



API Reference

# AWS Marketplace Metering Service



**API Version 2016-01-14**

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Marketplace Metering Service: API Reference

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

---

# Table of Contents

<b>Welcome .....</b>	<b>1</b>
<b>Actions .....</b>	<b>4</b>
BatchMeterUsage .....	5
Request Syntax .....	6
Request Parameters .....	6
Response Syntax .....	7
Response Elements .....	8
Errors .....	9
See Also .....	10
MeterUsage .....	12
Request Syntax .....	12
Request Parameters .....	13
Response Syntax .....	14
Response Elements .....	15
Errors .....	15
See Also .....	16
RegisterUsage .....	18
Request Syntax .....	19
Request Parameters .....	19
Response Syntax .....	20
Response Elements .....	20
Errors .....	21
Examples .....	22
See Also .....	23
ResolveCustomer .....	24
Request Syntax .....	24
Request Parameters .....	25
Response Syntax .....	25
Response Elements .....	25
Errors .....	26
See Also .....	27
<b>Data Types .....</b>	<b>28</b>
Tag .....	29
Contents .....	29

See Also .....	29
UsageAllocation .....	31
Contents .....	31
See Also .....	31
UsageRecord .....	33
Contents .....	33
See Also .....	34
UsageRecordResult .....	35
Contents .....	35
See Also .....	36
<b>Common Errors .....</b>	<b>37</b>

# Welcome

This reference provides descriptions of the low-level AWS Marketplace Metering Service API.

AWS Marketplace sellers can use this API to submit usage data for custom usage dimensions.

For information about the permissions that you need to use this API, see [AWS Marketplace metering and entitlement API permissions](#) in the *AWS Marketplace Seller Guide*.

## Submitting metering records

### *MeterUsage*

- Submits the metering record for an AWS Marketplace product.
- Called from: Amazon Elastic Compute Cloud (Amazon EC2) instance or a container running on either Amazon Elastic Kubernetes Service (Amazon EKS) or Amazon Elastic Container Service (Amazon ECS)
- Supported product types: Amazon Machine Images (AMIs) and containers
- Vendor-metered tagging: Supported allocation tagging

### *BatchMeterUsage*

- Submits the metering record for a set of customers. BatchMeterUsage API calls are captured by AWS CloudTrail. You can use CloudTrail to verify that the software as a subscription (SaaS) metering records that you sent are accurate by searching for records with the eventName of BatchMeterUsage. You can also use CloudTrail to audit records over time. For more information, see the [AWS CloudTrail User Guide](#).
- Called from: SaaS applications
- Supported product type: SaaS
- Vendor-metered tagging: Supports allocation tagging

## Accepting new customers

### *ResolveCustomer*

- Resolves the registration token that the buyer submits through the browser during the registration process. Obtains a `CustomerIdentifier` along with the `CustomerAWSAccountId` and `ProductCode`.
- Called from: SaaS application during the registration process
- Supported product type: SaaS
- Vendor-metered tagging: Not applicable

## Entitlement and metering for paid container products

### *RegisteredUsage*

- Provides software entitlement and metering. Paid container software products sold through AWS Marketplace must integrate with the AWS Marketplace Metering Service and call the `RegisterUsage` operation. Free and Bring Your Own License model (BYOL) products for Amazon ECS or Amazon EKS aren't required to call `RegisterUsage`. However, you can do so if you want to receive usage data in your seller reports. For more information about using the `RegisterUsage` operation, see [Container-based products](#).
- Called from: Paid container software products
- Supported product type: Containers
- Vendor-metered tagging: Not applicable

## Entitlement custom metering for container products

- Metering:`MeterUsage` API is available in GovCloud Regions but only supports AMI FCP products in GovCloud Regions. Flexible Consumption Pricing (FCP) Container products aren't supported in GovCloud Regions: us-gov-west-1 and us-gov-east-1. For more information, see [Container-based products](#).
- Custom metering for container products are called using the Metering:`MeterUsage` API. The API is used for FCP AMI and FCP Container product metering.

## Custom metering for Amazon EKS is available in 17 AWS Regions

- The metering service supports Amazon ECS and EKS for Flexible Consumption Pricing (FCP) products using Metering:`MeterUsage` API. Amazon ECS is supported in all AWS Regions that Metering:`MeterUsage` API is available except for GovCloud.

