in this request.

Type: String

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

Pattern: `^[a-zA-Z0-9:*/-]+$`

Required: No

ChangeSetId

The unique identifier for a change set.

Type: String

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

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

Required: No

ChangeSetName

The non-unique name for the change set.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 100.

Pattern: `^[\\w\\s+=.:@-]+$`

Required: No

EndTime

The time, in ISO 8601 format (2018-02-27T13:45:22Z), when the change set was finished.

Type: String

Length Constraints: Fixed length of 20.

Pattern: `^([\\d]{4})\\-(1[0-2]|0[1-9])\\-(3[01]|0[1-9]|12)[\\d])T(2[0-3]|[01][\\d]):([0-5][\\d]):([0-5][\\d])Z$`

Required: No

EntityIdList

This object is a list of entity IDs (string) that are a part of a change set. The entity ID list is a maximum of 20 entities. It must contain at least one entity.

Type: Array of strings

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

Pattern: `^\w\-$`

Required: No

FailureCode

Returned if the change set is in `FAILED` status. Can be either `CLIENT_ERROR`, which means that there are issues with the request (see the `ErrorDetailList` of `DescribeChangeSet`), or `SERVER_FAULT`, which means that there is a problem in the system, and you should retry your request.

Type: String

Valid Values: `CLIENT_ERROR` | `SERVER_FAULT`

Required: No

StartTime

The time, in ISO 8601 format (2018-02-27T13:45:22Z), when the change set was started.

Type: String

Length Constraints: Fixed length of 20.

Pattern: `^([\d]{4})\-(1[0-2]|0[1-9])\-(3[01]|0[1-9]|12)[\d]T(2[0-3]|01)[\d]:([0-5][\d]):([0-5][\d])Z$`

Required: No

Status

The current status of the change set.

Type: String

Valid Values: `PREPARING` | `APPLYING` | `SUCCEEDED` | `CANCELLED` | `FAILED`

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](#)

ChangeSummary

This object is a container for common summary information about the change. The summary doesn't contain the whole change structure.

Contents

ChangeType

The type of the change.

Type: String

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

Pattern: `^[A-Z][\w]*$`

Required: No

Details

This object contains details specific to the change type of the requested change.

Type: String

Length Constraints: Minimum length of 2. Maximum length of 16384.

Pattern: `^\s*\{[\s\S]*\}\s*$`

Required: No

Entity

The entity to be changed.

Type: [Entity \(p. 48\)](#) object

Required: No

ErrorDetailList

An array of `ErrorDetail` objects associated with the change.

Type: Array of [ErrorDetail \(p. 51\)](#) 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](#)

Entity

An entity contains data that describes your product, its supported features, and how it can be used or launched by your customer.

Contents

Identifier

The identifier for the entity.

Type: String

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

Pattern: `^[\w\ -@]+$`

Required: No

Type

The type of entity.

Type: String

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

Pattern: `^[a-zA-Z]+$`

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](#)

EntitySummary

This object is a container for common summary information about the entity. The summary doesn't contain the whole entity structure, but it does contain information common across all entities.

Contents

EntityArn

The ARN associated with the unique identifier for the entity.

Type: String

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

Pattern: `^[a-zA-Z0-9:*/-]+$`

Required: No

EntityId

The unique identifier for the entity.

Type: String

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

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

Required: No

EntityType

The type of the entity.

Type: String

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

Pattern: `^[a-zA-Z]+$`

Required: No

LastModifiedDate

The last time the entity was published, using ISO 8601 format (2018-02-27T13:45:22Z).

Type: String

Required: No

Name

The name for the entity. This value is not unique. It is defined by the seller.

Type: String

Required: No

Visibility

The visibility status of the entity to buyers. This value can be `Public` (everyone can view the entity), `Limited` (the entity is visible to limited accounts only), or `Restricted` (the entity was published and then unpublished and only existing buyers can view it).

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

ErrorDetail

Details about the error.

Contents

ErrorCode

The error code that identifies the type of error.

Type: String

Required: No

ErrorMessage

The message for the error.

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

Filter

A filter object, used to optionally filter results from calls to the `ListEntities` and `ListChangeSets` actions.

Contents

Name

For `ListEntities`, the supported value for this is an `EntityId`.

For `ListChangeSets`, the supported values are as follows:

Type: String

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

Pattern: `^[a-zA-Z]+$`

Required: No

ValueList

`ListEntities` - This is a list of unique `EntityIds`.

`ListChangeSets` - The supported filter names and associated `ValueLists` is as follows:

- `ChangeSetName` - The supported `ValueList` is a list of non-unique `ChangeSetNames`. These are defined when you call the `StartChangeSet` action.
- `Status` - The supported `ValueList` is a list of statuses for all change set requests.
- `EntityId` - The supported `ValueList` is a list of unique `EntityIds`.
- `BeforeStartTime` - The supported `ValueList` is a list of all change sets that started before the filter value.
- `AfterStartTime` - The supported `ValueList` is a list of all change sets that started after the filter value.
- `BeforeEndTime` - The supported `ValueList` is a list of all change sets that ended before the filter value.
- `AfterEndTime` - The supported `ValueList` is a list of all change sets that ended after the filter value.

Type: Array of strings

Array Members: Minimum number of 1 item. Maximum number of 10 items.

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](#)

Sort

An object that contains two attributes, `SortBy` and `SortOrder`.

Contents

SortBy

For `ListEntities`, supported attributes include `LastModifiedDate` (default), `Visibility`, `EntityId`, and `Name`.

For `ListChangeSets`, supported attributes include `StartTime` and `EndTime`.

Type: String

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

Pattern: `^[a-zA-Z]+$`

Required: No

SortOrder

The sorting order. Can be `ASCENDING` or `DESCENDING`. The default value is `DESCENDING`.

Type: String

Valid Values: `ASCENDING` | `DESCENDING`

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](#)

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

Document history

The following table describes the documentation for this release of the *AWS Marketplace Catalog API Reference*.

update-history-change	update-history-description	update-history-date
Support for managing private marketplaces (p. 58)	Added the ability to create and maintain private marketplaces for AWS Organizations programmatically. See Working with a private marketplace (p. 8) .	December 3, 2020
The AWS Marketplace Discovery API is now available (p. 58)	The Discovery API provides programmatic access to find products in the AWS Marketplace. For details, see Discovery API (p. 20)	September 30, 2020
The AWS Marketplace Catalog API is now generally available (p. 58)	This service provides an API interface for approved providers to programmatically access the self-service publishing capabilities on the AWS Marketplace Management Portal.	November 12, 2019