Quick Start Guide

1. Introduction

1.1. Overview

Paraview API Management Platform, aka Platform, provides functionalities such as API management, traffic control, security, and analytics. The functionality of the Platform is extensible through plugins, making it a flexible and powerful solution for managing APIs and microservices. It features high reliability, high performance, powerful security protection and scalability.

1.2. Terminology

#	Terminology	Explanation
1	Cluster	Cluster refers to a group of interconnected API gateway instances that work together to handle incoming API requests. The cluster provides high availability, scalability, and fault tolerance for the API gateway infrastructure. By distributing the API traffic across multiple gateway instances, the cluster can handle large volumes of API requests and ensure that the system remains responsive even in the event of individual gateway failures. Additionally, the cluster configuration enables load balancing and can be used to apply consistent policies and configurations across all gateway instances in the cluster.
2	Plugin	An extension to the Gateway.
3	Gateway	Paraview API Gateway
4	HTTP	HTTP stands for Hypertext Transfer Protocol. It is an application protocol used for transmitting hypermedia documents, such as HTML files, over the internet. HTTP is the foundation of data communication for the World Wide Web.
5	LUA	Lua is a lightweight, high-level programming language designed primarily for embedding in applications. It is known for its simplicity, flexibility, and extensibility. Lua is often used as a scripting language in video games, embedded systems, and other applications that require a customizable and efficient scripting language.
6	Platform	API Management Platform

2. Operating Instructions

This documentation provides users with practical guidance to enhance their understanding of the Platform's features and operational procedures.

2.1. User Sign in

l **Description:** Users log in to Platform and the corresponding menus and data are displayed according to their account privileges.

l **Instruction:** Visit the Platform's address, open the login page, enter your username, password and dynamic authentication code, and click "Sign in".

2.2. Create User

l **Description:** Used by administrators to create users and give them role-specific permissions and allowlist IP access.

l **Instruction:**

1. Platform administrator logs into Platform and selects the [Settings] module.
2. Click [Users] to display all current users.
3. Click [Users] to enter the Add User page, which is divided into Basic Information and Detailed Information, Required Options: Username, Name, Password, Department, Role, Email, User Status (Enabled, Disabled, Locked).

The password rules are as follows: Supports 8-20 characters, including a combination of numbers, uppercase letters, lowercase letters, and special characters ($@!.%*#_~?&).

2.3. Initial Config

l **Description:** Platform operation and maintenance personnel maintain the initial configuration and settings.

l **Instruction:**

1. Click [Initial Configs] in [Settings].
2. In the General Configuration tab, administrators can configure

   (1) Frame Stamp Setting, Platform watermark function is not enabled by default. Supports watermark configuration to display the current username, current time, and manually configured static text.

   (2) Concurrent Login Configuration to allow the same account to log in from multiple locations by default.

   (3) Password Update Requirement, which is not enabled by default to force users to update their passwords.

   (4) International Configuration to allow users to switch languages by default and follow the default language for the first time.

   (5) Two-factor Authentication enable the MFA feature, which currently supports multi-factor authentication with email/SMS and password, and the MFA feature is not enabled by default

   (6) Sign In Configuration, the number of failed login attempts, the default is 5, and the account lockout time, the default is 60 minutes.
3. In the Logo Setting tab, administrators can upload and modify the Platform Logo and Platform Favicon.
4. In the Version Information tab, when deploying Platform for the first time or when the license expires, the administrator can upload the license to Update Authorization.

2.4. Clusters

2.4.1. Create Cluster

l **Description:** Gateway administrators can create API Gateway clusters.

l **Instruction:** Open the [Clusters] menu under [API Gateway] and click the [Add] button. Configure Cluster Name, ID, Cluster URL, API URL Prefix and other information to create the cluster.

Cluster Deployment Installation, please refer to the "Paraview API Gateway and Management Platform Installation Guide" for more information.

2.4.2. Node Management

l **Description:** Gateway administrators can view the status of nodes in each cluster to determine whether the cluster nodes are healthy or not.

l **Instruction:** Open the [Clusters] menu under [API Gateway] and click the [Node Status] button for the cluster. View the status of each node.

Node installation please refer to the "Paraview API Gateway and Management Platform Installation Guide" for more information.

The node is in an unavailable state, you can delete the node data. However, the node is in an available state, deleting the node data will still result in the node being automatically registered.