- Amazon EKS is supported in the following: us-east-1, us-east-2, us-west-1, us-west-2, eu-west-1, eu-central-1, eu-west-2, eu-west-3, eu-north-1, ap-east-1, ap-southeast-1, ap-northeast-1, ap-southeast-2, ap-northeast-2, ap-south-1, ca-central-1, sa-east-1.

 **Note**

For questions about adding AWS Regions for metering, contact [AWS Marketplace Seller Operations](#).

This document was last published on April 19, 2024.

# Actions

The following actions are supported:

- [BatchMeterUsage](#)
- [MeterUsage](#)
- [RegisterUsage](#)
- [ResolveCustomer](#)

# BatchMeterUsage

BatchMeterUsage is called from a SaaS application listed on AWS Marketplace to post metering records for a set of customers. BatchMeterUsage is used for metering SaaS Flexible Consumption Pricing (FCP). BatchMeterUsage is supported in the following AWS Regions:

## Commercial Regions

eu-north-1, me-south-1, ap-south-1, eu-west-3, ap-southeast-3, us-east-2, af-south-1, eu-west-1, me-central-1, eu-central-1, sa-east-1, ap-east-1, ap-south-2, us-east-1, ap-northeast-2, ap-northeast-3, eu-west-2, ap-southeast-4, eu-south-1, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-southeast-2, il-central-1, ca-central-1, eu-south-2, eu-central-2

## China Regions

cn-northwest-1

For identical requests, the API is idempotent; requests can be retried with the same records or a subset of the input records.

Every request to BatchMeterUsage is for one product. If you need to meter usage for multiple products, you must make multiple calls to BatchMeterUsage.

Usage records are expected to be submitted as quickly as possible after the event that is being recorded, and are not accepted more than 6 hours after the event.

BatchMeterUsage can process up to 25 UsageRecords at a time.

A UsageRecord can optionally include multiple usage allocations, to provide customers with usage data split into buckets by tags that you define (or allow the customer to define).

BatchMeterUsage returns a list of UsageRecordResult objects, showing the result for each UsageRecord, as well as a list of UnprocessedRecords, indicating errors in the service side that you should retry.

BatchMeterUsage requests must be less than 1MB in size.

**Note**

For an example of using BatchMeterUsage, see [BatchMeterUsage code example](#) in the *AWS Marketplace Seller Guide*.

## Request Syntax

```
{  
    "ProductCode": "string",  
    "UsageRecords": [  
        {  
            "CustomerIdentifier": "string",  
            "Dimension": "string",  
            "Quantity": number,  
            "Timestamp": number,  
            "UsageAllocations": [  
                {  
                    "AllocatedUsageQuantity": number,  
                    "Tags": [  
                        {  
                            "Key": "string",  
                            "Value": "string"  
                        }  
                    ]  
                }  
            ]  
        }  
    ]  
}
```

## Request Parameters

The request accepts the following data in JSON format.

**Note**

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

## ProductCode

Product code is used to uniquely identify a product in AWS Marketplace. The product code should be the same as the one used during the publishing of a new product.

Type: String

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

Pattern: ^[-a-zA-Z0-9/=:\_.@]\*\$

Required: Yes

## UsageRecords

The set of UsageRecords to submit. BatchMeterUsage accepts up to 25 UsageRecords at a time.

Type: Array of [UsageRecord](#) objects

Array Members: Minimum number of 0 items. Maximum number of 25 items.

Required: Yes

## Response Syntax

```
{  
  "Results": [  
    {  
      "MeteringRecordId": "string",  
      "Status": "string",  
      "UsageRecord": {  
        "CustomerIdentifier": "string",  
        "Dimension": "string",  
        "Quantity": number,  
        "Timestamp": number,  
        "UsageAllocations": [  
          {  
            "AllocatedUsageQuantity": number,  
            "Tags": [  
              {  
                "Key": "string",  
                "Value": "string"  
              }  
            ]  
          }  
        ]  
      }  
    }  
  ]  
}
```

```
        "Value": "string"
    }
]
}
]
},
"UnprocessedRecords": [
{
    "CustomerIdentifier": "string",
    "Dimension": "string",
    "Quantity": number,
    "Timestamp": number,
    "UsageAllocations": [
        {
            "AllocatedUsageQuantity": number,
            "Tags": [
                {
                    "Key": "string",
                    "Value": "string"
                }
            ]
        }
    ]
}
]
```

## Response Elements

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

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

### Results

Contains all UsageRecords processed by BatchMeterUsage. These records were either honored by AWS Marketplace Metering Service or were invalid. Invalid records should be fixed before being resubmitted.

