

Tag Management Service

API Reference

Issue 02
Date 2022-10-31



Copyright © Huawei Technologies Co., Ltd. 2024. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions



HUAWEI and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Security Declaration

Vulnerability

Huawei's regulations on product vulnerability management are subject to the *Vul. Response Process*. For details about this process, visit the following web page:

<https://www.huawei.com/en/psirt/vul-response-process>

For vulnerability information, enterprise customers can visit the following web page:

<https://securitybulletin.huawei.com/enterprise/en/security-advisory>

Contents

1 Before You Start	1
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Endpoints.....	1
1.4 Notes and Constraints.....	1
1.5 Concepts.....	2
2 API Overview	4
3 Calling APIs	5
3.1 Making an API Request.....	5
3.2 Authentication.....	9
3.3 Response.....	11
4 Getting Started	13
5 API Description	15
5.1 Querying the API Version.....	15
5.1.1 Querying the API Versions.....	15
5.1.2 Querying Details About a Specified TMS API Version.....	17
5.2 Predefined Tag Operations.....	19
5.2.1 Creating or Deleting Predefined Tags.....	19
5.2.2 Querying Predefined Tags.....	21
5.2.3 Modifying Predefined Tags.....	24
5.3 Querying Services Supported by TMS.....	26
5.3.1 Querying Services Supported by TMS.....	26
6 Permissions Policies and Supported Actions	30
6.1 Introduction.....	30
6.2 TMS API Actions.....	31
A Appendix	32
A.1 Status Codes.....	32
A.2 Error Codes.....	33
A.3 Obtaining a Project ID.....	36
B Change History	38

1 Before You Start

1.1 Overview

Welcome to *Tag Management Service API Reference*. Tags are used to identify cloud resources. When you have many cloud resources of the same type, you can use tags to classify cloud resources by dimension (for example, use, owner, or environment). Tag Management Service (TMS) is a visualized service for fast and unified cross-region tagging and categorization of cloud services.

This document describes how to use application programming interfaces (APIs) to perform operations on tags, such as creating or deleting predefined tags, and querying or modify predefined tags. For details about all supported operations, see [API Overview](#).

If you plan to access TMS through an API, ensure that you are familiar with TMS concepts. For details, see section "Service Overview" in the *Tag Management Service User Guide*.

1.2 API Calling

TMS supports Representational State Transfer (REST) APIs, allowing you to call APIs using HTTPS. For details about API calling, see [Calling APIs](#).

1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. For the TMS endpoint, see [Regions and Endpoints](#).

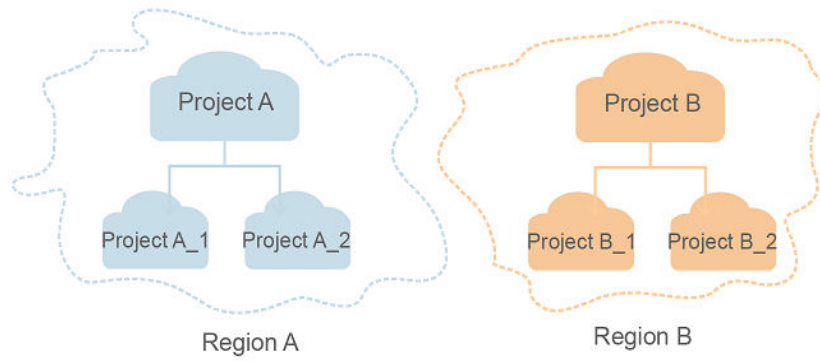
1.4 Notes and Constraints

- The number of tags that you can create is determined by your quota. To view or increase the quota, see section "Modifying Resource Quotas" in the *Tag Management Service User Guide*.
- For more constraints, see API description.

1.5 Concepts

- **Account**
An account is created upon successful registration. The account has full access permissions for all of its cloud services and resources. It can be used to reset user passwords and grant user permissions. The account is a payment entity, which should not be used directly to perform routine management. For security purposes, create Identity and Access Management (IAM) users and grant them permissions for routine management.
- **User**
An IAM user is created by an account in IAM to use cloud services. Each IAM user has its own identity credentials (password and access keys).
API authentication requires information such as the account name, username, and password.
- **Region**
A region is a geographic area in which cloud resources are deployed. Availability zones (AZs) in the same region can communicate with each other over an intranet, while AZs in different regions are isolated from each other. Deploying cloud resources in different regions can better suit certain user requirements or comply with local laws or regulations.
- **AZ**
An AZ comprises of one or more physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Computing, network, storage, and other resources in an AZ are logically divided into multiple clusters. AZs within a region are interconnected using high-speed optical fibers to allow you to build cross-AZ high-availability systems.
- **Project**
A project corresponds to a region. Default projects are defined to group and physically isolate resources (including computing, storage, and network resources) across regions. Users can be granted permissions in a default project to access all resources under their accounts in the region associated with the project. If you need more refined access control, create subprojects under a default project and create resources in subprojects. Then you can assign users the permissions required to access only the resources in the specific subprojects.

