

**Anti-DDoS**  
**23.9.0**

# **API Reference**

**Issue**            01  
**Date**             2024-04-17



**Copyright © Huawei Cloud Computing 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 Cloud Computing Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are the property 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 Cloud 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.

## **Huawei Cloud Computing Technologies Co., Ltd.**

Address: Huawei Cloud Data Center Jiaoxinggong Road  
Qianzhong Avenue  
Gui'an New District  
Gui Zhou 550029  
People's Republic of China

Website: <https://www.huaweicloud.com/intl/en-us/>

---

# Contents

---

<b>1 Before You Start.....</b>	<b>1</b>
1.1 Overview.....	1
1.2 API Calling.....	1
1.3 Endpoints.....	1
1.4 Notes and Constraints.....	1
1.5 Concepts.....	2
<b>2 API Overview.....</b>	<b>3</b>
<b>3 API Calling.....</b>	<b>4</b>
3.1 Making an API Request.....	4
3.2 Authentication.....	6
3.3 Response.....	11
<b>4 API.....</b>	<b>13</b>
4.1 Anti-DDoS APIs.....	13
4.1.1 Querying Optional Anti-DDoS Defense Policies.....	13
4.1.2 Enabling Anti-DDoS.....	18
4.1.3 Querying Configured Anti-DDoS Defense Policies.....	19
4.1.4 Updating Anti-DDoS Defense Policies.....	21
4.1.5 Querying Anti-DDoS Tasks.....	22
4.1.6 Querying the List of Defense Statuses of EIPs.....	24
4.1.7 Querying the Defense Status of a Specified EIP.....	26
4.1.8 Querying the Traffic of a Specified EIP.....	28
4.1.9 Querying Events of a Specified EIP.....	29
4.1.10 Querying Weekly Defense Statistics.....	32
4.2 Alarm Reminding APIs.....	34
4.2.1 Querying Alarm Configuration.....	34
4.2.2 Updating Alarm Configuration.....	36
<b>A Appendix.....</b>	<b>39</b>
A.1 Status Code.....	39
A.2 Error Codes.....	40
A.3 Obtaining a Project ID.....	42

---

**B Change History..... 43**

# 1 Before You Start

---

## 1.1 Overview

The Anti-DDoS service protects public IP addresses against Layer 4 to Layer 7 distributed denial of service (DDoS) attacks and sends alarms immediately when detecting an attack. Anti-DDoS improves the bandwidth utilization and ensures the stable running of user services.

Anti-DDoS monitors the service traffic from the Internet to public IP addresses and detects attack traffic in real time. It then scrubs attack traffic based on user-configured defense policies without interrupting service running. It also generates monitoring reports that provide visibility into the network traffic security.

This document describes how to use application programming interfaces (APIs) to perform operations to Anti-DDoS, such as querying or updating Anti-DDoS protection policy. For details about all supported operations, see [API Overview](#).

Before calling Anti-DDoS APIs, ensure that you are familiar with Anti-DDoS concepts. For details, see section "Service Overview" of the *Anti-DDoS User Guide*.

## 1.2 API Calling

Anti-DDoS provides Representational State Transfer (REST) APIs, allowing you to use HTTPS requests to call them. For details, see [API Calling](#).

## 1.3 Endpoints

An endpoint is the **request address** for calling an API. Endpoints vary depending on services and regions. Obtain the regions and endpoints from the enterprise administrator.

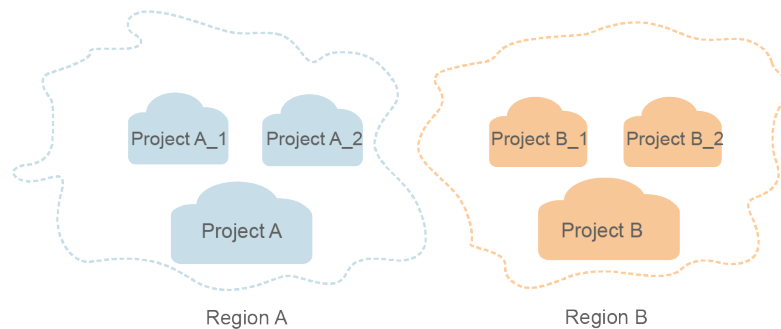
## 1.4 Notes and Constraints

For details about the constraints, see the API description.

## 1.5 Concepts

- **Region**  
Regions are divided based on geographical location and network latency. Public services, such as Elastic Cloud Server (ECS), Elastic Volume Service (EVS), Object Storage Service (OBS), Virtual Private Cloud (VPC), Elastic IP (EIP), and Image Management Service (IMS), are shared within the same region. Regions are classified as universal regions and dedicated regions. A universal region provides universal cloud services for common tenants. A dedicated region provides services of the same type only or for specific tenants.
- **Availability Zone (AZ)**  
An AZ comprises one or multiple physical data centers equipped with independent ventilation, fire, water, and electricity facilities. Compute, 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 support cross-AZ high-availability systems.
- **Project**  
Projects group and isolate compute, storage, and network resources across physical regions. A default project is provided for each region, and subprojects can be created under each default project. Users can be granted permissions to access all resources in a specific project. For more refined access control, create subprojects under a project and create resources in the subprojects. Users can then be assigned permissions to access only specific resources in the subprojects.

**Figure 1-1** Project isolation model



# 2 API Overview

---

You can use all functions of Anti-DDoS through its APIs.

Type	Description
APIs of Anti-DDoS	Anti-DDoS APIs, including the APIs for querying and updating the Anti-DDoS defense policies.
Alarm Reminding APIs	Alarm reminding APIs, including the API for querying alarm configuration information

# 3 API Calling

---

## 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.

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

#### 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:

- **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.
- **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://{{endpoint}}/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:

- **Content-Type**: specifies the request body type or format. This field is mandatory and its default value is **application/json**. Other values of this field will be provided for specific APIs if any.
- **X-Auth-Token**: specifies a user token only for token-based API authentication. The user token is a response to the API used to obtain a user token. This API is the only one that does not require authentication.

### NOTE

In addition to supporting token-based authentication, APIs also support authentication using access key ID/secret access key (AK/SK). During AK/SK-based authentication, an SDK is used to sign the request, and the **Authorization** (signature information) and **X-Sdk-Date** (time when the request is sent) header fields are automatically added to the request.