Type: Array of [UsageRecordResult](#) objects

## UnprocessedRecords

Contains all UsageRecords that were not processed by BatchMeterUsage. This is a list of UsageRecords. You can retry the failed request by making another BatchMeterUsage call with this list as input in the BatchMeterUsageRequest.

Type: Array of [UsageRecord](#) objects

Array Members: Minimum number of 0 items. Maximum number of 25 items.

## Errors

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

### **DisabledApiException**

The API is disabled in the Region.

HTTP Status Code: 400

### **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

### **InvalidCustomerIdentifierException**

You have metered usage for a CustomerIdentifier that does not exist.

HTTP Status Code: 400

### **InvalidProductCodeException**

The product code passed does not match the product code used for publishing the product.

HTTP Status Code: 400

### **InvalidTagException**

The tag is invalid, or the number of tags is greater than 5.

HTTP Status Code: 400

## InvalidUsageAllocationsException

Sum of allocated usage quantities is not equal to the usage quantity.

HTTP Status Code: 400

## InvalidUsageDimensionException

The usage dimension does not match one of the UsageDimensions associated with products.

HTTP Status Code: 400

## ThrottlingException

The calls to the API are throttled.

HTTP Status Code: 400

## TimestampOutOfBoundsException

The timestamp value passed in the UsageRecord is out of allowed range.

For BatchMeterUsage, if any of the records are outside of the allowed range, the entire batch is not processed. You must remove invalid records and try again.

HTTP Status Code: 400

## See Also

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)



# MeterUsage

API to emit metering records. For identical requests, the API is idempotent. It simply returns the metering record ID. This is used for metering Flexible Consumption Pricing (FCP) Amazon Machine Images (AMI) and Container products. MeterUsage is available in the following Regions:

## Commercial Regions

eu-north-1, me-south-1, ap-south-1, eu-west-3, ap-southeast-3, us-east-2, af-south-1, eu-west-1, me-central-1, eu-central-1, sa-east-1, ap-east-1, ap-south-2, us-east-1, ap-northeast-2, ap-northeast-3, eu-west-2, ap-southeast-4, eu-south-1, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-southeast-2, il-central-1, ca-central-1, eu-south-2, eu-central-2

## GovCloud Regions

us-gov-east-1, us-gov-west-1

MeterUsage is authenticated on the buyer's AWS account using credentials from the Amazon EC2 instance, Amazon ECS task, or Amazon EKS pod.

MeterUsage can optionally include multiple usage allocations, to provide customers with usage data split into buckets by tags that you define (or allow the customer to define).

Usage records are expected to be submitted as quickly as possible after the event that is being recorded, and are not accepted more than 6 hours after the event.

## Request Syntax

```
{
  "DryRun": boolean,
  "ProductCode": string,
  "Timestamp": number,
  "UsageAllocations": [
    {
      "AllocatedUsageQuantity": number,
      "Tags": [
        {
          "Key": string,
          "Value": string
        }
      ]
    }
  ]
}
```

```
        ]
    }
],
"UsageDimension": "string",
"UsageQuantity": number
}
```

## Request Parameters

The request accepts the following data in JSON format.

 **Note**

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

### ProductCode

Product code is used to uniquely identify a product in AWS Marketplace. The product code should be the same as the one used during the publishing of a new product.

Type: String

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

Pattern: ^[-a-zA-Z0-9/:\_.@]\*\$

Required: Yes

### Timestamp

Timestamp, in UTC, for which the usage is being reported. Your application can meter usage for up to six hours in the past. Make sure the timestamp value is not before the start of the software usage.

Type: Timestamp

Required: Yes

### UsageDimension

It will be one of the fcp dimension name provided during the publishing of the product.

Type: String

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

Pattern: [\s\S]+

Required: Yes

### DryRun

Checks whether you have the permissions required for the action, but does not make the request. If you have the permissions, the request returns DryRunOperation; otherwise, it returns UnauthorizedException. Defaults to false if not specified.