Figure 1-1 Project isolation model



2 API Overview

You can use TMS APIs to create, delete, query, or modify predefined tags, or query the version information.

Table 2-1 API description

API	Description
Querying the API Versions	Query the TMS API versions.
Querying Details About a Specified TMS API Version	Query details about a specified TMS API version.
Creating or Deleting Predefined Tags	Create or delete predefined tags. You can use predefined tags to tag resources.
Querying Predefined Tags	Query the predefined tag list of a specified tenant.
Modifying Predefined Tags	Modify predefined tags.

3 Calling APIs

3.1 Making an API Request

This section describes the structure of a REST API request, and uses the IAM API for **obtaining a user token** as an example to demonstrate how to call an API. The obtained token can then be used to authenticate the calling of other APIs.

Request URI

A request URI is in the following format:

{URI-scheme}://{Endpoint}/{resource-path}?{query-string}

Although a request URI is included in the request header, most programming languages or frameworks require the request URI to be transmitted separately.

Table 3-1 URI parameter description

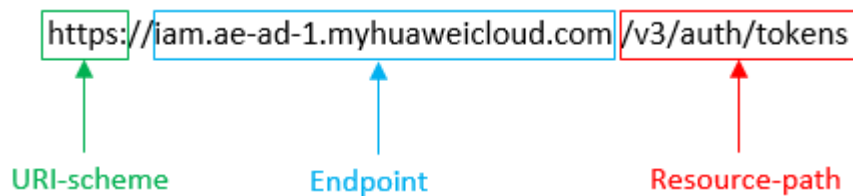
Parameter	Description
URI-scheme	Protocol used to transmit requests. All APIs use HTTPS.
Endpoint	Domain name or IP address of the server bearing the REST service. The endpoint varies between services in different regions. It can be obtained from Regions and Endpoints . For example, the endpoint of IAM in the UAE-Abu Dhabi region is iam.ae-ad-1.myhuaweicloud.com .
resource-path	Access path of an API for performing a specified operation. Obtain the path from the URI of an API. For example, the resource-path of the API used to obtain a user token is /v3/auth/tokens .

Parameter	Description
query-string	Query parameter, which is optional. Ensure that a question mark (?) is included before each query parameter that is in the format of <i>Parameter name=Parameter value</i> . For example, ?limit=10 indicates that a maximum of 10 data records will be displayed.

For example, to obtain an IAM token in the **UAE-Abu Dhabi** region, obtain the endpoint of IAM (**iam.ae-ad-1.myhuaweicloud.com**) for this region and the **resource-path** (**/v3/auth/tokens**) in the URI of the API used to **obtain a user token**. Then, construct the URI as follows:

```
https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
```

Figure 3-1 Example URI



NOTE

To simplify the URI display in this document, each API is provided only with a **resource-path** and a request method. The **URI-scheme** of all APIs is **HTTPS**, and the endpoints of all APIs in the same region are identical.

Request Methods

The HTTP protocol defines the following request methods that can be used to send a request to the server.

Table 3-2 HTTP methods

Method	Description
GET	Requests the server to return specified resources.
PUT	Requests the server to update specified resources.
POST	Requests the server to add resources or perform special operations.
DELETE	Requests the server to delete specified resources, for example, an object.
HEAD	Same as GET except that the server must return only the response header.

Method	Description
PATCH	Requests the server to update partial content of a specified resource. If the resource does not exist, a new resource will be created.

For example, in the case of the API used to [obtain a user token](#), the request method is **POST**. The request is as follows:

POST https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens

Request Header

You can also add additional header fields to a request, such as the fields required by a specified URI or HTTP method. For example, to request for the authentication information, add **Content-Type**, which specifies the request body type.

Common request header fields are as follows.

Table 3-3 Common request header fields

Parameter	Description	Mandatory	Example Value
Host	Specifies the server domain name and port number of the resources being requested. The value can be obtained from the URL of the service API. The value is in the format of <i>Hostname:Port number</i> . If the port number is not specified, the default port is used. The default port number for https is 443 .	No This field is mandatory for AK/SK authentication.	code.test.com or code.test.com:443
Content-Type	Specifies the type (or format) of the message body. The default value application/json is recommended. Other values of this field will be provided for specific APIs if any.	Yes	application/json
Content-Length	Specifies the length of the request body. The unit is byte.	No	3495

Parameter	Description	Mandatory	Example Value
X-Project-Id	Specifies the project ID. Obtain the project ID by following the instructions in Obtaining a Project ID .	No	e9993fc787d94b6c886cbaa340f9c0f4
X-Auth-Token	Specifies the user token. It is a response to the API for obtaining a user token (This is the only API that does not require authentication). After the request is processed, the value of X-Subject-Token in the response header is the token value.	No This field is mandatory for token authentication.	The following is part of an example token: MIIPAgYJKoZlhvcNAQcCo...ggg1BBIINPXsidG9rZ

 **NOTE**

In addition to supporting authentication using tokens, APIs support authentication using AK/SK, which uses SDKs to sign a request. During the signature, the **Authorization** (signature authentication) and **X-Sdk-Date** (time when a request is sent) headers are automatically added in the request.