For more information, see [AK/SK-based 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://{{endpoint}}/v3/auth/tokens  
Content-Type: application/json
```

## Request Body

The body of a request is often sent in a structured format 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. Set **username** to the name of a user, **domainname** to the name of the account that the user belongs to, **\*\*\*\*\*** to the user's login password, and **xxxxxxxxxxxxxxxxxxxx** to the project name. You can learn more information about projects from the administrator.

 **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://{{endpoint}}/v3/auth/tokens
Content-Type: application/json
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

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-based authentication: Requests are authenticated using a token.
- AK/SK-based authentication: Requests are authenticated by encrypting the request body using an AK/SK pair. This method is recommended because it provides higher security than token-based authentication.

## Token-based 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.

The token can be obtained by calling the required API. For more information, see Obtaining a User Token. A project-level token is required for calling this API, that is, **auth.scope** must be set to **project** in the request body. Example:

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}
```

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://{endpoint}/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

## AK/SK-based Authentication

An AK/SK is used to verify the identity of a request sender. In AK/SK-based authentication, a signature needs to be obtained and then added to requests.

### NOTE

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 used in conjunction with an AK to sign requests cryptographically. It identifies a request sender and prevents the request from being modified.

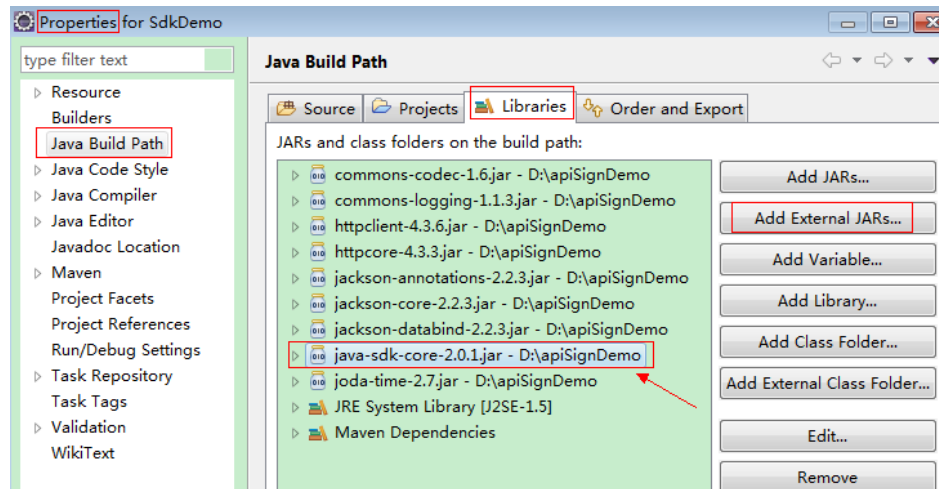
The following uses a demo project to show how to sign a request and use an HTTP client to send an HTTPS request.

Download the demo project at <https://github.com/api-gate-way/SdkDemo>.

If you do not need the demo project, obtain the API Gateway signing SDK from the enterprise administrator.

Decompress the downloaded package and reference the obtained JAR files as dependencies.

**Figure 3-1** Importing a JAR file



**Step 1** Generate an AK/SK. If an AK/SK pair is already available, skip this step and go to **Step 2**. Find the downloaded AK/SK file, which is usually named **credentials.csv**.

1. Register an account and log in to the management console.
2. Click the username and choose **My Credential** from the drop-down list.
3. On the **My Credentials** page, choose **Access Keys** in the navigation pane.
4. Click **Add Access Key**.

**NOTE**

For users created in IAM that have not bound any email address or mobile number, only the login password needs to be entered.

5. Click **OK**. Download the access key after it is created.

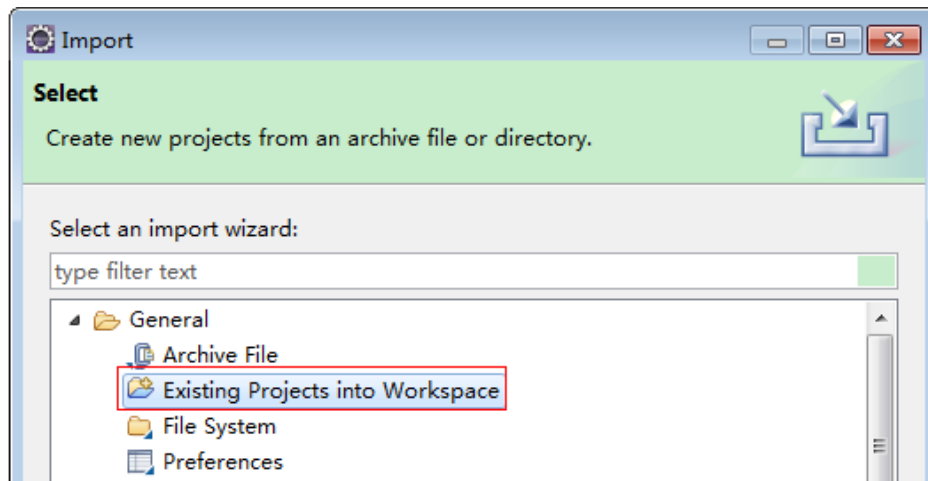
**NOTE**

Keep the access key secure.

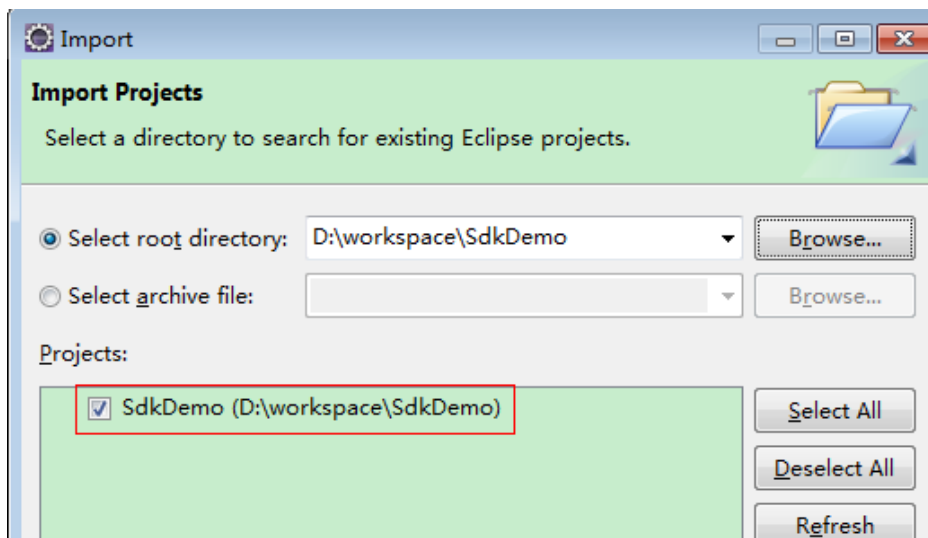
**Step 2** Download and decompress the demo project.

