## Table of Contents

Welcome ........................................................................................................................................... 1  
Actions ............................................................................................................................................. 2  
  BatchMeterUsage ....................................................................................................................... 3  
    Request Syntax .................................................................................................................. 3  
    Request Parameters ............................................................................................................ 3  
    Response Syntax ................................................................................................................ 3  
    Response Elements ............................................................................................................. 4  
    Errors ............................................................................................................................... 4  
    See Also ............................................................................................................................ 5  
  MeterUsage ............................................................................................................................... 6  
    Request Syntax .................................................................................................................. 6  
    Request Parameters ............................................................................................................ 6  
    Response Syntax ................................................................................................................ 7  
    Response Elements ............................................................................................................. 7  
    Errors ............................................................................................................................... 7  
    See Also ............................................................................................................................ 8  
  ResolveCustomer ........................................................................................................................ 9  
    Request Syntax .................................................................................................................. 9  
    Request Parameters ............................................................................................................ 9  
    Response Syntax ................................................................................................................ 9  
    Response Elements ............................................................................................................. 9  
    Errors ............................................................................................................................... 10  
    See Also .......................................................................................................................... 10  
  Data Types ...................................................................................................................................... 11  
  UsageRecord ............................................................................................................................ 12  
    Contents ............................................................................................................................. 12  
    See Also ............................................................................................................................ 12  
  UsageRecordResult ..................................................................................................................... 14  
    Contents ............................................................................................................................. 14  
    See Also ............................................................................................................................ 14  
  Common Errors ........................................................................................................................ 15
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.

Submitting Metering Records

- **MeterUsage**- Submits the metering record for a Marketplace product. MeterUsage is called from an EC2 instance.
- **BatchMeterUsage**- Submits the metering record for a set of customers. BatchMeterUsage is called from a software-as-a-service (SaaS) application.

Accepting New Customers

- **ResolveCustomer**- 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 the browser. The Registration Token is resolved through this API to obtain a CustomerIdentifier and Product Code.

BatchMeterUsage API calls are captured by AWS CloudTrail. You can use Cloudtrail to verify the SaaS metering records 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 AWS CloudTrail User Guide.

This document was last published on June 21, 2018.
Actions

The following actions are supported:

- BatchMeterUsage (p. 3)
- MeterUsage (p. 6)
- ResolveCustomer (p. 9)
**BatchMeterUsage**

BatchMeterUsage is called from a SaaS application listed on the AWS Marketplace to post metering records for a set of customers.

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.

BatchMeterUsage can process up to 25 UsageRecords at a time.

**Request Syntax**

```
{
  "ProductCode": "string",
  "UsageRecords": [
    {
      "CustomerIdentifier": "string",
      "Dimension": "string",
      "Quantity": number,
      "Timestamp": number
    }
  ]
}
```

**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 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.

Required: Yes

**UsageRecords (p. 3)**

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

Type: Array of UsageRecord (p. 12) 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
    }
  },
  
  "UnprocessedRecords": [ 
    {
      "CustomerIdentifier": "string",
      "Dimension": "string",
      "Quantity": number,
      "Timestamp": number
    }
  ]
]

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

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

Results (p. 3)
Contains all UsageRecords processed by BatchMeterUsage. These records were either honored by
AWS Marketplace Metering Service or were invalid.

Type: Array of UsageRecordResult (p. 14) objects

UnprocessedRecords (p. 3)
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 (p. 12) 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 (p. 15).

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

InvalidUsageDimensionException
The usage dimension does not match one of the UsageDimensions associated with products.
HTTP Status Code: 400

ThrottlingException
The calls to the MeterUsage API are throttled.
HTTP Status Code: 400

TimestampOutOfBoundsException
The timestamp value passed in the meterUsage() is out of allowed range.
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
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
MeterUsage

API to emit metering records. For identical requests, the API is idempotent. It simply returns the metering record ID.

MeterUsage is authenticated on the buyer's AWS account, generally when running from an EC2 instance on the AWS Marketplace.

Request Syntax

```json
{
   "DryRun": boolean,
   "ProductCode": "string",
   "Timestamp": number,
   "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.

**DryRun (p. 6)**

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.

Type: Boolean

Required: Yes

**ProductCode (p. 6)**

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.

Required: Yes

**Timestamp (p. 6)**

Timestamp of the hour, recorded in UTC. The seconds and milliseconds portions of the timestamp will be ignored.

Type: Timestamp

Required: Yes

**UsageDimension (p. 6)**

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.

Required: Yes

**UsageQuantity (p. 6)**

Consumption value for the hour.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 1000000.

Required: Yes

---

**Response Syntax**

```json
{
    "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 (p. 7)**

Type: String

---

**Errors**

For information about the errors that are common to all actions, see Common Errors (p. 15).

**DuplicateRequestException**

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

HTTP Status Code: 400

**InternalServerErrorException**

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 region different from your EC2 instance. The region of the Metering service endpoint and the region of the EC2 instance 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

**InvalidUsageDimensionException**

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

HTTP Status Code: 400

**ThrottlingException**

The calls to the MeterUsage API are throttled.

HTTP Status Code: 400

**TimestampOutOfBoundsException**

The timestamp value passed in the meterUsage() is out of allowed range.

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
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
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 and product code.

Request Syntax

```json
{
   "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 (p. 9)**

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 and product code.

Type: String

Pattern: \S+

Required: Yes

Response Syntax

```json
{
   "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.

**CustomerIdentifier (p. 9)**

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.
ProductCode (p. 9)

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.

Errors

For information about the errors that are common to all actions, see Common Errors (p. 15).

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

HTTP Status Code: 400

ThrottlingException

The calls to the MeterUsage 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
- AWS SDK for JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V2
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:

- UsageRecord (p. 12)
- UsageRecordResult (p. 14)
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 deduplicated 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.

Required: Yes

Dimension

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

Type: String

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

Required: Yes

Quantity

The quantity of usage consumed by the customer for the given dimension and time.

Type: Integer

Valid Range: Minimum value of 0. Maximum value of 1000000.

Required: Yes

Timestamp

Timestamp of the hour, recorded in UTC. The seconds and milliseconds portions of the timestamp will be ignored.

Your application can meter usage for up to one hour in the past.

Type: Timestamp

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
• AWS SDK for Ruby V2
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 subscribed to your product. The UsageRecord was not honored. Future UsageRecords for this customer will fail until the customer subscribes to your product.
- 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 (p. 12) 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
- AWS SDK for Ruby V2
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

**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