For more details, see "Authentication Using AK/SK" in [Authentication](#).

The API used to [obtain a user token](#) does not require authentication. Therefore, only the **Content-Type** field needs to be added to requests for calling the API. An example of such requests is as follows:

```
POST https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

(Optional) Request Body

This part is optional. The body of a request is often sent in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The request body transfers content except the request header.

The request body varies between APIs. Some APIs do not require the request body, such as the APIs requested using the GET and DELETE methods.

In the case of the API used to [obtain a user token](#), the request parameters and parameter description can be obtained from the API request. The following provides an example request with a body included. Replace *username*, *domainname*, *\$ADMIN_PASS* (login password), and *xxxxxxxxxxxxxxxxxxxx* (project name) with the actual values. Obtain a project name from [Regions and Endpoints](#).

 NOTE

The **scope** parameter specifies where a token takes effect. You can set **scope** to an account or a project under an account. In the following example, the token takes effect only for the resources in a specified project. For more information about this API, see [Obtaining a User Token](#).

```
POST https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json

{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "$ADMIN_PASS", //You are advised to store it in ciphertext in the
configuration file or an environment variable and decrypt it when needed to ensure security.
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

If all data required for the API request is available, you can send the request to call the API through [curl](#), [Postman](#), or coding. In the response to the API used to obtain a user token, **X-Subject-Token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

3.2 Authentication

Requests for calling an API can be authenticated using either of the following methods:

- Token authentication: Requests are authenticated using tokens.
- AK/SK authentication: Requests are encrypted using AK/SK pairs. AK/SK authentication is recommended because it is more secure than token authentication.

Token Authentication

 NOTE

The validity period of a token is 24 hours. When using a token for authentication, cache it to prevent frequently calling the IAM API used to obtain a user token.

A token specifies temporary permissions in a computer system. During API authentication using a token, the token is added to requests to get permissions for calling the API. You can obtain a token by calling the [Obtaining User Token](#) API.

A cloud service can be deployed as either a project-level service or global service.

- For a project-level service, you need to obtain a project-level token. When you call the API, set **auth.scope** in the request body to **project**.
- For a global service, you need to obtain a global token. When you call the API, set **auth.scope** in the request body to **domain**.

TMS is a global service. When you call the API, set **auth.scope** in the request body to **domain**. For details about how to obtain the user token, see [Obtaining a User Token](#).

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username", // IAM user name
          "password": "*****", // IAM user password
          "domain": {
            "name": "domainname" // Name of the account to which the IAM user belongs
          }
        }
      }
    },
    "scope": {
      "domain": {
        "name": "xxxxxxxx" // Tenant name
      }
    }
  }
}
```

After a token is obtained, the **X-Auth-Token** header field must be added to requests to specify the token when calling other APIs. For example, if the token is **ABCDEFJ....**, **X-Auth-Token: ABCDEFJ....** can be added to a request as follows:

```
POST https://iam.ae-ad-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

AK/SK Authentication

NOTE

AK/SK authentication supports API requests with a body not larger than 12 MB. For API requests with a larger body, token authentication is recommended.

In AK/SK authentication, AK/SK is used to sign requests and the signature is then added to the requests for authentication.

- AK: access key ID, which is a unique identifier used in conjunction with a secret access key to sign requests cryptographically.
- SK: secret access key, which is used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

In AK/SK authentication, you can use an AK/SK to sign requests based on the signature algorithm or using the signing SDK. For details about how to sign requests and use the signing SDK, see [API Request Signing Guide](#).

 NOTE

The signing SDK is only used for signing requests and is different from the SDKs provided by services.

3.3 Response

Status Code

After sending a request, you will receive a response, including a status code, response header, and response body.

A status code is a group of digits, ranging from 1xx to 5xx. It indicates the status of a request. For more information, see [Status Codes](#).

For example, if status code **201** is returned for calling the API used to [obtain a user token](#), the request is successful.

Response Header

Similar to a request, a response also has a header, for example, **Content-Type**.

[Figure 3-2](#) shows the response header fields for the API used to [obtain a user token](#). The **X-Subject-Token** header field is the desired user token. This token can then be used to authenticate the calling of other APIs.

 NOTE

For security purposes, you are advised to set the token in ciphertext in configuration files or environment variables and decrypt it when using it.

Figure 3-2 Header fields of the response to the request for obtaining a user token

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5
x-subject-token → [REDACTED]
x-xss-protection → 1; mode=block;
```

(Optional) Response Body

The body of a response is often returned in a structured format (for example, JSON or XML) as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following is part of the response body for the API used to [obtain a user token](#).

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "az-01",
            .....

