API Gateway

Service Overview

Issue	01
Date	2023-04-07

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.

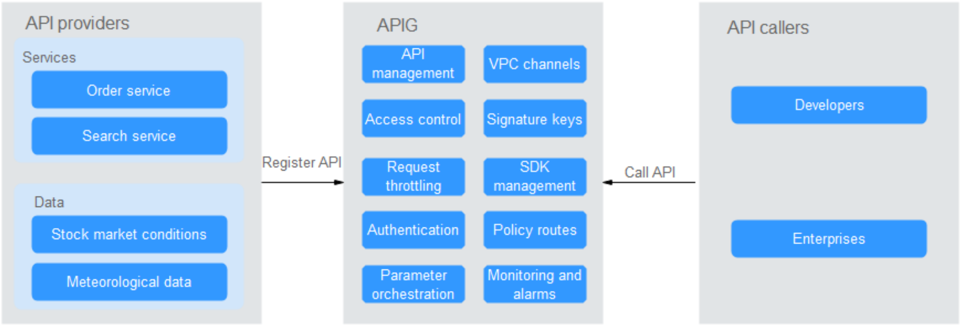
Contents

1 What Is APIG?

API Gateway (APIG) is your fully managed API hosting service. With APIG, you can build, manage, and deploy APIs at any scale to package your capabilities. With just a few clicks, you can integrate internal systems, monetize service capabilities, and selectively expose capabilities with minimal costs and risks.

- To monetize your capabilities (services and data), you can open them up by creating APIs in APIG. Then you can provide the APIs for API callers using offline channels.
- You can also obtain open APIs from APIG to reduce your development time and costs.

Figure 1-1 APIG architecture



Product Functions

- **API lifecycle management**

  The lifecycle of an API involves creating, publishing, removing, and deleting the API. API lifecycle management enables you to quickly and efficiently expose service capabilities.
- **Cloud native gateway**

  APIG integrates traffic ingress (Kubernetes Ingress) and microservice governance (Kubernetes Gateway API) in one gateway, improving performance, simplifying the architecture, and reducing deployment and O&M costs.
- **Built-in debugging tool**

With the built-in debugging tool, you can debug APIs using different HTTP headers and request bodies. This tool simplifies the API development process and reduces the API development and maintenance costs.

- **Version management**

  An API can be published in different environments. Publishing an API again in the same environment will override the API's previous version. APIG displays the publication history (including the version, description, date and time, and environment) of each API. You can roll back an API to any historical version to meet dark launch and version upgrade requirements.

- **Environment variables**

  Environment variables are manageable and specific to environments. Variables of an API will be replaced by the values of the variables in the environment where the API will be published. You can create variables in different environments to call different backend services using the same API.

- **Refined request throttling**
  - For different service demands and user levels, you can control the frequency at which an API can be called by a user, credential, or IP address, ensuring that backend services can run stably.
  - The throttling can be accurate to the second, minute, hour, or day.
  - Set throttling limits for excluded applications and tenants.

- **Monitoring and alarms**

  APIG provides visualized, real-time API monitoring, and displays multiple metrics, including number of requests, invocation latency, and number of errors. The metrics help you understand the API usage, allowing you to identify potential service risks.

- **Security**
  - Domain name access can be authenticated with TLS 1.1 and TLS 1.2.
  - Access control policies limit API access from specific IP addresses or accounts. You can blacklist or whitelist certain IP addresses and accounts to access your APIs.
  - Identity authentication can be based on AK/SK, tokens, and function-based custom authorizers. APIG verifies your backend services via certificates and is verified by your backend services through signature keys.

- **Monitoring and alarms**

  APIG provides visualized, real-time API monitoring, and displays multiple metrics, including number of requests, invocation latency, and number of errors. The metrics help you understand the API usage, allowing you to identify potential service risks.

- **VPC channels**

  VPC channels can be created for accessing resources in Virtual Private Clouds (VPCs) and exposing capabilities of backend services deployed in VPCs. A VPC channel forwards API requests to different servers for load balancing.

- **Mock response**

  Mock backends simulate API responses for circuit breakers, service degradation, and redirection.

2 Product Advantages

Available Out-of-the-Box

You can quickly create APIs by configuring the required settings on the APIG console. APIG provides an inline debugging tool to simplify API development, and allows you to publish an API in multiple environments for easy testing and fast iteration.