Type: Boolean

Required: No

### UsageAllocations

The set of UsageAllocations to submit.

The sum of all UsageAllocation quantities must equal the UsageQuantity of the MeterUsage request, and each UsageAllocation must have a unique set of tags (include no tags).

Type: Array of [UsageAllocation](#) objects

Array Members: Minimum number of 1 item. Maximum number of 2500 items.

Required: No

### UsageQuantity

Consumption value for the hour. Defaults to 0 if not specified.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 2147483647.

Required: No

## Response Syntax

```
{  
  "MeteringRecordId": "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.

### MeteringRecordId

Metering record id.

Type: String

## Errors

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

### **CustomerNotEntitledException**

Exception thrown when the customer does not have a valid subscription for the product.

HTTP Status Code: 400

### **DuplicateRequestException**

A metering record has already been emitted by the same EC2 instance, ECS task, or EKS pod for the given {usageDimension, timestamp} with a different usageQuantity.

HTTP Status Code: 400

### **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

### **InvalidEndpointRegionException**

The endpoint being called is in a AWS Region different from your EC2 instance, ECS task, or EKS pod. The Region of the Metering Service endpoint and the AWS Region of the resource must match.

HTTP Status Code: 400

### **InvalidProductCodeException**

The product code passed does not match the product code used for publishing the product.

HTTP Status Code: 400

### **InvalidTagException**

The tag is invalid, or the number of tags is greater than 5.

HTTP Status Code: 400

### **InvalidUsageAllocationsException**

Sum of allocated usage quantities is not equal to the usage quantity.

HTTP Status Code: 400

### **InvalidUsageDimensionException**

The usage dimension does not match one of the UsageDimensions associated with products.

HTTP Status Code: 400

### **ThrottlingException**

The calls to the API are throttled.

HTTP Status Code: 400

### **TimestampOutOfBoundsException**

The timestamp value passed in the UsageRecord is out of allowed range.

For BatchMeterUsage, if any of the records are outside of the allowed range, the entire batch is not processed. You must remove invalid records and try again.

HTTP Status Code: 400

## **See Also**

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

- [AWS Command Line Interface](#)
- [AWS SDK for .NET](#)
- [AWS SDK for C++](#)
- [AWS SDK for Go](#)
- [AWS SDK for Java V2](#)
- [AWS SDK for JavaScript V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# RegisterUsage

Paid container software products sold through AWS Marketplace must integrate with the AWS Marketplace Metering Service and call the `RegisterUsage` operation for software entitlement and metering. Free and BYOL products for Amazon ECS or Amazon EKS aren't required to call `RegisterUsage`, but you may choose to do so if you would like to receive usage data in your seller reports. The sections below explain the behavior of `RegisterUsage`. `RegisterUsage` performs two primary functions: metering and entitlement.

- *Entitlement:* `RegisterUsage` allows you to verify that the customer running your paid software is subscribed to your product on AWS Marketplace, enabling you to guard against unauthorized use. Your container image that integrates with `RegisterUsage` is only required to guard against unauthorized use at container startup, as such a `CustomerNotSubscribedException` or `PlatformNotSupportedException` will only be thrown on the initial call to `RegisterUsage`. Subsequent calls from the same Amazon ECS task instance (e.g. task-id) or Amazon EKS pod will not throw a `CustomerNotSubscribedException`, even if the customer unsubscribes while the Amazon ECS task or Amazon EKS pod is still running.
- *Metering:* `RegisterUsage` meters software use per ECS task, per hour, or per pod for Amazon EKS with usage prorated to the second. A minimum of 1 minute of usage applies to tasks that are short lived. For example, if a customer has a 10 node Amazon ECS or Amazon EKS cluster and a service configured as a Daemon Set, then Amazon ECS or Amazon EKS will launch a task on all 10 cluster nodes and the customer will be charged:  $(10 * \text{hourly\_rate})$ . Metering for software use is automatically handled by the AWS Marketplace Metering Control Plane -- your software is not required to perform any metering specific actions, other than call `RegisterUsage` once for metering of software use to commence. The AWS Marketplace Metering Control Plane will also continue to bill customers for running ECS tasks and Amazon EKS pods, regardless of the customers subscription state, removing the need for your software to perform entitlement checks at runtime. For containers, `RegisterUsage` should be called immediately at launch. If you don't register the container in the first 6 hours of the container launch, AWS Marketplace Metering Service doesn't provide any metering guarantees for previous months. However, the metering will continue for the current month forward until the container ends. `RegisterUsage` is used for metering Paid Hourly container products. `RegisterUsage` is supported in the following AWS Regions:

## Commercial Regions

eu-north-1, ap-south-1, eu-west-3, us-east-2, eu-west-1, eu-central-1, sa-east-1, ap-east-1, us-east-1, ap-northeast-2, eu-west-2, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-southeast-2, ca-central-1

## Request Syntax

