AWS Marketplace Entitlement Service

API Reference

API Version 2017-01-11

AWS Marketplace Entitlement Service: API Reference

Table of Contents

Welcome

This reference provides descriptions of the AWS Marketplace Entitlement Service API.

AWS Marketplace Entitlement Service is used to determine the entitlement of a customer to a given product. An entitlement represents capacity in a product owned by the customer. For example, a customer might own some number of users or seats in an SaaS application or some amount of data capacity in a multi-tenant database.

Getting Entitlement Records

- *GetEntitlements*- Gets the entitlements for a Marketplace product.

This document was last published on September 24, 2021.

Actions

The following actions are supported:

- GetEntitlements (p. 3)

GetEntitlements

GetEntitlements retrieves entitlement values for a given product. The results can be filtered based on customer identifier or product dimensions.

Request Syntax

```
{
   "Filter": { 
      "string" : [ "string" ]
   },
   "MaxResults": number,
   "NextToken": "string",
   "ProductCode": "string"
}
```

Request Parameters

The request accepts the following data in JSON format.

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

ProductCode (p. 3)

Product code is used to uniquely identify a product in AWS Marketplace. The product code will be provided by AWS Marketplace when the product listing is created.

Type: String

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

Required: Yes

Filter (p. 3)

Filter is used to return entitlements for a specific customer or for a specific dimension. Filters are described as keys mapped to a lists of values. Filtered requests are *unioned* for each value in the value list, and then *intersected* for each filter key.

Type: String to array of strings map

Valid Keys: `CUSTOMER_IDENTIFIER | DIMENSION`

Array Members: Minimum number of 1 item.

Required: No

MaxResults (p. 3)

The maximum number of items to retrieve from the GetEntitlements operation. For pagination, use the NextToken field in subsequent calls to GetEntitlements.

Type: Integer

Required: No

NextToken (p. 3)

For paginated calls to GetEntitlements, pass the NextToken from the previous GetEntitlementsResult.

Type: String

Pattern: `\S+`

Required: No

Response Syntax

```
{
   "Entitlements": [
      {
         "CustomerIdentifier": "string",
         "Dimension": "string",
         "ExpirationDate": number,
         "ProductCode": "string",
         "Value": {
            "BooleanValue": boolean,
            "DoubleValue": number,
            "IntegerValue": number,
            "StringValue": "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.

Entitlements (p. 4)

The set of entitlements found through the GetEntitlements operation. If the result contains an empty set of entitlements, NextToken might still be present and should be used.

Type: Array of Entitlement (p. 7) objects

Array Members: Minimum number of 0 items.

NextToken (p. 4)

For paginated results, use NextToken in subsequent calls to GetEntitlements. If the result contains an empty set of entitlements, NextToken might still be present and should be used.

Type: String

Pattern: `\S+`

Errors

For information about the errors that are common to all actions, see Common Errors (p. 10).

InternalServiceErrorException

An internal error has occurred. Retry your request. If the problem persists, post a message with details on the AWS forums.

HTTP Status Code: 500

InvalidParameterException

One or more parameters in your request was invalid.

HTTP Status Code: 400

ThrottlingException

The calls to the GetEntitlements API are throttled.

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 AWSMarketplace entitlement 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:

- Entitlement (p. 7)
- EntitlementValue (p. 9)

Entitlement

An entitlement represents capacity in a product owned by the customer. For example, a customer might own some number of users or seats in an SaaS application or some amount of data capacity in a multi-tenant database.

Contents

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

CustomerIdentifier

The customer identifier is a handle to each unique customer in an application. Customer identifiers are obtained through the ResolveCustomer operation in AWS Marketplace Metering Service.

Type: String

Pattern: `\S+`

Required: No

Dimension

The dimension for which the given entitlement applies. Dimensions represent categories of capacity in a product and are specified when the product is listed in AWS Marketplace.

Type: String

Pattern: `\S+`

Required: No

ExpirationDate

The expiration date represents the minimum date through which this entitlement is expected to remain valid. For contractual products listed on AWS Marketplace, the expiration date is the date at which the customer will renew or cancel their contract. Customers who are opting to renew their contract will still have entitlements with an expiration date.

Type: Timestamp

Required: No

ProductCode

The product code for which the given entitlement applies. Product codes are provided by AWS Marketplace when the product listing is created.

Type: String

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

Required: No

Value

The EntitlementValue represents the amount of capacity that the customer is entitled to for the product.

Type: EntitlementValue (p. 9) object

Required: No

See Also

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

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

EntitlementValue

The EntitlementValue represents the amount of capacity that the customer is entitled to for the product.

Contents

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

BooleanValue

The BooleanValue field will be populated with a boolean value when the entitlement is a boolean type. Otherwise, the field will not be set.

Type: Boolean

Required: No

DoubleValue

The DoubleValue field will be populated with a double value when the entitlement is a double type. Otherwise, the field will not be set.

Type: Double

Required: No

IntegerValue

The IntegerValue field will be populated with an integer value when the entitlement is an integer type. Otherwise, the field will not be set.

Type: Integer

Required: No

StringValue

The StringValue field will be populated with a string value when the entitlement is a string type. Otherwise, the field will not be set.

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

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