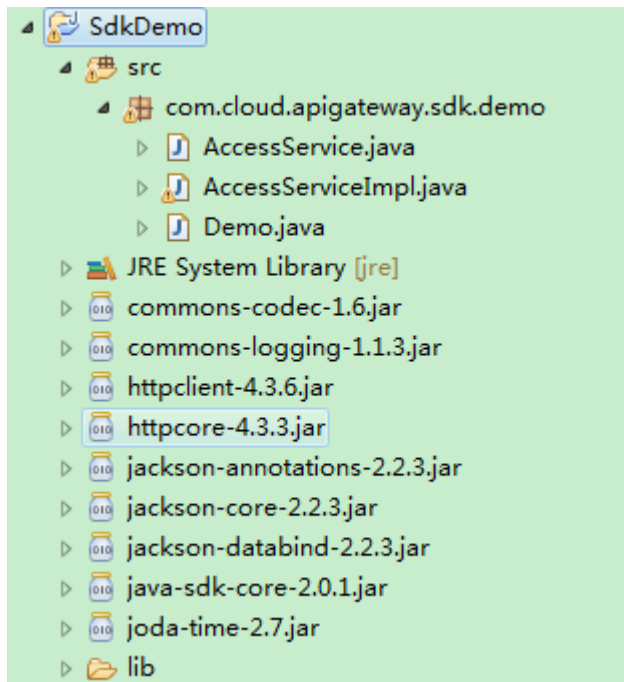
**Step 3** Import the demo project to Eclipse.

**Figure 3-2** Selecting Existing Projects into Workspace



**Figure 3-3** Selecting the demo project



**Figure 3-4** Structure of the demo project**Step 4** Sign the request.

The request signing method is integrated in the JAR files imported in [Step 3](#). The request needs to be signed before it is sent. The signature will then be added as part of the HTTP header to the request.

The demo code is classified into the following classes to demonstrate signing and sending the HTTP request:

- **AccessService**: An abstract class that merges the GET, POST, PUT, and DELETE methods into the access method.
- **Demo**: Execution entry used to simulate the sending of GET, POST, PUT, and DELETE requests.
- **AccessServiceImpl**: Implements the access method, which contains the code required for communication with API Gateway.

1. Edit the main( ) method in the **Demo.java** file, and replace the bold text with actual values.

As shown in the following code, if you use other methods such as POST, PUT, and DELETE, see the corresponding comment.

Specify **region**, **serviceName**, **ak/sk**, and **url** as the actual values. In this demo, the URLs for accessing VPC resources are used.

To obtain the project ID in the URLs, see [Obtaining a Project ID](#). To obtain the endpoint, contact the enterprise administrator.

```
//TODO: Replace region with the name of the region in which the service to be accessed is located.  
private static final String region = "";  
  
//TODO: Replace vpc with the name of the service you want to access. For example, ecs, vpc, iam,  
and elb.  
private static final String serviceName = "";  
  
public static void main(String[] args) throws UnsupportedEncodingException  
{
```

```
//TODO: Replace the AK and SK with those obtained on the My Credential page.
String ak = System.getenv("CLOUD_SDK_AK")
String sk = System.getenv("CLOUD_SDK_SK")

//TODO: To specify a project ID (multi-project scenarios), add the X-Project-Id header.
//TODO: To access a global service, such as IAM, DNS, CDN, and TMS, add the X-Domain-Id header to
specify an account ID.
//TODO: To add a header, find "Add special headers" in the AccessServiceImple.java file.

//TODO: Test the API
String url = "https://{Endpoint}/v1/{project_id}/vpcs";
get(ak, sk, url);

//TODO: When creating a VPC, replace {project_id} in postUrl with the actual value.
//String postUrl = "https://serviceEndpoint/v1/{project_id}/cloudservers";
//String postbody = "{\"vpc\": {\"name\": \"vpc\", \"cidr\": \"192.168.0.0/16\"}}";
//post(ak, sk, postUrl, postbody);

//TODO: When querying a VPC, replace {project_id} in url with the actual value.
//String url = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//get(ak, sk, url);

//TODO: When updating a VPC, replace {project_id} and {vpc_id} in putUrl with the actual values.
//String putUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//String putbody = "{\"vpc\": {\"name\": \"vpc1\", \"cidr\": \"192.168.0.0/16\"}}";
//put(ak, sk, putUrl, putbody);

//TODO: When deleting a VPC, replace {project_id} and {vpc_id} in deleteUrl with the actual values.
//String deleteUrl = "https://serviceEndpoint/v1/{project_id}/vpcs/{vpc_id}";
//delete(ak, sk, deleteUrl);
}
```

2. Compile the code and call the API.

In the **Package Explorer** area on the left, right-click **Demo.java**, choose **Run AS > Java Application** from the shortcut menu to run the demo code.

You can view the API call logs on the console.

----End

## 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 Code](#).

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**.

The following shows the response header for the API to obtain a user token, in which **x-subject-token** is the desired user token. This token can then be used to authenticate the calling of other APIs.