```

If an error occurs during API calling, an error code and a message will be displayed. The following shows an error response body.

```
{
  "error_msg": "The request message format is invalid.",
  "error_code": "IMG.0001"
}
```

In the response body, **error_code** is an error code, and **error_msg** provides information about the error.

4 Getting Started

This section describes how to create predefined tags by invoking the TMS API.

NOTE

The token obtained from IAM is valid for only 24 hours. If you want to use a token for authentication, you can cache it to avoid frequently calling the IAM API.

Involved APIs

If you use a token for authentication, you must obtain the user's token and add **X-Auth-Token** to the request header of the IMS API when making an API call.

- IAM API used to obtain the token
- TMS API used to create predefined tags

Procedure

1. Obtain the token by referring to [Authentication](#).
2. Send **POST https://TMS endpoint/v1.0/predefine_tags/action**.

Add **Content-Type** and **X-Auth-Token** to the request header.

Specify the following parameters in the request body:

```
{
  "action": "Create", //Operation (mandatory, string)
  "tags": [
    {
      "key": "ENV1", //Key (mandatory, string)
      "value": "DEV1" //Value (mandatory, string)
    },
    {
      "key": "ENV2",
      "value": "DEV2"
    }
  ]
}
```

After the request is sent, the status code 204 is returned and the response body is empty.

 **NOTE**

- The response message may be a success or a failure. This document takes the successful response as an example.
- If the request fails, an error code and error information are returned. For details, see [Error Codes](#).
- For details about the elements and return values of response messages, see [Creating and Deleting Predefined Tags](#).

5 API Description

5.1 Querying the API Version

5.1.1 Querying the API Versions

Function

This API is used to query the versions of all TMS APIs.

URI

GET /

Request

Example request

```
GET https://{TMS endpoint}/
```

NOTE

You can obtain the TMS endpoint at [Regions and Endpoints](#).

Response

- Parameter description

Table 5-1 Parameters in the response

Name	Type	Description
versions	Array	Specifies all API versions. For details, see Table 5-2 .

- **versions** field description

Table 5-2 Parameter description

Name	Type	Description
id	String	Specifies the version ID, for example, v1.0.
links	List<Link>	Specifies the API URL. For details, see Table 5-3 .
version	String	If the APIs of this version support microversions, set this parameter to the supported latest microversion. If not, leave this parameter blank.
status	String	Specifies the version status. Possible statuses are as follows: <ul style="list-style-type: none"> ● CURRENT: indicates that the version is the primary version. ● SUPPORTED: indicates that the version is an old version, but it is still supported. ● DEPRECATED: indicates a deprecated version which may be deleted later.
updated	String	Specifies the version release time, which must be the UTC time. For example, the release time of TMS 1.0 is 2016-12-09T00:00:00Z.
min_version	String	If the APIs of this version support microversions, set this parameter to the supported earliest microversion. If not, leave this parameter blank.

- **Links** field description

Table 5-3 Parameter description

Name	Type	Description
href	String	Specifies the API URL.
rel	String	self

- Example response

Status code: 200

Successful operation

```
{
  "versions": [
    {
      "id": "v1.0",
      "links": [
        {
          "rel": "self",
```

```

        "href": "https://API URL/v1.0"
      }
    ],
    "version": "",
    "status": "CURRENT",
    "updated": "2016-12-09T00:00:00Z",
    "min_version": ""
  }
]
}

```

Status Codes

See [Status Codes](#).

Error Codes

See [Error Codes](#).

5.1.2 Querying Details About a Specified TMS API Version

Function

This API is used to query the details of a specified TMS API version.

URI

GET `/{api_version}`

Request

- Parameter description

Table 5-4 Parameters in the request

Parameter	Mandatory	Type	Description
api_version	Yes	String	Specifies the API version.

- Example request
GET `https://{TMS endpoint}/v1.0`

Response

- Parameter description

Table 5-5 Parameters in the response

Name	Type	Description
version	object	Specifies the version of a specified API. For details, see Table 5-6 .

- **version** field description

Table 5-6 Parameter description

Name	Type	Description
id	String	Specifies the version ID, for example, v1.0.
links	List<Link>	Specifies the API URL. For details, see Table 5-7 .
version	String	If the APIs of this version support microversions, set this parameter to the supported latest microversion. If not, leave this parameter blank.
status	String	Specifies the version status. Possible values are as follows: <ul style="list-style-type: none"> • CURRENT: indicates that the version is the primary version. • SUPPORTED: indicates that the version is an old version, but it is still supported. • DEPRECATED: indicates a deprecated version which may be deleted later.
updated	String	Specifies the version release time, which must be the UTC time. For example, the release time of TMS 1.0 is 2016-12-09T00:00:00Z.
min_version	String	If the APIs of this version support microversions, set this parameter to the supported earliest microversion. If not, leave this parameter blank.