2.4.3. Redis Config

l **Description:** Redis is a key-value storage system. Rate limiting plugins, circuit breaking plugins require Redis configuration. Redis is used to cache counters.

l **Instruction:** Open the [Clusters] menu under [API Gateway], and click the [Redis Configuration] button of the cluster.

2.4.4. Certificates

l **Description:** HTTPS service binding HTTPS certificate, certificate public key must be carried when requesting service.

l **Instruction:** Click into the cluster to open the [Certificates] page and click the [Add] button and configure certificate information.

Add CA certificate and private key.

2.5. Quick Start

2.5.1. Create API

l **Description:** Quickly create APIs for the HTTP protocol.

l **Instruction:** Click into the cluster to open the [APIs] page and click the [Add] button.

Configure basic information, including API Name, Path, Protocols and HTTP Method, and click [Next] to save the API. Skip plugin addition.

Support creating APIs for HTTP, HTTPS, and TCP protocols.

2.5.2. Create Services

l **Description:** Quickly create Services for the HTTP, HTTPS, and TCP protocols.

LDAP uses TCP (Transmission Control Protocol) as the transport layer protocol for communication.

The WebSocket protocol is built on top of the HTTP protocol and establishes connections through handshake.

l **Instruction:** Configure the Service, including Name, Service URL, Connection Timeout (ms), Write Timeout (ms), Read Timeout (ms), Retries, and then click Next to save.

2.6. Publish the HTTP protocol API

2.6.1. Create Upstreams

l **Description:** Configuring load balancing allows requests to be distributed to multiple servers, avoiding overloading a single server and thus improving the overall performance and availability of Platform.

l **Instruction:**

1. Click into the cluster to open the [Upstreams] page and click the [Add] button.
2. Configure Basic Information, including Name, Proxy Service Name.
3. Configure Targets, including multiple groups of Host, Port and Weight, the default is to support load balancing according to weights.
4. To configure the Upstream Plugin, you can set the Preferred Strategy and set the Downgrade Strategy by means of Polling, Client IP, Request Header, Request Cookie, and Consumer.

2.6.2. Create Services

l **Description:** Services that support HTTP, HTTPS, and TCP protocols, and HTTPS services support configuring HTTPS certificates.

The WebSocket protocol is built on top of the HTTP protocol and establishes connections through handshake.

l **Instruction:** Configure the Service, including Name, Service URL, Connection Timeout (ms), Write Timeout (ms), Read Timeout (ms), Retries, and then click Next to save.

2.6.3. Create API

l **Description:** Quickly create APIs for the HTTP protocol.

l **Instruction:** Click into the cluster to open the [APIs] page and click the [Add] button. Configure Basic Information, including API Name, Path, Protocols and HTTP Method, and click Next to save the API.

Select an existing HTTP service, or create an HTTP service quickly.

2.6.4. Add Plugins

l **Description:** Gateway plugins are components used to extend and enhance the functionality of a gateway by providing a variety of additional functions and features at the gateway level. These include Traffic Control, Transformations, Synchronization, Authentication, Grayscale, Analytics & Monitoring, Security and Audit Logging.

l **Instruction:**

1. Click into the cluster to open the [APIs] page and click the [Plugins] button.
2. Go to the [Plugins] page and click the Add button.
3. Select a policy and enter the configuration information. Plugin configuration, please refer to Common Plugins for details.

2.6.5. Authorization

l **Description:** Create a consumer and set the authentication method, then the API authorizes the consumer and only the authorized consumer can access the API.

l **Instruction:**

1. Click into the cluster to open the [Consumers] page and click the [Add] button.
2. Configuring Consumer Name and Inside App Identity.
3. Click on the [Authentications] button, which has a variety of authentication methods, including Paraview OAuth2, Key Auth, Paraview HMAC Auth, OAuth2, Paraview JWT, Basic Auth, and Paraview IP Auth.
4. Here choose the key Auth authentication method, automatically generated apiKey.
5. Click into the cluster, open the [APIs] page, and click the [Plugins] button and add plugins.
6. Select the authentication-related plugins, here we choose the Key Auth plugin, bind the consumer for the consumer just created, API authorization is complete!

2.6.6. API Test

l **Description:** The role of API testing is to verify and ensure the correctness, reliability and security of the application program interface. By testing APIs, you can ensure that the APIs work properly in different environments and meet business requirements.

l **Instruction:**

1. Visit the APIs page in the cluster and click [Copy] to get the interface access address.
2. Click [edit] button to confirm the API Path, Request Header, Protocols and HTTP Method information.
3. After confirming the interface information, test the interface through API testing tools, such as postman, soap UI and so on.