**Figure 3-5** 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 → MIIVXQYJKoZIhvcNAQcCoIIVTJCGEoCAQExDTALBgIghkgBZQMEAgEwgharBgkqhkiG9w0BBwGgghacBIIWmHsidG9rZW4iOansiZXhwaXJlc19hdCI6IiwMTktMDItMTNUMD
fj3KJs6YgKnpVNRbW2eZ5eb78SZOkajACgkIQ01wi4JIGzrpd18LGXK5bdfq4lqHCYb8P4NaYONYeJcAgzVVeFYtLWT1GSO0zxKZmlQHq82HBqHdgIZO9fuEbl5dMhdavj+33wEl
xHRCE9I87o+k9-
j+CMZSEB7bUGd5Uj6eRASXl1jipPEGA270g1FruooL6jggIFkNPQuFSOU8+uSsttVwRtnfsc+qTp22Rkd5MCqFGQ8LcuUxC3a+9CMBnOintWW7oeRUUVhVpxk8pxiX1wTEboX-
RzT6MUbpvGw-oPNFYxJECKnoH3HRozv0vN--n5d6Nbxg==
x-xss-protection → 1; mode=block;
```

### (Optional) Response Body

The body of a response is often returned in structured format as specified in the **Content-Type** header field. The response body transfers content except the response header.

The following shows part of the response body for the API to obtain a user token. For the sake of space, only part of the content is displayed here.

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

```

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 format of message is error",
  "error_code": "AS.0001"
}
```

In the response body, **error\_code** is an error code, and **error\_msg** provides information about the error.



# 4 API

## 4.1 Anti-DDoS APIs

### 4.1.1 Querying Optional Anti-DDoS Defense Policies

#### Functions

This API allows you to query optional Anti-DDoS defense policies. Based on your service, you can select a policy for Anti-DDoS traffic scrubbing.

#### URI

- URI format  
GET /v1/{project\_id}/antiddos/query\_config\_list
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID

#### Request

##### Request parameters

None

#### Response

- Parameter description

Parameter	Mandatory	Type	Description
traffic_limited_list	Yes	List data structure	List of traffic limits
http_limited_list	Yes	List data structure	List of HTTP limits
connection_limited_list	Yes	List data structure	List of limits of numbers of connections

- Data structure description of **traffic\_limited\_list**

Parameter	Mandatory	Type	Description
traffic_pos_id	Yes	Integer	Position ID of traffic
traffic_per_second	Yes	Integer	Threshold of traffic per second (Mbit/s)
packet_per_second	Yes	Integer	Threshold of number of packets per second

- Data structure description of **http\_limited\_list**

Parameter	Mandatory	Type	Description
http_request_pos_id	Yes	Integer	Position ID of number of HTTP requests
http_packet_per_second	Yes	Integer	Threshold of number of HTTP requests per second

- Data structure description of **connection\_limited\_list**

Parameter	Mandatory	Type	Description
cleaning_access_pos_id	Yes	Integer	Position ID of access limit during cleaning
new_connection_limited	Yes	Integer	Number of new connections of a source IP address
total_connection_limited	Yes	Integer	Total number of connections of a source IP address

## Example

- Example request  
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/query\_config\_list

- Example response

```
{
  "traffic_limited_list": [
    {
      "traffic_pos_id": 1,
      "traffic_per_second": 10,
      "packet_per_second": 2000
    },
    {
      "traffic_pos_id": 2,
      "traffic_per_second": 30,
      "packet_per_second": 6000
    },
    {
      "traffic_pos_id": 3,
      "traffic_per_second": 50,
      "packet_per_second": 10000
    },
    {
      "traffic_pos_id": 4,
      "traffic_per_second": 70,
      "packet_per_second": 15000
    },
    {
      "traffic_pos_id": 5,
      "traffic_per_second": 100,
      "packet_per_second": 20000
    },
    {
      "traffic_pos_id": 6,
      "traffic_per_second": 150,
      "packet_per_second": 25000
    },
    {
      "traffic_pos_id": 7,
      "traffic_per_second": 200,
      "packet_per_second": 35000
    },
    {
      "traffic_pos_id": 8,
      "traffic_per_second": 250,
      "packet_per_second": 50000
    },
    {
      "traffic_pos_id": 9,
      "traffic_per_second": 300,
      "packet_per_second": 70000
    }
  ],
  "http_limited_list": [
    {
      "http_request_pos_id": 1,
      "http_packet_per_second": 100
    },
    {
      "http_request_pos_id": 2,
      "http_packet_per_second": 150
    },
    {
      "http_request_pos_id": 3,
      "http_packet_per_second": 240
    },
    {
      "http_request_pos_id": 4,
```

```
"http_packet_per_second": 350
},
{
  "http_request_pos_id": 5,
  "http_packet_per_second": 480
},
{
  "http_request_pos_id": 6,
  "http_packet_per_second": 550
},
{
  "http_request_pos_id": 7,
  "http_packet_per_second": 700
},
{
  "http_request_pos_id": 8,
  "http_packet_per_second": 850
},
{
  "http_request_pos_id": 9,
  "http_packet_per_second": 1000
},
{
  "http_request_pos_id": 10,
  "http_packet_per_second": 1500
},
{
  "http_request_pos_id": 11,
  "http_packet_per_second": 2000
},
{
  "http_request_pos_id": 12,
  "http_packet_per_second": 3000
},
{
  "http_request_pos_id": 13,
  "http_packet_per_second": 5000
},
{
  "http_request_pos_id": 14,
  "http_packet_per_second": 10000
},
{
  "http_request_pos_id": 15,
  "http_packet_per_second": 20000
}
],
"connection_limited_list": [
  {
    "cleaning_access_pos_id": 1,
    "new_connection_limited": 10,
    "total_connection_limited": 30
  },
  {
    "cleaning_access_pos_id": 2,
    "new_connection_limited": 20,
    "total_connection_limited": 100
  },
  {
    "cleaning_access_pos_id": 3,
    "new_connection_limited": 30,
    "total_connection_limited": 200
  },
  {
    "cleaning_access_pos_id": 4,
    "new_connection_limited": 40,
    "total_connection_limited": 250
  },
  {
```