Convenient API Lifecycle Management

APIG provides full-lifecycle API management, including design, development, test, publish, and O&M, to help you quickly build, manage, and deploy APIs at any scale.

Refined Request Throttling

APIG combines synchronous and asynchronous traffic control and multiple algorithms to throttle requests at the second level. You can flexibly define request throttling policies to ensure stability and continuity of API services.

Visualized API Monitoring

APIG monitors the number of API calls, data latency, and number of errors, helping you identify potential service risks.

Comprehensive Security Protection

APIG provides multiple measures to secure API calling, such as Secure Sockets Layer (SSL) transfer, strict access control, IP address blacklist/whitelist, authentication, anti-replay, anti-attack, and multiple audit rules. In addition, APIG implements flexible and refined quota management and request throttling to help you flexibly and securely open your backend services.

Flexible Policy Routes

You can configure backends for an API to forward requests according to multiple policies. This facilitates dark launch and environment management.

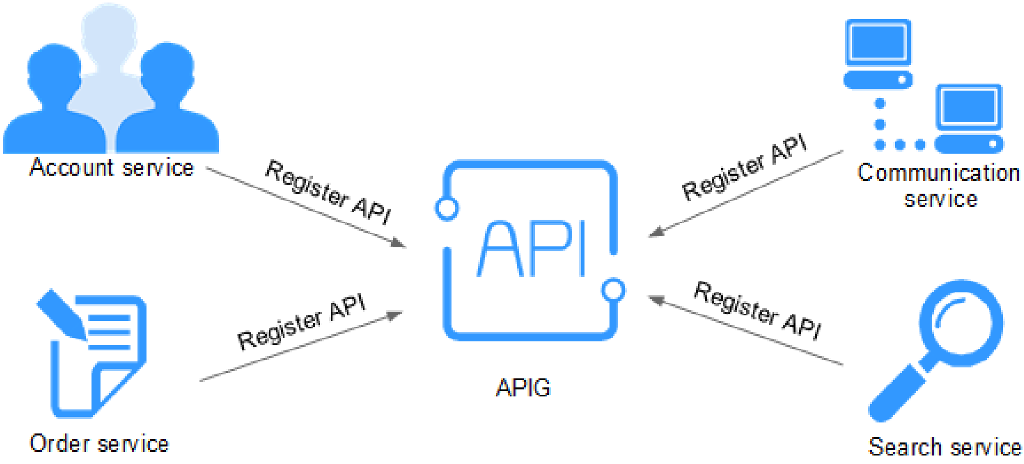
SDKs of Different Programming Languages

SDKs of different programming languages (such as Java, Go, Python, and C) are available for access from clients. Because the backends do not need to be modified, only one system is required to adapt to different service scenarios (such as mobile devices and IoT).

3 Application Scenarios

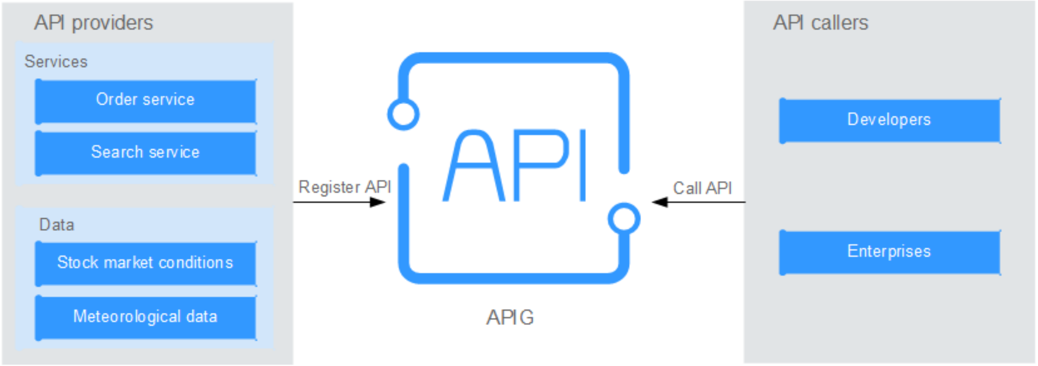
Internal System Decoupling

As enterprises develop rapidly with quick business changes, internal systems of enterprises need to keep pace with the development. However, it is difficult to ensure system universality and stability because internal systems are dependent on each other. APIG uses standard RESTful APIs to simplify the service architecture, decouples internal systems, and separates the frontend from backend. Existing capabilities can be reused to avoid repetitive development.