2.7. Publish the HTTPS protocol API

l **Description:** Create a HTTPS-based API, publishing it on API Management Platform.

l **Instruction:**

1. Click into the cluster to open the [APIs] page and click the [Add] button.
2. Fill in the API Name field, select the HTTPS protocol, and configure Basic Information. Then, click [NEXT].
3. On the plugins page, select the desired plugins and click [NEXT].
4. In the Service page, add a HTTPS service for the Service field. Service binding service certificate, certificate must be carried when requesting service key.

2.8. Publish the TCP protocol API

l **Description:** Create a TCP-based API, publishing it on API Management Platform.

LDAP uses TCP (Transmission Control Protocol) as the transport layer protocol for communication

l **Instruction:**

1. Click into the cluster to open the [APIs] page and click the [Add] button.
2. Fill in the API Name field, select the TCP protocol, and fill in the Source IP and port with the consumer's IP and port (optional). Then, click [NEXT].
3. Fill in the Target IP field with the IP and port that the gateway is open to, with port being a required field. Then, click [NEXT].
4. On the plugins page, select the desired plugins and click [NEXT].
5. In the Service page, choose a TCP service for the Service field and click [NEXT].
6. In the Preview page, review the soon-to-be-published TCP protocol API. If everything is correct, click [Submit] to complete the publication of the TCP protocol API.

2.9. Common Plugins

All plugin introductions can be found in the Appendix, and the following text only provides detailed explanations for some plugins.

A single plugin instance always runs once per request. The configuration with which it runs depends on the entities it has been configured for. Plugins can be configured for various entities, combinations of entities, or even globally. This is useful, for example, when you want to configure a plugin a certain way for most requests, but make authenticated requests behave slightly differently.

Therefore, there is an order of precedence for running a plugin when it has been applied to different entities with different configurations. The amount of entities configured to a specific plugin directly correlate to its priority. The more entities configured to a plugin the higher its order of precedence is. The complete order of precedence for plugins configured to multiple entities is:

1. Plugins configured on a combination of a consumer, a API, and a service. (Consumer means the request must be authenticated).
2. Plugins configured on a combination of a consumer and a API. (Consumer means the request must be authenticated).
3. Plugins configured on a combination of a consumer and a service.
4. Plugins configured on a API and service.
5. Plugins configured on a consumer.
6. Plugins configured on a API.
7. Plugins configured on a service.
8. Plugins configured globally.

2.9.1. Request Rate Limiting

l **Description:** In enterprise applications, many services need to be invoked by external or internal applications at the same time, and the concurrency at a certain time is very large, which may result in slow response or interface unavailability. It is mainly used to verify that the API gateway product can ensure smooth message processing under the swarming phenomenon. It also guarantees the normal operation of the back-end services.

1. Supports flow limiting for services, APIs, and gateways.
2. Supports hour, minute, second, year, month, and day cycles.
3. Support for service prioritization policy, high concurrency high priority services are executed first, low priority don't reject or response is slower.

l **Instruction:**

1. Select the Request Rate Limiting plugin for traffic limiting configuration.
2. Support the configuration of which consumers need to use the flow-limiting plugin, support the configuration of the flow-limiting mode full flow-limiting or requesting party IP flow-limiting, support in accordance with the year, month, day, hour, minute, minute and second to accurately limit the request flow.
3. After the configuration is complete, test through the interface test tool, the flow limiting prompt appears.

Testing the API with Postman. The access frequency of the request exceeds the limit set in the policy, and access will be terminated.

2.9.2. Paraview WAF

l **Description:** In enterprise applications, penetration test protection for APIs, services can be traffic level, support the following ways:Scripting attacks, SQL injection, CSRF, XSS and other attacks.

l **Instruction:**

1. Select the Paraview WAF plugin for security configuration.
2. Support the configuration of whether to open the WAF, you can choose to intercept through the URL, QUERY parameters, POST request body, client type, cookies and so on, respectively, here to open the QUERY parameter interception, save the configuration.

The interface query parameter contains SQL injection information, the interface response results suggest that there is a security risk.

2.9.3. IP Allowlist/Denylist

l **Description:** In enterprise applications, if you find that an accessed application has offensive behavior or other abnormal behavior of the operation, you can block the operation by prohibiting its IP, allowlist and denylist support API, service, application. Allowlist and denylist can only be configured with one.

l **Instruction:**

1. Select the IP Allowlist/Denylist plugin to configure.
2. Supports configuration of multiple IP allowlist and denylist.