```
    "cleaning_access_pos_id": 5,  
    "new_connection_limited": 50,  
    "total_connection_limited": 300  
  },  
  {  
    "cleaning_access_pos_id": 6,  
    "new_connection_limited": 60,  
    "total_connection_limited": 500  
  },  
  {  
    "cleaning_access_pos_id": 7,  
    "new_connection_limited": 70,  
    "total_connection_limited": 600  
  },  
  {  
    "cleaning_access_pos_id": 8,  
    "new_connection_limited": 80,  
    "total_connection_limited": 700  
  }  
],  
"extend_ddos_config": [  
  {  
    "new_connection_limited": 80,  
    "total_connection_limited": 700,  
    "http_packet_per_second": 500000,  
    "traffic_per_second": 1000,  
    "packet_per_second": 200000,  
    "setID": 33  
  },  
  {  
    "new_connection_limited": 80,  
    "total_connection_limited": 700,  
    "http_packet_per_second": 500000,  
    "traffic_per_second": 2000,  
    "packet_per_second": 200000,  
    "setID": 34  
  },  
  {  
    "new_connection_limited": 80,  
    "total_connection_limited": 700,  
    "http_packet_per_second": 500000,  
    "traffic_per_second": 5000,  
    "packet_per_second": 400000,  
    "setID": 35  
  },  
  {  
    "new_connection_limited": 80,  
    "total_connection_limited": 700,  
    "http_packet_per_second": 0,  
    "traffic_per_second": 0,  
    "packet_per_second": 0,  
    "setID": 36  
  }  
]  
}
```

#### NOTE

The **extend\_ddos\_config** field displays information about Anti-DDoS defense policies set by users based on their needs.

## Status Code

See [Status Code](#).

## 4.1.2 Enabling Anti-DDoS

### Functions

This asynchronous API allows you to enable the Anti-DDoS traffic scrubbing. Successfully invoking this API only means that the service node has received the enabling request. You need to use the task querying API to check the task execution status. For details about the task querying API, see [Querying Anti-DDoS Tasks](#).

### URI

- URI format  
POST /v1/{project\_id}/antiddos/{floating\_ip\_id}
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID
floating_ip_id	Yes	String	ID corresponding to the Elastic IP Address (EIP) of a user

### Request

**Table 4-1** Parameter description

Parameter	Mandatory	Type	Description
enable_L7	No	Boolean	Whether to enable L7 defense
traffic_pos_id	Yes	Integer	Position ID of traffic. The value ranges from 1 to 9.
http_request_pos_id	Yes	Integer	Position ID of number of HTTP requests. The value ranges from 1 to 15.
cleaning_access_pos_id	Yes	Integer	Position ID of access limit during cleaning. The value ranges from 1 to 8.
app_type_id	No	Integer	Application type ID. Possible values: <ul style="list-style-type: none"> <li>• 0</li> <li>• 1</li> </ul>

## Response

**Table 4-2** Parameter description

Name	Type	Description
error_code	String	Internal error code
error_description	String	Internal error description
task_id	String	ID of a task. This ID can be used to query the status of the task. This field is reserved for use in task auditing later. It is temporarily unused.

## Example

- Example request  
POST /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8  

```
{
  "enable_L7":true,
  "traffic_pos_id":1,
  "http_request_pos_id":1,
  "cleaning_access_pos_id":1,
  "app_type_id":1
}
```
- Example response  

```
{
  "error_code": "10000000",
  "error_description": "Task has been received and is being processed.",
  "task_id": "94e17e18-5b2c-40c6-a218-8ec5134e32a5"
}
```

## Status Code

See [Status Code](#).

## 4.1.3 Querying Configured Anti-DDoS Defense Policies

### Functions

This API enables you to query the Anti-DDoS defense policy of a specified EIP.

### URI

- URI format  
GET /v1/{project\_id}/antiddos/{floating\_ip\_id}
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

## Request

### Request parameters

None

## Response

**Table 4-3** Parameter description

Parameter	Mandatory	Type	Description
enable_L7	Yes	Boolean	Whether L7 defense has been enabled.
traffic_pos_id	Yes	Integer	Position ID of traffic. The value ranges from 1 to 9, or 99.
http_request_pos_id	Yes	Integer	Position ID of number of HTTP requests. The value ranges from 1 to 15.
cleaning_access_pos_id	Yes	Integer	Position ID of access limit during cleaning. The value ranges from 1 to 8, or 99.
app_type_id	Yes	Integer	Application type ID. Possible values: <ul style="list-style-type: none"> <li>• 0</li> <li>• 1</li> </ul>

## Example

- Example request  
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8
- Example response

```
{
  "enable_L7": true,
  "traffic_pos_id": 1,
  "http_request_pos_id": 1,
  "cleaning_access_pos_id": 1,
  "app_type_id": 1
}
```



## Status Code

See [Status Code](#).

## 4.1.4 Updating Anti-DDoS Defense Policies

### Functions

This API enables you to update the Anti-DDoS defense policy of a specified EIP. Successfully invoking this API only means that the service node has received the update request. You need to use the task querying API to check the task execution status. For details about the task querying API, see [Querying Anti-DDoS Tasks](#).

### URI

- URI format  
PUT /v1/{project\_id}/antiddos/{floating\_ip\_id}
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

### Request

**Table 4-4** Parameter description

Parameter	Mandatory	Type	Description
enable_L7	No	Boolean	Whether to enable L7 defense
traffic_pos_id	Yes	Integer	Position ID of traffic. The value ranges from 1 to 9, or 99.
http_request_pos_id	Yes	Integer	Position ID of number of HTTP requests. The value ranges from 1 to 15.
cleaning_access_pos_id	Yes	Integer	Position ID of access limit during cleaning. The value ranges from 1 to 8, or 99.
app_type_id	No	Integer	Application type ID. Possible values: <ul style="list-style-type: none"><li>• 0</li><li>• 1</li></ul>

## Response

**Table 4-5** Parameter description

Name	Type	Description
error_code	String	Internal error code
error_description	String	Internal error description
task_id	String	ID of a task. This ID can be used to query the status of the task. This field is reserved for use in task auditing later. It is temporarily unused.

## Example

- Example request  
PUT /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/ee0c854e-082f-499e-b7d8-1b42c22781af  

```
{
  "enable_L7":false,
  "traffic_pos_id":2,
  "http_request_pos_id":1,
  "cleaning_access_pos_id":1,
  "app_type_id":1
}
```
- Example response  

```
{
  "error_code": "10000000",
  "error_description": "Task has been received and is being processed.",
  "task_id": "4a4fefe7-34a1-40e2-a87c-16932af3ac4a"
}
```

## Status Code

See [Status Code](#).

## 4.1.5 Querying Anti-DDoS Tasks

### Functions

This API enables you to query the execution status of a specified Anti-DDoS configuration task.

### URI

- URI format  
GET /v1/{project\_id}/query\_task\_status

#### NOTE

You can use **?** and **&** behind the URI to add query conditions, as shown in the request example.

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID

## Request

**Table 4-6** Parameter description

Parameter	Mandatory	Type	Description
task_id	Yes	String	Task ID (nonnegative integer) character string

## Response

- Parameter description

Name	Type	Description
task_status	String	Status of a task, which can be one of the following: <ul style="list-style-type: none"> <li>success</li> <li>failed</li> <li>waiting</li> <li>running</li> <li>preprocess</li> <li>ready</li> </ul>
task_msg	String	Additional information about a task

## Example

- Example request  
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/query\_task\_status?  
task\_id=4a4fefe7-34a1-40e2-a87c-16932af3ac4a
- Example response  