- **Links** field description

Table 5-7 Parameter description

Name	Type	Description
href	String	Specifies the API URL.
rel	String	self

- Example response

Status code: 200

Successful operation

```
{
  "version": {
    "id": "v1.0",
    "links": [
      {
        "rel": "self",
        "href": "https://API URL/v1.0"
      }
    ]
  },
  "version": "",
  "status": "CURRENT",
  "updated": "2016-12-09T00:00:00Z",
  "min_version": ""
}
```

Status Codes

See [Status Codes](#).

Error Codes

See [Error Codes](#).

5.2 Predefined Tag Operations

5.2.1 Creating or Deleting Predefined Tags

Function

This API is used to create or delete predefined tags. You can add tags to resources using the predefined tags.

This API supports idempotency and batch processing.

NOTE

Idempotent operations refer to invoking the same API for multiple times by using the same parameters, which have the same impact on the system.

URI

POST /v1.0/predefine_tags/action

Request

- Parameter description

Table 5-8 Parameters

Name	Mandatory	Type	Description
action	Yes	String	Specifies the action identifier. The value is case sensitive and can be create or delete .
tags	Yes	Array of objects	Specifies the tags. For details, see Table 5-9 .

- **tags** field description

Table 5-9 Fields

Name	Mandatory	Type	Description
key	Yes	String	Specifies the key. A tag key can contain up to 36 characters. Only A-Z, a-z, 0-9, hyphens (-), underscores (_), and Unicode characters (\u4E00-\u9FFF) are allowed.
value	Yes	String	Specifies the value. A tag value can contain up to 43 characters and can be an empty string. Only A-Z, a-z, 0-9, periods (.), hyphens (-), underscores (_), and Unicode characters (\u4E00-\u9FFF) are allowed.

- Example request

```
POST https://{TMS endpoint}/v1.0/predefine_tags/action
{
  "action": "create",
  "tags": [
    {
      "key": "ENV1",
      "value": "DEV1"
    },
    {
      "key": "ENV2",
      "value": "DEV2"
    }
  ]
}
```


Example Response

Status code: 200

Successful operation

Status Codes

See [Status Codes](#).

Error Codes

See [Error Codes](#).

5.2.2 Querying Predefined Tags

Function

This API is used to query predefined tags.

URI

GET /v1.0/predefine_tags

Request

- Parameter description

Table 5-10 Parameters

Name	Mandator y	Type	Description
key	No	String	Specifies the key. Supports fuzzy search and is case insensitive. If this parameter value contains non-URL-safe characters, it must be URL encoded.
value	No	String	Specifies the value. Supports fuzzy search and is case insensitive. If this parameter value contains non-URL-safe characters, it must be URL encoded.
limit	No	Integer	Specifies the number of query records. The value ranges from 1 to 1000 . If no value is specified, the value is 10 by default. If the value is set to 0 , the number of query records is not limited.

Name	Mandatory	Type	Description
marker	No	String	<p>Specifies the paging location identifier (index).</p> <p>The query starts from the next piece of data indexed by this parameter.</p> <p>NOTE When querying the data on the first page, you do not need to specify this parameter. When querying the data on subsequent pages, set this parameter to the value in the response body returned by querying data of the previous page. When the returned tags is an empty list, the last page has been queried.</p>
order_field	No	String	<p>Specifies the field for sorting.</p> <p>The parameter value is case sensitive and can be update_time, key, or value.</p> <p>Its default value is update_time.</p> <p>You can choose only one of the three values and based on the value of order_method to sort the remaining two default fields.</p> <p>For example:</p> <ul style="list-style-type: none"> • If order_field is set to update_time, both key and value are sorted in the ascending order. • If order_field is set to key, update_time is sorted in the descending order, and value is sorted in the ascending order. • If order_field is set to value, update_time is sorted in the descending order, and key is sorted in the ascending order. • If order_field is not specified, its default value update_time is taken. In this case, key and value are sorted in the ascending order.

Name	Mandator y	Type	Description
order_method	No	String	Specifies the sorting method of the order_field field. The method can be (case sensitive): <ul style="list-style-type: none"> • asc: ascending order • desc: descending order Only one of the preceding sorting methods can be selected. If this parameter is not specified, the default value is desc .

- Example request
GET https://{TMS endpoint}/v1.0/predefine_tags?key=ENV&value=DEV&limit=10&marker=9&order_field=key&order_method=asc

Response

- Parameter description

Table 5-11 Parameters

Name	Type	Description
tags	Array of objects	Specifies the tags. For details, see Table 5-12 .
total_count	Integer	Specifies the total number of tags that meet the filtering criteria, which is not affected by pagination.
marker	String	Specifies the paging location identifier. It indicates the location of the last query record.