3. The IP denylist plugin is used to obtain the requester's IP and determine whether it exists on the allowlist or denylist during access, and then restrict or release it separately.

Both allowlist and denylist support IP and CIDR. CIDR is a method used to represent a range of IP addresses, such as: 10.10.1.0/24.

If the configured IP is allowlist, only requests originating from allowlist IPs are passed.After the configuration is complete, test with the interface test tool to verify that interface requests originating from "10.11.102.104" are passed.

2.9.4. Basic Converter

l **Description:** In enterprise applications, there are a lot of legacy Platforms need and the latest business services for data interoperability, legacy services due to the lack of development capabilities, the need for API gateway for format conversion, reducing the stability of the core system, the general situation needs to support the following two formats: JSON and XML

l **Instruction:**

1. Select the Basic Request/Response Converter plugin to configure.
2. Add, delete, or modify the HTTP header and response body of a request or response. Replacement, addition and append must be in 'key:value' format. When configured at the same time, the execution order is: remove->replace->add->append.
3. Here is an example of adding a response, configure add JSON:code:10086; configure add header:apiName:API.

API test through the interface test tool to add code node data in the response message and apiName header information in the response header.

2.9.5. Web scraping detection

l **Description:** Detects and prevents web scraping programs from automatically accessing and capturing content

l **Instruction:** Select the Web scraping detection plugin to configure.

The configuration is explained as follows:

Configuration items	Explain	Configuration Example
Allowlist	Check the User Agent header based on the configured regular expression.	(chrome)
Denylist	Check the User Agent header based on the configured regular expression.	(HTTrack|harvest|audit|dirbuster|pangolin|nmap|sqln|scan|hydra|Parser|libwww|BBBike|sqlmap|w3af|owasp|Nikto|fimap|havij|PycURL|zmeu|BabyKrokodil|netsparker|httperf|bench)

2.9.6. Request termination

l **Description:** Returns a fixed message and response code directly for the request.

l **Instruction:** Select Request Termination plugin to configure.

The configuration is explained as follows:

Configuration items	Explain	Configuration Example
HTTP Code	The code of the HTTP response header.	503
Response message	The response message message, if this field is filled in, the response type and response body must be empty.	
Response type	HTTP response type. If this field is filled in, the response body cannot be empty, and the response message must be empty.	application/json
Response body	The response message type is mutually exclusive to the response message field. If this field is filled in, the response message must be empty	{"message": "Terminate Request"}

API test with Postman. When the response code is 503, a fixed response message will be returned.

2.9.7. Gray Level Release

l **Description:** Returns a fixed message and response code directly for the request.

l **Instruction:** Select Gray Level Release plugin to configure.

The configuration is explained as follows:

Configuration items	Explain	Configuration Example
Request Path	List, configure the regular expression (LUA) for the request path. When the request path matches the current expression, the current request is marked as grayscale traffic.	/test/.*
Request IP	List, configure the source IP, support classless inter domain routing expressions. When the request source IP matches the list, the current request is marked as grayscale traffic.	128.14.32.0/20 128.14.32.1
HTTP Method	List, method for configuring requests. When the requested method matches the configuration list, the current request is marked as grayscale traffic	GET

Configuration items	Explain	Configuration Example
Request header	List, configure a filtering list for request headers. When the requested header information matches the list rules, the current request is marked as grayscale traffic	
key	The tag name of the request header	User-Agent
value	The label value or regularization of the request header	*.firefox.*
Is it regular	If the switch is turned on, the value is regular and matches with the value of the request header according to the regularity	Open
Request Body	List, configure the filtering list for the request body. When the body (JSON) field of the request matches the list rule, the current request is marked as grayscale traffic	10
key	Request body JSON field name	username
value	Request body JSON field value or regularization	test
Is it regular	If the switch is turned on, the value is regular and matches with the value of the request header according to the regularity	close
Universal configuration		
Add request header	Request header attached when forwarding to the interface	
key	Request header signature	is_gray
value	Request header label value	true
proxy_host	When hitting the grayscale rule, the host name (payload name) forwarded by the proxy can be configured with a fixed value or reference to the original host name, such as:${proxy_host}_gray	${proxy_host}_gray
proxy_port	When hitting the grayscale rule, the host port number forwarded by the proxy can be configured with a fixed port number or referenced from the original port number, such as ${proxy_port}	${proxy_port}

When requesting, if the User Agent header does not contain 'Firefox', it should be forwarded normally. Otherwise, it should be forwarded to Baidu (the commonly configured proxy_host).