```
{  
    "Nonce": "string",  
    "ProductCode": "string",  
    "PublicKeyVersion": number  
}
```

## Request Parameters

The request accepts the following data in JSON format.

### Note

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

### ProductCode

Product code is used to uniquely identify a product in AWS Marketplace. The product code should be the same as the one used during the publishing of a new product.

Type: String

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

Pattern: ^[-a-zA-Z0-9/=:\_.@]\*\$

Required: Yes

### PublicKeyVersion

Public Key Version provided by AWS Marketplace

Type: Integer

Valid Range: Minimum value of 1.

Required: Yes

### Nonce

(Optional) To scope down the registration to a specific running software instance and guard against replay attacks.

Type: String

Length Constraints: Maximum length of 255.

Pattern: [\s\S]\*

Required: No

## Response Syntax

```
{  
  "PublicKeyRotationTimestamp": number,  
  "Signature": "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.

### PublicKeyRotationTimestamp

(Optional) Only included when public key version has expired

Type: Timestamp

### Signature

JWT Token

Type: String

Pattern: [\s\S]+

## Errors

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

### **CustomerNotEntitledException**

Exception thrown when the customer does not have a valid subscription for the product.

HTTP Status Code: 400

### **DisabledApiException**

The API is disabled in the Region.

HTTP Status Code: 400

### **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

### **InvalidProductCodeException**

The product code passed does not match the product code used for publishing the product.

HTTP Status Code: 400

### **InvalidPublicKeyVersionException**

Public Key version is invalid.

HTTP Status Code: 400

### **InvalidRegionException**

`RegisterUsage` must be called in the same AWS Region the ECS task was launched in. This prevents a container from hardcoding a Region (e.g. `withRegion("us-east-1")`) when calling `RegisterUsage`.

HTTP Status Code: 400

### **PlatformNotSupportedException**

AWS Marketplace does not support metering usage from the underlying platform. Currently, Amazon ECS, Amazon EKS, and AWS Fargate are supported.

HTTP Status Code: 400

## ThrottlingException

The calls to the API are throttled.

HTTP Status Code: 400

## Examples

### Example

Below are the sample request and response for RegisterUsage

#### Sample Request

```
{  
    "ProductCode" : "cqcvf9f0ugw8rkbgmf1c9dxyz",  
    "PublicKeyVersion": 1,  
    "Nonce": "2ead20e4-3e6d-42cd-8f56-24f02d1cc4e1"  
}
```

#### Sample Response

```
{  
    "PublicKeyRotationTimestamp": null,  
    "Signature": "eyJhbGciOiJQUzI1Ni..."  
}  
  
// Where the signature is composed of 3 dot-separated,  
// base-64 URL Encoded sections.  
// e.g. eyJhbGcVCJ9.eyJzdWIMzkwMjJ9.rr09Qw0SXRWT  
  
// Section 1: Header/Algorithm  
  
{  
    "alg": "PS256",  
    "typ": "JWT"
```

```
}

// Section 2: Payload

{
    "ProductCode" : "cqcvf9f0ugw8rkbgmf1c9dxyz",
    "PublicKeyVersion": 1,
    "Nonce": "2ead20e4-3e6d-42cd-8f56-24f02d1cc4e1",
    "PublicKeyRotationTimestamp": null
}

// Section 3: RSA-PSS SHA256 signature

"rr09Q4FEi3gweH3X4lrt2okf5zwIatUUwERlw016wTy_21Nv8S..."
```

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

## ResolveCustomer

`ResolveCustomer` is called by a SaaS application during the registration process. When a buyer visits your website during the registration process, the buyer submits a registration token through their browser. The registration token is resolved through this API to obtain a `CustomerIdentifier` along with the `CustomerAWSAccountId` and `ProductCode`.