Enterprise Capabilities Opening

An enterprise cannot develop without partners' capabilities, such as a third-party payment platform and partner account login. APIG enables you to selectively expose capabilities to partners by using standard APIs and share services and data with partners to build a new ecosystem.


API providers
Services
Order service
Search service
Data
Stock market conditions
Meteorological data
Register API
API
APIG
Call API
API callers
Developers
Enterprises

4 Specifications

Dedicated Gateway Specifications

Table 4-1 lists the specifications of dedicated API gateways.

Table 4-1 Specifications of dedicated gateways

Edition	Maximum Number of Requests per Second
Basic	2000
Professional	4000
Enterprise	6000
Platinum	10,000

📖 NOTE

- For dedicated gateways, you can adjust the maximum number of requests per second for each API.
- The specifications of dedicated gateways cannot be modified.
- The dedicated gateway specifications are obtained by testing in the following conditions:
  - Protocol: HTTPS
  - Connection type: long connection
  - Concurrent requests: 100
  - Authentication mode: none
  - Size of returned data: 1 KB
  - Bandwidth: 10 MB/s

5 Notes and Constraints

To change the default restrictions, contact technical support to increase the quota. For details about parameter configuration of a dedicated gateway, see **Modifying Configuration Parameters**.

Table 5-1 Dedicated API gateway quotas

Item	Default Restriction	Modifiable
Gateways	5	√
API groups	1500	√
APIs	Number of APIs for each gateway edition: ● Basic: 250 ● Professional: 800 ● Enterprise: 2000 ● Platinum: 8000	√
Backend policies	5	√
Apps	50. The app quota includes the apps you have created.	√
Request throttling policies	● You can create a maximum of 300 request throttling policies for each gateway. ● The call limit for a single user cannot exceed that for the target API. ● The call limit for a single app cannot exceed that for a single user. ● The call limit for a single IP address cannot exceed that for the target API.	√
Environments	10	√

Item	Default Restriction	Modifiable
Signature keys	200	√
Access control policies	100	√
VPC channels	200	√
Variables	You can create a maximum of 50 variables for an API group in each environment.	√
Independent domain names	A maximum of five independent domain names can be bound to an API group.	√
Cloud servers	A maximum of 10 cloud servers can be added to a VPC channel.	√
Parameters	A maximum of 50 parameters can be created for an API.	√
API publication records	A maximum of 10 publication records of an API can be retained for each environment.	√
API access rate	Up to 6000 times per second	√
Excluded apps	A maximum of 30 excluded apps can be added to a request throttling policy.	√
Excluded tenants	A maximum of 30 excluded tenants can be added to a request throttling policy.	√
Access to a subdomain name	A subdomain name can be accessed up to 1000 times a day.	x
Maximum size of an API request package	12 MB	√
TLS protocol	TLS 1.1 and TLS 1.2 are supported. TLS 1.2 is recommended.	√
Custom authorizers	50	x
Plug-ins	500	√

6 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your APIG resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your resources.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to the employees to control their access to specific resources.

If your account does not require individual IAM users for permissions management, skip this chapter.

APIG Permissions

By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups, and attach policies or roles to these groups. The user then inherits permissions from the groups to which the user belongs, and can perform specified operations on cloud services based on the permissions.

APIG is a project-level service deployed and accessed in specific physical regions. To assign APIG permissions to a user group, you need to specify region-specific projects for which the permissions will take effect. If you select **All projects**, the permissions will be granted for both the global service project and all region-specific projects. When accessing APIG, the users need to switch to a region where they have been authorized to use this service.

You can grant permissions by using roles and policies.

- Roles: A type of coarse-grained authorization mechanism that defines permissions related to user responsibilities. This mechanism provides only a limited number of service-level roles for authorization. When using roles to grant permissions, you need to also assign other dependent roles for permissions to take effect. However, roles are not an ideal choice for fine-grained authorization and secure access control.
- Policies: A type of fine-grained authorization mechanism that defines permissions required to perform operations on specific cloud resources under certain conditions. This mechanism allows for more flexible policy-based authorization and meets requirements for secure access control. For example, you can grant APIG users only the permissions for performing specific

operations. Most policies define permissions based on APIs. For the API actions supported by APIG, see **Permissions Policies and Supported Actions**.