```
{
  "task_status": "running",
  "task_msg": ""
}
```

## Status Code

See [Status Code](#).

## 4.1.6 Querying the List of Defense Statuses of EIPs

### Functions

This API enables you to query the defense statuses of all EIPs, regardless whether an EIP has been bound to an Elastic Cloud Server (ECS) or not.

### URI

- URI format  
GET /v1/{project\_id}/antiddos

 **NOTE**

You can use **?** and **&** behind the URI to add query conditions, as shown in the request example.

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID

### Request

**Table 4-7** Parameter description

Parameter	Mandatory	Type	Description
status	No	String	Possible values: <ul style="list-style-type: none"> <li><b>normal</b>: indicates that the defense status is normal.</li> <li><b>configuring</b>: indicates that defense is being configured.</li> <li><b>notConfig</b>: indicates that defense is not configured.</li> <li><b>packetcleaning</b>: indicates that traffic cleaning is underway.</li> <li><b>packetdropping</b>: indicates that traffic is discarded.</li> </ul> If this parameter is not used, the defense statuses of all ECSs are displayed in the Neutron-queried order by default.
limit	No	Integer	Maximum number of returned results. The value ranges from 1 to 100.
offset	No	Integer	Offset. The value ranges from 0 to 2147483647.

Parameter	Mandatory	Type	Description
ip	No	String	IP address. Both IPv4 and IPv6 addresses are supported. For example, if you enter <b>?ip=192.168</b> , the defense status of EIPs corresponding to 192.168.111.1 and 10.192.168.8 is returned.

## Response

- Parameter description

Name	Type	Description
total	Integer	Total number of EIPs
ddosStatus	List data structure	List of defense statuses

- Data structure description of **ddosStatus**

Parameter	Mandatory	Type	Description
floating_ip_address	Yes	String	Floating IP address
floating_ip_id	Yes	String	ID of an EIP
network_type	Yes	String	EIP type. The value can be: <ul style="list-style-type: none"><li><b>EIP</b>: EIP that is bound or not bound with ECS.</li><li><b>ELB</b>: EIP that is bound with ELB.</li></ul>
status	Yes	String	Defense status, the possible value of which is one of the following: <ul style="list-style-type: none"><li><b>normal</b>: indicates that the defense status is normal.</li><li><b>configuring</b>: indicates that defense is being configured.</li><li><b>notConfig</b>: indicates that defense is not configured.</li><li><b>packetcleaning</b>: indicates that traffic cleaning is underway.</li><li><b>packetdropping</b>: indicates that traffic is discarded.</li></ul>

## Example

- Example request  
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos?status=packetdropping

- Example response

```
{
  "total": 5,
  "ddosStatus": [
    {
      "floating_ip_id": "1867f954-fc11-4202-8247-6af2144867ea",
      "floating_ip_address": "192.168.42.221",
      "network_type": "EIP",
      "status": "notConfig"
    },
    {
      "floating_ip_id": "49c6af49-9ace-42e6-ab89-1eee1f4ac821",
      "floating_ip_address": "192.168.35.152",
      "network_type": "EIP",
      "status": "normal"
    },
    {
      "floating_ip_id": "7a8dc957-083b-499d-b7cf-6fa48f4880c5",
      "floating_ip_address": "192.168.42.222",
      "network_type": "EIP",
      "status": "notConfig"
    },
    {
      "floating_ip_id": "7c6676a0-b281-4163-9d0d-cb6485ae9860",
      "floating_ip_address": "192.168.44.69",
      "network_type": "EIP",
      "status": "normal"
    },
    {
      "floating_ip_id": "969c1d48-6a92-4ef1-b66c-b17c7e7d7ce7",
      "floating_ip_address": "192.168.47.192",
      "network_type": "EIP",
      "status": "notConfig"
    }
  ]
}
```

## Status Code

See [Status Code](#).

## 4.1.7 Querying the Defense Status of a Specified EIP

### Functions

This API allows you to query the defense status of a specified EIP.

### URI

- URI format  
GET /v1/{project\_id}/antiddos/{floating\_ip\_id}/status
- Parameter description

Parameter	Mandator y	Type	Description
project_id	Yes	String	User ID

Parameter	Mandatory	Type	Description
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

## Request

### Request parameters

None

## Response

- Parameter description

Parameter	Mandatory	Type	Description
status	Yes	String	<p>Defense status, the possible value of which is one of the following:</p> <ul style="list-style-type: none"> <li><b>normal</b>: indicates that the defense status is normal.</li> <li><b>configuring</b>: indicates that defense is being configured.</li> <li><b>notConfig</b>: indicates that defense is not configured.</li> <li><b>packetcleaning</b>: indicates that traffic cleaning is underway.</li> <li><b>packetdropping</b>: indicates that traffic is discarded.</li> </ul>

## Example

- Example request

```
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8/status
```
- Example response

```
{"status": "normal"}
```

## Status Code

See [Status Code](#).

## 4.1.8 Querying the Traffic of a Specified EIP

### Functions

This API allows you to query the traffic of a specified EIP in the last 24 hours. Traffic is detected in five-minute intervals.

### URI

- URI format  
GET /v1/{project\_id}/antiddos/{floating\_ip\_id}/daily
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

### Request

#### Request parameters

None

### Response

- Parameter description

Name	Type	Description
data	Data structure	Traffic in the last 24 hours

- Data structure description of **data**

Parameter	Mandatory	Type	Description
period_start	Yes	Long integer	Start time
bps_in	Yes	Integer	Inbound traffic (bit/s)
bps_attack	Yes	Integer	Attack traffic (bit/s)
total_bps	Yes	Integer	Total traffic
pps_in	Yes	Integer	Inbound packet rate (number of packets per second)



Parameter	Mandatory	Type	Description
pps_attack	Yes	Integer	Attack packet rate (number of packets per second)
total_pps	Yes	Integer	Total packet rate

## Example

- Example request  
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8/daily

- Example response