### Note

The API needs to be called from the seller account id used to publish the SaaS application to successfully resolve the token.

For an example of using `ResolveCustomer`, see [ResolveCustomer code example](#) in the [AWS Marketplace Seller Guide](#).

Permission is required for this operation. Your IAM role or user performing this operation requires a policy to allow the `aws-marketplace:ResolveCustomer` action. For more information, see [Actions, resources, and condition keys for AWS Marketplace Metering Service](#) in the *Service Authorization Reference*.

`ResolveCustomer` is supported in the following AWS Regions:

### Commercial Regions

eu-north-1, me-south-1, ap-south-1, eu-west-3, ap-southeast-3, us-east-2, af-south-1, eu-west-1, me-central-1, eu-central-1, sa-east-1, ap-east-1, ap-south-2, us-east-1, ap-northeast-2, ap-northeast-3, eu-west-2, ap-southeast-4, eu-south-1, ap-northeast-1, us-west-2, us-west-1, ap-southeast-1, ap-southeast-2, il-central-1, ca-central-1, eu-south-2, eu-central-2

### China Regions

cn-northwest-1

## Request Syntax

```
{  
    "RegistrationToken": "string"  
}
```

## Request Parameters

The request accepts the following data in JSON format.

 **Note**

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

### [RegistrationToken](#)

When a buyer visits your website during the registration process, the buyer submits a registration token through the browser. The registration token is resolved to obtain a `CustomerIdentifier` along with the `CustomerAWSAccountId` and `ProductCode`.