2.9.8. Anti-Replay Attack

l **Description:** Prevents duplicate requests within a short period of time.

l **Instruction:** Select Anti-Replay Attack plugin to configure. Anti replay attack plugin depends on Redis, please configure it in the cluster first.

The configuration is explained as follows:

Configuration items	Explain	Configuration Example
Type	Determine whether Redis adopts cluster or standalone mode	Single machine Redis mode
Request timestamp format	The format of the registration request timestamp includes YYYYMMDDHHmmssSSS and long_ Time_ Millis, two types	
Request timestamp key	Register the timestamp key in the request header	X-Timestamp
Serial number key	The key for the serial number in the registration request header	X-Sequence-No
Prefix for key concatenation	Prefix for registering Redis storage	repeat
Clock deviation (seconds)	The maximum time interval allowed for registration requests is in seconds. If the interval between the timestamp of a request and the current time exceeds the allowed range, access is not allowed	1
Fault-tolerant switch	If Redis is abnormal, will the request be released	close

Access is not allowed if the interval between the timestamp of the request and the current time exceeds the allowed range.

2.9.9. Mock

l **Description:** Be able to simulate various expected API response results for front-end page debugging when the backend interface is not yet developed.

l **Instruction:** Select mock plugin to configure. Prevents duplicate requests within a short period of time.

The configuration is explained as follows:

Configuration items	Explain	Configuration Example
Consumer	The default restriction is "all", and specific consumers can be selected from the dropdown menu	
Service	The default restriction is "all", and specific services can be selected from the dropdown menu	
API	The default restriction is "all", and specific APIs can be selected from the dropdown menu.	
Status code	Response status code	200
Response header	Response header information	key:name value: zhangsan
Response body	Response message information	{"msg":"error"}
Delay response time	Delay response time in milliseconds	1000
Remarks		

2.10. Plugins Hot Deployment

l **Description:** In many scenarios, plugins need to be hot deployed without affecting business continuity, and the interface and code of the plugins can be extended in the management platform and synchronized to the gateway in real time.

l **Instruction:**

1. Open the [Plugins] page and click the [Versions] button.
2. Add or modify the current version of the package, then upload the package and set a new version number.
3. Open the [Clusters] page and click the [Plugin] button.
4. After entering the cluster plugin management page, click [Replace Version] to update the plugin version.

2.11. Logs

l To understand who performed what actions in the platform, the operator or administrator can review the Operation Log.

l To investigate API call exceptions, the operator or administrator can check the Error Log.

l To monitor API invocation status, the operator or administrator can review the Proxy Log.

l To understand the operations performed by plugins on API requests and responses, the operator or administrator can review the Runtime Log.

2.11.1. Operation Log

l **Description:** The purpose of operation logs is to record and track the actions performed in a system or application. Operation logs provide a detailed history of events and activities, capturing important information such as who performed the action, what action was performed, when it was performed, and any relevant details or parameters.

l **Instruction:**

1. Platform administrator logs into the API platform, selects the [Audit Logging] module, and clicks [Operation Log] to enter the logging page.
2. You can view the user name, type, operation module, operation object, operation status, operation time, and select a specific time range for filtering.

2.11.2. Error Log

l **Description:** The purpose of API exception logs is to capture and record information about exceptions or errors that occur during the execution of an API (Application Programming Interface) call. These logs are essential for troubleshooting, debugging, and maintaining the reliability and efficiency of API-based systems.

l **Instruction:**

1. Platform administrator logs into the API platform, selects the [Audit Logging] module, clicks [Error Log], and enters the log page.
2. You can view the Service-Name, API-Name, Consumer, Sequence-No, Time-Consuming, Status-Code, select a specific time range for filtering, and support failure to manually re-transmission.

2.11.3. Proxy Log

l **Description:** The purpose of API proxy logs is to capture and record information about API requests and responses that pass through an API proxy or gateway. These logs are essential for monitoring, troubleshooting, and securing API traffic within an organization’s infrastructure.

API called but no log entries ca be found in the proxy log. Please add the log plugin to the global configuration of the current cluster.

l **Instruction:**

1. Platform administrator logs into the API platform, selects the [Audit Logging] module, clicks [Proxy Log], and enters the log page.
2. You can view the Service-Name, API-Name, Consumer, Sequence-No, Time-Consuming, Status-Code, select a specific time range for filtering, and support failure to manually re-transmission.