```
{
  "data": [
    {
      "period_start": 1472713370609,
      "bps_in": 0,
      "bps_attack": 0,
      "total_bps": 0,
      "pps_in": 0,
      "pps_attack": 0,
      "total_pps": 0
    },
    ...
    {
      "period_start": 1472713670609,
      "bps_in": 0,
      "bps_attack": 0,
      "total_bps": 0,
      "pps_in": 0,
      "pps_attack": 0,
      "total_pps": 0
    }
  ]
}
```

## Status Code

See [Status Code](#).

## 4.1.9 Querying Events of a Specified EIP

### Functions

This API allows you to query events of a specified EIP in the last 24 hours. Events include cleaning and blackhole events, and the query delay is within five minutes.

### URI

- URI format  
GET /v1/{project\_id}/antiddos/{floating\_ip\_id}/logs

 NOTE

You can use **?** and **&** behind the URI to add query conditions, as shown in the request example.

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID
floating_ip_id	Yes	String	ID corresponding to the EIP of a user

## Request

**Table 4-8** Parameter description

Parameter	Mandatory	Type	Description
limit	No	Integer	Limit of number of returned results or the maximum number of returned results of a query. The value ranges from 1 to 100, and this parameter is used together with the <b>offset</b> parameter. If neither <b>limit</b> nor <b>offset</b> is used, query results of all ECSs are returned.
offset	No	Integer	Offset. This parameter is valid only when used together with the <b>limit</b> parameter.
sort_dir	No	String	Possible values: <ul style="list-style-type: none"> <li>• <b>desc</b>: indicates that query results are given and sorted by time in descending order.</li> <li>• <b>asc</b>: indicates that query results are given and sorted by time in ascending order.</li> </ul> The default value is <b>desc</b> .

## Response

- Parameter description

Name	Type	Description
total	Integer	Total number of EIPs
logs	Data structure	List of events

- Data structure description of **logs**

Parameter	Mandatory	Type	Description
start_time	Yes	Long integer	Start time
end_time	Yes	Long integer	End time
status	Yes	Integer	Defense status, the possible value of which is one of the following: <ul style="list-style-type: none"> <li><b>1</b>: indicates that traffic cleaning is underway.</li> <li><b>2</b>: indicates that traffic is discarded.</li> </ul>
trigger_bps	Yes	Integer	Traffic at the triggering point
trigger_pps	Yes	Integer	Packet rate at the triggering point
trigger_http_pps	Yes	Integer	HTTP request rate at the triggering point

## Example

- Example request  
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/1df977c2-fdc6-4483-bc1c-ba46829f57b8/logs

- Example response

```
{
  "total": 1,
  "logs": [
    {
      "start_time": 1473217200000,
      "end_time": 1473242400000,
      "status": 1,
      "trigger_bps": 51106,
      "trigger_pps": 2600,
      "trigger_http_pps": 3589
    }
  ]
}
```

## Status Code

See [Status Code](#).

## 4.1.10 Querying Weekly Defense Statistics

### Functions

This API allows you to query weekly defense statistics about all your IP addresses, including the number of intercepted DDoS attacks, number of attacks, and ranking by the number of attacks. Currently, you can query weekly statistics up to four weeks before the current time. Data older than four weeks cannot be queried.

### URI

- URI format  
GET /v1/{project\_id}/antiddos/weekly

#### NOTE

You can use **?** and **&** behind the URI to add query conditions, as shown in the request example.

- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID

### Request

**Table 4-9** Parameter description

Parameter	Mandatory	Type	Description
period_start_date	No	String	Start date of a seven-day period

### Response

- Parameter description

Name	Type	Description
ddos_intercept_times	Integer	Number of DDoS attacks blocked in a week
weekdata	Data structure	Number of attacks in a week
top10	Data structure	Top 10 attacked IP addresses

- Data structure description of **weekdata**

Parameter	Mandatory	Type	Description
ddos_intercept_times	Yes	Integer	Number of DDoS attacks blocked
ddos_blackhole_times	Yes	Integer	Number of DDoS black holes
max_attack_bps	Yes	Integer	Maximum attack traffic
max_attack_conns	Yes	Integer	Maximum number of attack connections
period_start_date	Yes	Long integer	Start time

- Data structure description of **top10**

Parameter	Mandatory	Type	Description
floating_ip_addresses	Yes	String	EIP
times	Yes	Integer	Number of DDoS attacks intercepted, including cleaning operations and black holes

## Example

- Example request  
GET /v1/67641fe6886f43fcb78edbbf0ad0b99f/antiddos/weekly?  
period\_start\_date=1006510306

- Example response
 

```
{
  "ddos_intercept_times": 23,
  "weekdata": [
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474214461651
    },
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474300861651
    },
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474387261651
    }
  ]
}
```

```
    },
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474473661651
    },
    {
      "ddos_intercept_times": 0,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474560061651
    },
    {
      "ddos_intercept_times": 2,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 16375,
      "max_attack_conns": 0,
      "period_start_date": 1474646461651
    },
    {
      "ddos_intercept_times": 1,
      "ddos_blackhole_times": 0,
      "max_attack_bps": 0,
      "max_attack_conns": 0,
      "period_start_date": 1474732861651
    }
  ],
  "top10": [
    {
      "floating_ip_address": "192.168.44.69",
      "times": 23
    }
  ]
}
```

## Status Code

See [Status Code](#).

## 4.2 Alarm Reminding APIs

### 4.2.1 Querying Alarm Configuration

#### Functions

This API allows you to query alarm configuration, such as whether a certain type of alarms will be received, and whether alarms are received through SMS messages or emails.

#### URI

- URI format  
GET /v2/{project\_id}/warnalert/alertconfig/query
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID

## Request

### Request parameters

None

## Response

- Parameter description

Parameter	Type	Description
warn_config	List data structure	Alarm configuration
topic_urn	String	ID of an alarm group
display_name	String	Specifies the name of the SMN topic used for sending alarm notifications.

- Data structure description of **warn\_config**

Parameter	Type	Description
antiDDoS	Boolean	DDoS attacks
bruce_force	Boolean	Brute force cracking (system logins, FTP, and DB)
remote_login	Boolean	Alarms about remote logins
weak_password	Boolean	Weak passwords (system and database)
high_privilege	Boolean	Overly high rights of a database process
back_doors	Boolean	Webshells
waf	Boolean	Reserved field
send_frequency	Integer	Possible values: <ul style="list-style-type: none"><li><b>0</b>: indicates that alarms are sent once a day.</li><li><b>1</b>: indicates that alarms are sent once every half hour.</li></ul> This parameter is mandatory for the Host Intrusion Detection (HID) service.

## Example

- Example request  
GET /v2/67641fe6886f43fcb78edbbf0ad0b99f/warnalert/alertconfig/query

- Example response