Type: String

Pattern: `[\s\S]+`

Required: Yes

## Response Syntax

```
{  
  "CustomerAWSAccountId": "string",  
  "CustomerIdentifier": "string",  
  "ProductCode": "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.

### [CustomerAWSAccountId](#)

The `CustomerAWSAccountId` provides the AWS account ID associated with the `CustomerIdentifier` for the individual customer.

Type: String

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

Pattern: ^[0-9]+\$

### CustomerIdentifier

The `CustomerIdentifier` is used to identify an individual customer in your application. Calls to `BatchMeterUsage` require `CustomerIdentifiers` for each `UsageRecord`.

Type: String

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

Pattern: [\s\S]+

### ProductCode

The product code is returned to confirm that the buyer is registering for your product. Subsequent `BatchMeterUsage` calls should be made using this product code.

Type: String

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

Pattern: ^[-a-zA-Z0-9/=:\_@]\*\$

## Errors

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

### **DisabledApiException**

The API is disabled in the Region.

HTTP Status Code: 400

### **ExpiredTokenException**

The submitted registration token has expired. This can happen if the buyer's browser takes too long to redirect to your page, the buyer has resubmitted the registration token, or your application has held on to the registration token for too long. Your SaaS registration website should redeem this token as soon as it is submitted by the buyer's browser.

HTTP Status Code: 400

## 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

## InvalidTokenException

Registration token is invalid.

HTTP Status Code: 400

## ThrottlingException

The calls to the 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 V3](#)
- [AWS SDK for PHP V3](#)
- [AWS SDK for Python](#)
- [AWS SDK for Ruby V3](#)

# Data Types

The AWSMarketplace Metering 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:

- [Tag](#)
- [UsageAllocation](#)
- [UsageRecord](#)
- [UsageRecordResult](#)

# Tag

Metadata assigned to an allocation. Each tag is made up of a key and a value.

## Contents

### Note

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

### Key

One part of a key-value pair that makes up a tag. A key is a label that acts like a category for the specific tag values.

Type: String

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

Pattern: ^[a-zA-Z0-9+ -=.\_:@]+\$

Required: Yes

### Value

One part of a key-value pair that makes up a tag. A value acts as a descriptor within a tag category (key). The value can be empty or null.

Type: String

Length Constraints: Minimum length of 1. Maximum length of 256.

Pattern: ^[a-zA-Z0-9+ -=.\_:@]+\$

Required: Yes

## See Also

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

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

# UsageAllocation

Usage allocations allow you to split usage into buckets by tags.

Each UsageAllocation indicates the usage quantity for a specific set of tags.

## Contents

### Note

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

### AllocatedUsageQuantity

The total quantity allocated to this bucket of usage.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 2147483647.

Required: Yes

### Tags

The set of tags that define the bucket of usage. For the bucket of items with no tags, this parameter can be left out.

Type: Array of [Tag](#) objects

Array Members: Minimum number of 1 item. Maximum number of 5 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](#)

# UsageRecord

A UsageRecord indicates a quantity of usage for a given product, customer, dimension and time.

Multiple requests with the same UsageRecords as input will be de-duplicated to prevent double charges.

## Contents

### Note

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

### CustomerIdentifier

The CustomerIdentifier is obtained through the ResolveCustomer operation and represents an individual buyer in your application.

Type: String

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

Pattern: [\s\S]+

Required: Yes

### Dimension

During the process of registering a product on AWS Marketplace, dimensions are specified. These represent different units of value in your application.

Type: String

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

Pattern: [\s\S]+

Required: Yes

### Timestamp

Timestamp, in UTC, for which the usage is being reported.

Your application can meter usage for up to one hour in the past. Make sure the timestamp value is not before the start of the software usage.

Type: Timestamp

Required: Yes

## Quantity

The quantity of usage consumed by the customer for the given dimension and time. Defaults to 0 if not specified.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 2147483647.

Required: No

## UsageAllocations

The set of UsageAllocations to submit. The sum of all UsageAllocation quantities must equal the Quantity of the UsageRecord.

Type: Array of [UsageAllocation](#) objects

Array Members: Minimum number of 1 item. Maximum number of 2500 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](#)

# UsageRecordResult

A UsageRecordResult indicates the status of a given UsageRecord processed by BatchMeterUsage.

## Contents

### Note

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

### MeteringRecordId

The MeteringRecordId is a unique identifier for this metering event.

Type: String

Required: No

### Status

The UsageRecordResult Status indicates the status of an individual UsageRecord processed by BatchMeterUsage.

- *Success*- The UsageRecord was accepted and honored by BatchMeterUsage.
- *CustomerNotSubscribed*- The CustomerIdentifier specified is not able to use your product. The UsageRecord was not honored. There are three causes for this result:
  - The customer identifier is invalid.
  - The customer identifier provided in the metering record does not have an active agreement or subscription with this product. Future UsageRecords for this customer will fail until the customer subscribes to your product.
  - The customer's AWS account was suspended.
- *DuplicateRecord*- Indicates that the UsageRecord was invalid and not honored. A previously metered UsageRecord had the same customer, dimension, and time, but a different quantity.

Type: String

Valid Values: Success | CustomerNotSubscribed | DuplicateRecord

Required: No

## UsageRecord

The UsageRecord that was part of the BatchMeterUsage request.

Type: [UsageRecord](#) 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](#)

# Common Errors

This section lists the errors common to the API actions of all AWS services. For errors specific to an API action for this service, see the topic for that API action.

## **AccessDeniedException**

You do not have sufficient access to perform this action.

HTTP Status Code: 400

## **IncompleteSignature**

The request signature does not conform to AWS standards.

HTTP Status Code: 400

## **InternalFailure**

The request processing has failed because of an unknown error, exception or failure.

HTTP Status Code: 500

## **InvalidAction**

The action or operation requested is invalid. Verify that the action is typed correctly.

HTTP Status Code: 400

## **InvalidClientTokenId**

The X.509 certificate or AWS access key ID provided does not exist in our records.

HTTP Status Code: 403

## **NotAuthorized**

You do not have permission to perform this action.

HTTP Status Code: 400

## **OptInRequired**

The AWS access key ID needs a subscription for the service.

HTTP Status Code: 403

## **RequestExpired**

The request reached the service more than 15 minutes after the date stamp on the request or more than 15 minutes after the request expiration date (such as for pre-signed URLs), or the date stamp on the request is more than 15 minutes in the future.

HTTP Status Code: 400

## **ServiceUnavailable**

The request has failed due to a temporary failure of the server.

HTTP Status Code: 503

## **ThrottlingException**

The request was denied due to request throttling.

HTTP Status Code: 400

## **ValidationException**

The input fails to satisfy the constraints specified by an AWS service.

HTTP Status Code: 400