- **tags** field description

Table 5-12 Fields

Name	Type	Description
key	String	Specifies the key. A tag key can contain up to 36 characters. Only A-Z, a-z, 0-9, hyphens (-), underscores (_), and Unicode characters (\u4E00-\u9FFF) are allowed.

Name	Type	Description
value	String	Specifies the value. A tag value can contain up to 43 characters and can be an empty string. Only A-Z, a-z, 0-9, periods (.), hyphens (-), underscores (_), and Unicode characters (\u4E00-\u9FFF) are allowed.
update_time	String	Specifies the update time, which must be the UTC time, for example, 2016-12-09T00:00:00Z .

- Example response

Status code: 200

Successful operation

```
{
  "marker": "12",
  "total_count": 13,
  "tags": [
    {
      "key": "ENV1",
      "value": "DEV1",
      "update_time": "2017-04-12T14:22:34Z"
    },
    {
      "key": "ENV2",
      "value": "DEV2",
      "update_time": "2017-04-12T14:22:34Z"
    }
  ]
}
```

Status Codes

See [Status Codes](#).

Error Codes

See [Error Codes](#).

5.2.3 Modifying Predefined Tags

Function

This API is used for modifying predefined tags.

URI

PUT /v1.0/predefine_tags

Request

- Parameter description

Table 5-13 Parameters

Name	Mandatory	Type	Description
old_tag	Yes	Object	Specifies the tag to be modified. For details, see Table 5-14 .
new_tag	Yes	Object	Specifies the tag that has been modified. For details, see Table 5-15 .

- old_tag** field description

Table 5-14 Fields

Name	Mandatory	Type	Description
key	Yes	String	Specifies the key. A tag key can contain up to 36 characters. Only A-Z, a-z, 0-9, hyphens (-), underscores (_), and Unicode characters (\u4E00-\u9FFF) are allowed.
value	Yes	String	Specifies the value. A tag value can contain up to 43 characters and can be an empty string. Only A-Z, a-z, 0-9, periods (.), hyphens (-), underscores (_), and Unicode characters (\u4E00-\u9FFF) are allowed.

- new_tag** field description

Table 5-15 Fields

Name	Mandatory	Type	Description
key	Yes	String	Specifies the key. A tag key can contain up to 36 characters. Only A-Z, a-z, 0-9, hyphens (-), underscores (_), and Unicode characters (\u4E00-\u9FFF) are allowed.

Name	Mandatory	Type	Description
value	Yes	String	Specifies the value. A tag value can contain up to 43 characters and can be an empty string. Only A-Z, a-z, 0-9, periods (.), hyphens (-), underscores (_), and Unicode characters (\u4E00-\u9FFF) are allowed.

- Example request
PUT `https://{TMS endpoint}/v1.0/predefined_tags`

```
{
  "new_tag": {
    "key": "ENV1",
    "value": "DEV1"
  },
  "old_tag": {
    "key": "ENV2",
    "value": "DEV2"
  }
}
```

Example Response

Status code: 200

Successful operation

Status Codes

See [Status Codes](#).

Error Codes

See [Error Codes](#).

5.3 Querying Services Supported by TMS

5.3.1 Querying Services Supported by TMS

Function

You can use this API to query services supported by TMS.

URI

GET `/v1.0/tms/providers`

Table 5-16 Query parameters

Parameter	Mandatory	Type	Description
locale	No	String	Specifies the display language.
limit	No	Integer	The maximum queries supported. The value 10 is used by default if this parameter is not set. The value range is 1 to 200.
offset	No	Integer	Specifies the index position, which starts from the next data record specified by offset . The value must be a number and cannot be negative. The default value is 0 .
provider	No	String	Specifies the cloud service name.

Request

Table 5-17 Header parameters

Parameter	Mandatory	Type	Description
X-Auth-Token	Yes	String	Specifies the user token. TMS is a global service. So you need to set scope to domain when calling an IAM API to obtain a user token. The value of X-Subject-Token in the response header is the user token.

Response

Status code: 200

Table 5-18 Body parameters

Parameter	Type	Description
providers	Array of ProviderResponseBody objects	Specifies the cloud services

Parameter	Type	Description
total_count	Integer	Specifies the total cloud services supported.

Table 5-19 ProviderResponseBody

Parameter	Type	Description
provider	String	Specifies the cloud service name.
provider_i18n_display_name	String	Specifies the display name of the resource. You can set the language by setting the locale parameter.
resource_types	Array of ResourceTypeBody objects	Specifies the resource type.

Table 5-20 ResourceTypeBody

Parameter	Type	Description
resource_type	String	Specifies the resource type.
resource_type_i18n_display_name	String	Specifies the display name of the resource type. You can set the language by setting the locale parameter.
regions	Array of strings	Specifies the supported regions.
global	Boolean	Specifies whether the resource is a global resource.