```
{
  "warn_config": {
    "antiDDoS": true,
    "bruce_force": false,
    "remote_login": false,
    "weak_password": false,
    "high_privilege": false,
    "back_doors": false,
    "waf": false
  },
  "topic_urn": "urn:smn:region01:67641fe6886f43fcb78edbbf0ad0b99f:test_soft",
  "display_name": "group_1"
}
```

### NOTE

SFTP is more secure than FTP. To secure data transmission, use SFTP to transfer files.

## Status Code

See [Status Code](#).

## 4.2.2 Updating Alarm Configuration

### Functions

This API allows you to update alarm configuration, such as whether a certain type of alarms will be received, and whether alarms are received through SMS messages or emails.

### URI

- URI format  
POST /v2/{project\_id}/warnalert/alertconfig/update
- Parameter description

Parameter	Mandatory	Type	Description
project_id	Yes	String	User ID

### Request

- Parameter description

Parameter	Mandatory	Type	Description
warn_config	Yes	List data structure	Alarm configuration
topic_urn	Yes	String	ID of an alarm group



Parameter	Mandatory	Type	Description
display_name	Yes	String	Specifies the name of the SMN topic used for sending alarm notifications.

- Data structure description of **warn\_config**

Parameter	Mandatory	Type	Description
antiDDoS	No	Boolean	DDoS attacks
bruce_force	No	Boolean	Brute force cracking (system logins, FTP, and DB)
remote_login	No	Boolean	Alarms about remote logins
weak_password	No	Boolean	Weak passwords (system and database)
high_privilege	No	Boolean	Overly high rights of a database process
back_doors	No	Boolean	Webshells
waf	No	Boolean	Reserved
send_frequency	No	Integer	Possible values: <ul style="list-style-type: none"> <li>• <b>0</b>: indicates that alarms are sent once a day.</li> <li>• <b>1</b>: indicates that alarms are sent once every half hour.</li> </ul> This parameter is mandatory for the Host Intrusion Detection (HID) service.

 **NOTE**

SFTP is more secure than FTP. To secure data transmission, use SFTP to transfer files.

## Response Message

Parameter	Type	Description
error_code	String	Internal error code

Parameter	Type	Description
error_msg	String	Internal error description
task_id	String	Task ID

## Example

- Example request

```
{
  "warn_config": {
    "antiDDoS": true,
    "bruce_force": false,
    "remote_login": false,
    "weak_password": false,
    "high_privilege": false,
    "back_doors": false,
    "waf": false
  },
  "topic_urn": "urn:smn:region01:67641fe6886f43fcb78edbbf0ad0b99f:test_soft",
  "display_name": "group_1"
}
```

- Example response

```
{
  "error_code": "10000000",
  "error_msg": "Ok",
  "task_id": ""
}
```

## Status Code

For details, see [Status Code](#).

# A Appendix

## A.1 Status Code

- Normal

Returned Value	Description
200	The request is successfully processed.

- Abnormal

Status Code	Status	Description
400	Bad Request	The server fails to process the request.
401	Unauthorized	The requested page requires a username and a password.
403	Forbidden	Access to the requested page is denied.
404	Not Found	The server fails to find the requested page.
405	Method Not Allowed	Method specified in the request is not allowed.
406	Not Acceptable	Response generated by the server is not acceptable to the client.
407	Proxy Authentication Required	Proxy authentication is required before the request is processed.
408	Request Timeout	A timeout error occurs because the request is not processed within the specified waiting period of the server.

Status Code	Status	Description
409	Conflict	The request cannot be processed due to a conflict.
500	Internal Server Error	The request is not processed due to a server error.
501	Not Implemented	The request is not processed because the server does not support the requested function.
502	Bad Gateway	The request is not processed, and the server receives an invalid response from the upstream server.
503	Service Unavailable	The request is not processed due to a temporary system abnormality.
504	Gateway Timeout	A gateway timeout error occurs.

## A.2 Error Codes

Status Code	Error Codes	Error Message	Description	Solution
200	Anti-DDoS.0	Succeeded	Succeeded	No need to deal with it
200	Anti-DDoS.10000000	The task has been received and is being handled	The task has been received and is being handled	No need to deal with it
400	Anti-DDoS.10000001	Enter a valid request message	Enter a valid request message	Check the parameters
400	Anti-DDoS.10001008	An incorrect task ID is used	An incorrect task ID is used	Check the parameters
400	Anti-DDoS.10001010	Invalid time	Invalid time	Check the parameters
401	Anti-DDoS.10000004	Public test service denied	Public test service denied	Apply for public test

Status Code	Error Codes	Error Message	Description	Solution
403	Anti-DDoS.10000002	Failed to authenticate the token in the request	Failed to authenticate the token in the request	Re apply for token
403	Anti-DDoS.10000009	The account is restricted	The account is restricted	Apply for authority
403	Anti-DDoS.10000010	The account is frozen	The account is frozen	Apply for unfreeze
403	Anti-DDoS.10000012	Unknown user type	Unknown user type	Apply for authority
403	Anti-DDoS.10000016	VPC access failed or EIP is not exist	VPC access failed or EIP is not exist	Contact to administrator
403	Anti-DDoS.10000030	You have not been authenticated. Perform real-name authentication first.	You have not been authenticated. Perform real-name authentication first.	Real name authentication
403	Anti-DDoS.10001009	The operation permission is restricted	The operation permission is restricted	Apply for authority
403	Anti-DDoS.11000001	Access to the database is rejected	Access to the database is rejected	Contact to administrator
500	Anti-DDoS.11000000	Internal system exception. Contact technical support engineers	Internal system exception. Contact technical support engineers	Contact to administrator

## A.3 Obtaining a Project ID

### Obtaining a 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 the administrator. For details about API authentication, see [Authentication](#).

In the following example, **id** indicates the project ID.

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "xxxxxxx",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

### Obtaining a Project ID from the Console

A project ID is required for some URLs when an API is called. To obtain a project ID, perform the following operations:

1. Log in to the management console.
2. Click the username and choose **My Credential** from the drop-down list.  
On the **My Credential** page, view project IDs in the project list.

# B Change History

---

Release Date	Description
2024-04-15	This is the first official release.