Table 6-1 lists all the system-defined roles and policies supported by APIG.

Table 6-1 System-defined roles and policies supported by APIG

Role/ Policy Name	Description	Type	Dependency
APIG Administrator	Administrator permissions for APIG. Users granted these permissions can use all functions of API gateways.	System-defined role	None
APIG FullAccess	Full permissions for APIG. Users granted these permissions can use all functions of **dedicated** gateways.	System-defined policy	None
APIG ReadOnly Access	Read-only permissions for APIG. Users granted these permissions can only view **dedicated** gateways.	System-defined policy	None

You can view the content of the preceding roles and policies on the IAM console. For example, the content of the **APIG FullAccess** policy is as follows:

```
{
    "Version": "1.1",
    "Statement": [
        {
            "Action": [
                "apig:*:*",
                "vpc:*:get*",
                "vpc:*:list*",
                "vpc:ports:create",
                "vpc:ports:update",
                "vpc:ports:delete",
                "vpc:publicIps:update",
                "FunctionGraph:function:listVersion",
                "FunctionGraph:function:list",
                "FunctionGraph:function:getConfig",
                "ecs:servers:list",
                "lts:groups:list",
                "lts:logs:list",
                "lts:topics:list"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Related Documents

- Section "Service Overview" in the *Identity and Access Management User Guide*
- Section "Creating a User and Granting Permissions" in the *API Gateway User Guide*

7 Basic Concepts

API

A set of predefined functions that encapsulates application capabilities. You can create APIs and make them accessible to users.

When creating an API, you need to configure the basic information and the frontend and backend request paths, parameters, and protocols.

API Group

A collection of APIs used for the same service. API groups facilitate API management.

Environment

A stage in the lifecycle of an API. An environment, such as API testing or development environment, specifies the usage scope of APIs, facilitating API lifecycle management. The same API can be published in different environments.

To call an API in different environments, you need to add the **x-stage** header parameter to the request sent to call the API. The value of this parameter is an environment name.

Environment Variable

A variable that is manageable and specific to an environment. You can create variables in different environments to call different backend services using the same API.

Request Throttling

Controls the number of times APIs can be called by a user, an app, or an IP address during a specific period to protect backend services.

Request throttling can be accurate to the minute and second.

Access Control

Access control policies are one of the security measures provided by APIG. They allow or deny API access from specific IP addresses or accounts.

App

An entity that requests for APIs. An app can be authorized to access multiple APIs, and multiple apps can be authorized to access the same API.

Signature Key

Consists of a key and secret, which are used by backend services to verify the identity of API Gateway and ensure secure access.

When an API bound with a signature key is called, API Gateway adds signature information to the API requests. The backend service of the API signs the requests in the same way, and verifies the identity of API Gateway by checking whether the signature is consistent with that in the **Authorization** header sent by API Gateway.

VPC Channel

A method for accessing VPC resources from API Gateway, allowing you to selectively expose backend services deployed in VPCs to third-party users.

Custom Authentication

A mechanism defined with custom rules for API Gateway to verify the validity and integrity of requests initiated by API callers. The mechanism is also used for backend services to verify the requests forwarded by API Gateway.

The following two types of custom authentication are provided:

- Frontend custom authentication: A custom authorizer is configured with a function to authenticate requests for an API.
- Backend custom authentication: A custom authorizer can be configured to authenticate requests for different backend services, eliminating the need to customize APIs for different authentication systems and simplifying API development. You only need to create a function-based custom authorizer in API Gateway to connect to the backend authentication system.

Simple Authentication

Simple authentication facilitates quick response for API requests by adding the **X-Apig-AppCode** parameter (whose value is an AppCode) to the HTTP request header. API Gateway verifies only the AppCode and does not verify the request signature.

Gateway Response

Gateway responses are returned if API Gateway fails to process API requests. API Gateway provides default responses for multiple scenarios and allows you to customize response status codes and content. You can add a gateway response in JSON format on the **API Groups** page.