Example Request

Querying supported services by TMS

```
GET https://{Endpoint}/v1.0/tms/providers?locale=en-us&limit=200
```

Example Response

Status code: 200

Successful operation

```
{
  "providers": [ {
    "provider": "evs",
    "provider_i18n_display_name": "Elastic Volume Service",
    "resource_types": {
      "resource_type_i18n_display_name": "EVS-Disk",
      "global": false,

```



```
"resource_type" : "disk",  
"regions" : [ "regionId1" ]  
}  
}],  
"total_count" : 1  
}
```

Status Codes

See [Status Codes](#).

Error Codes

See [Error Codes](#).

6 Permissions Policies and Supported Actions

6.1 Introduction

You can use Identity and Access Management (IAM) to perform fine-grained permissions management for your TMS resources. If your account does not need individual IAM users, you can skip this section.

By default, new IAM users do not have permissions assigned. You need to add a user to one or more groups, and attach permissions policies or roles to these groups. Users inherit permissions from the groups to which they are added and can perform specified operations on cloud services based on the permissions.

You can grant users permissions using roles and policies. Roles are a type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. Policies define API-based permissions for operations on specific resources under certain conditions, allowing for more fine-grained, secure access control of cloud resources.

NOTE

Policy-based authorization is useful if you want to allow or deny the access to an API.

An account has all of the permissions required to call all APIs, but IAM users must be assigned the required permissions. The permissions required for calling an API are determined by the actions supported by the API. Only users who have been granted permissions allowing the actions can call the API successfully. For example, if an IAM user wants to query predefined tags using an API, the user must have been granted permissions that allow the **tms:predefineTags:list** action.

Supported Actions

Operations supported by a fine-grained policy are specific to APIs. The following are common concepts related to policies:

- Permissions: Statements in a policy that allow or deny certain operations.
- APIs: REST APIs that can be called by a user who has been granted specific permissions.

- **Actions:** Specific operations that are allowed or denied.
- **Dependencies:** actions which a specific action depends on. When allowing an action for a user, you also need to allow any existing action dependencies for that user.
- **IAM or enterprise projects:** Type of projects for which an action will take effect. Policies that contain actions supporting both IAM and enterprise projects can be used and take effect in both IAM and Enterprise Management. Policies that only contain actions for IAM projects can be used and only take effect for IAM. Administrators can check whether an action supports IAM projects or enterprise projects in the action list.

6.2 TMS API Actions

Table 6-1 API actions

Permission	API	Action	IAM Project	Enterprise Project
Querying predefined tags	GET /v1.0/predefine_tags	tms:predefineTags:list	Supported	Not supported
Creating predefined tags	POST /v1.0/predefine_tags/action	tms:predefineTags:create	Supported	Not supported
Deleting predefined tags	POST /v1.0/predefine_tags/action	tms:predefineTags:delete	Supported	Not supported
Modifying a predefined tag	PUT /v1.0/predefine_tags	tms:predefineTags:update	Supported	Not supported

A Appendix

A.1 Status Codes

- Normal

Returned Value	Description
200 OK	The results of GET and PUT operations are returned as expected.
201 Created	The results of the POST operation are returned as expected.
202 Accepted	The request has been accepted for processing.
204 No Content	Normal response code

- Abnormal

Returned Value	Description
400 Bad Request	The server failed to process the request.
401 Unauthorized	You must enter a username and password to access the requested page.
403 Forbidden	Access to the requested page is denied.
404 Not Found	The server cannot find the requested page.
405 Method Not Allowed	You are not allowed to use the method specified in the request.
406 Not Acceptable	The response generated by the server cannot be accepted by the client.

Returned Value	Description
407 Proxy Authentication Required	You must use the proxy server for authentication so that the request can be processed.
408 Request Timeout	The request timed out.
409 Conflict	The request could not be processed due to a conflict.
500 Internal Server Error	Failed to complete the request because of a service error.
501 Not Implemented	Failed to complete the request because the server does not support the requested function.
502 Bad Gateway	Failed to complete the request because the request is invalid.
503 Service Unavailable	Failed to complete the request. The service is unavailable.
504 Gateway Timeout	A gateway timeout error occurred.

A.2 Error Codes

Function Description

If the returned status code of a TMS API is **400**, the customized error information will also be returned. This section describes the meaning of each TMS error code.

Response Format

```
STATUS CODE 400
{
  "error_code": "TMS.0009",
  "error_msg": "Key is invalid."
}
```

Error Code Description

Statu s Code	Error Code	Error Message	Description	Solution
403	TMS.20 30	You do not have the required permission to perform this operation. The required permission is: <i>xxx</i> .	Required permissions are not granted.	Grant required permissions.
400	TMS.20 09	Projects is empty.	The Projects parameter is empty.	Specify Projects .
400	TMS.20 17	Invalid element in projects .	Invalid projects element	Enter a valid projects value.
400	APIGW. 0106	Orchestration error.	There is orchestration error.	Check whether the frontend and backend parameters are properly set for the API.
400	TMS.50 27	Invalid resource type .	Invalid resource type	Enter a valid resource type.
400	TMS.20 11	Project_id is invalid.	Invalid project ID	Enter a valid project ID.
500	TMS.00 01	System error.	System error.	Contact technical support.
400	TMS.00 02	Bad request.	Invalid request from the client.	Enter valid parameters.
401	TMS.00 03	The user is unauthorized.	Authentication fails or the valid authentication information is not provided.	Check whether the username or password for obtaining the token is correct.
403	TMS.00 04	You do not have permissions to perform the operation.	The authentication information is incorrect or the service invoker does not have sufficient permissions.	Check whether the username, password, or the user permissions for obtaining the token are correct.

Statu s Code	Error Code	Error Message	Description	Solution
404	TMS.00 05	The resources requested cannot be found.	The requested resource cannot be found.	Enter a valid resource ID.
403	TMS.00 06	The request is Too much , try again later.	The numbers of requests are too many.	Reduce the number of concurrent requests or try again later.
400	TMS.00 07	Limit is invalid.	Limit is invalid.	Enter a valid Limit value.
400	TMS.00 08	Marker is invalid.	Marker is invalid.	Enter a valid Marker value.
400	TMS.00 09	Key is invalid.	Key is invalid.	Enter a valid Key value.
400	TMS.00 10	Value is invalid.	Value is invalid.	Enter a valid Value value.
400	TMS.00 11	Action is invalid.	Action is invalid.	Enter a valid Action value.
400	TMS.00 12	Tags is empty.	Tags is left blank.	Specify Tags .
400	TMS.00 13	Empty element in Tags .	Tags is invalid.	Enter a valid Tags value.
409	TMS.00 14	Values is empty.	Values is invalid.	Enter a valid Values value.
400	TMS.00 16	Values is too much.	The maximum number of values for Values has been exceeded.	Enter no more than 10 values.
400	TMS.00 17	Offset is invalid.	Offset is invalid.	Enter a valid Offset value.
504	TMS.00 18	Query Time Out.	Query timed out.	Try again later.
400	TMS.10 01	The number of predefine tag exceeds the upper limit.	The number of predefined tags exceeds the quota.	Enter no more than 500 predefined tags.
400	TMS.10 02	Old_tag cannot be found.	Old_tag cannot be found.	Specify Old_tag .

Statu s Code	Error Code	Error Message	Description	Solution
400	TMS.10 03	New_tag already exists.	New_tag already exists.	Enter another value for New_tag .
400	TMS.10 04	Old_tag is empty.	Old_tag is left blank.	Specify Old_tag .
400	TMS.10 05	Invalid key in Old_tag .	The key in Old_tag is invalid.	Enter a valid key in Old_tag .
400	TMS.10 06	Invalid value in Old_tag .	The value in Old_tag is invalid.	Enter a valid value in Old_tag .
400	TMS.10 07	New_tag is empty.	New_tag is left blank.	Specify New_tag .
400	TMS.10 08	Invalid key in New_tag .	The key in New_tag is invalid.	Enter a valid key in New_tag .
400	TMS.10 09	Invalid value in New_tag .	The value in New_tag is invalid.	Enter a valid value in New_tag .
400	TMS.10 10	Order_field is invalid.	Order_field is invalid.	Enter a valid Order_field value.
400	TMS.10 11	Order_method is invalid.	Order_method is invalid.	Enter a valid Order_method value.

A.3 Obtaining a Project ID

Scenarios

A project ID is required for some URLs when an API is called. Therefore, you need to obtain a project ID in advance. Two methods are available:

- [Obtain the Project ID by Calling an API](#)
- [Obtain the Project ID from the Console](#)

Obtain the Project ID by Calling an API

You can obtain the project ID by calling the IAM API used to query project information based on the specified criteria.

The API used to obtain a project ID is GET <https://{Endpoint}/v3/projects>. {Endpoint} is the IAM endpoint and can be obtained from [Regions and Endpoints](#). For details about API authentication, see [Authentication](#).

The following is an example response. The value of **id** is the project ID.

```
{
  "projects": [
    {
      "domain_id": "65ewtrgaggshhk1223245sghjlse684b",
      "is_domain": false,
      "parent_id": "65ewtrgaggshhk1223245sghjlse684b",
      "name": "project_name",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4adasfjljaaakla12334jklga9sasfg"
      },
      "id": "a4adasfjljaaakla12334jklga9sasfg",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

Obtain a Project ID from the Console

To obtain a project ID from the console, perform the following operations:

1. Log in to the management console.
2. Click the username and select **My Credentials** from the drop-down list.

On the **My Credentials** page, view the project ID (value in the **Project ID** column).

B Change History

Release On	Description
2022-10-31	This issue is the second official release, which incorporates the following changes: <ul style="list-style-type: none">• Added Introduction.• Added TMS API Actions.
2020-07-30	This issue is the first official release.