2.11.4. Runtime Log

l **Description:** The purpose of API gateway plugin runtime logs is to capture and record information about the execution of API gateway plugins. These logs provide valuable insights into the behavior, performance, and security of API requests and responses processed by an API gateway with specific plugins enabled.

l **Instruction:**

1. Platform administrator logs into the API platform, selects the [Audit Logging] module, clicks [Runtime Log], and enters the log page.
2. You can view Plugin-ID, Log-Level, Log-Info, Cluster-ID, Client-IP, API-Name, Service-Name, and select a specific time range for filtering.

Appendix

#	Plugin Type	Name	Description
1	TRAFFIC_CONTROL	Access Time Control	Only allows resource access within specific time periods.
2	ANALYTICS&MONITORING	Link Tracking	Assigns a global serial number to request traffic and sends it to upstream services via request headers. The gateway traffic logs record the global serial number.
3	TRAFFIC_CONTROL	Request Termination	Returns a fixed message and response code directly for the request.
4	SYNCHRONIZATION	API Parsing	Stores API basic information in the context for use by other strategies.
5	LOGGING	Client Traffic Logging	Records client request logs.
6	TRANSFORMATIONS	Basic-Correlation-ID	Use a unique ID to associate requests and responses.
7	AUTHENTICATION	Unified Authentication	Authenticates whether an API meets current application authentication requirements.
8	SECURITY	Basic-ACL-Auth	Control which users can access.
9	TRANSFORMATIONS	HTTP to MQTT Protocol Conversion	Converts HTTP requests to MQTT requests.
10	AUTHENTICATION	Universal Application Authentication	
11	AUTHENTICATION	Paraview-Oauth2.0-Auth	Validates the Token carried in the request.
12	AUTHENTICATION	Basic Auth	Validates the username and password in the request.
13	TRANSFORMATIONS	HTTP to Dubbo Protocol Conversion	Converts HTTP requests to Dubbo requests.

#	Plugin Type	Name	Description
14	SECURITY	Application Authorization	Authenticates whether an API has application permissions.
15	SECURITY	Static Resource Compression	Compresses static resources of web pages to reduce bandwidth usage.
16	SECURITY	Circuit Breaker Plugin	Intelligently performs circuit breaker based on different conditions and can recover after a certain period of time.
17	SECURITY	Paraview WAF	Provides common threat protection against XML/SOAP format validation, SQL injection, XSS attacks, script attacks, XEE attacks, CSRF, etc.
18	SECURITY	Parameter Validation	Validates API interface request parameters.
19	AUTHENTICATION	Paraview-HMAC-Auth	Verifies the request header and body using HMAC digest algorithm.
20	AUTHENTICATION	Key Auth	Validates the API Key carried in the request HTTP header.
21	GRAY_LEVEL	Gray Level Release	Performs Gray Level release based on users, business models, and regions.
22	AUTHENTICATION	Basic-OAuth2.0-Auth	Validates the Token carried in the request.
23	AUTHENTICATION	IAM Login and Authorization Plugin	Integrate with IAM to implement fine-grained permission control for displaying menus after user login.
24	SECURITY	Cross-Origin Resource Sharing (CORS)	Used for data transmission and interaction between different domains. The cross-domain plugin bypasses the restrictions of the Same-Origin Policy, allowing cross-domain data transfer.

#	Plugin Type	Name	Description
25	SECURITY	Universal Signature	API authentication method that uses "Digital Certificate Authentication". Computes a unique data digest value for the data to be signed, encrypts the data digest using the private key, and generates a signature value.
26	SECURITY	Static Resource Caching	Caches commonly used static resources for faster response.
27	AUTHENTICATION	Paraview-JWT-Auth	Validates the JWT Token carried in the request.
28	AUTHENTICATION	Paraview-IP-Auth	Identifies the application's identity based on the client IP address.
29	TRAFFIC_CONTROL	Request Payload Size Limitation	Intercepts requests if the request payload size exceeds the configured limit.
30	SECURITY	Data Masking	Masks sensitive information in API interface request and response data
31	SECURITY	Anti-Replay Attack	Prevents duplicate requests within a short period of time.
32	SECURITY	Universal Decryption	API authentication method that uses "Digital Certificate Authentication". Decrypts API interface request and response data.
33	SECURITY	Universal Signature Verification	Based on the API with the authentication method of "digital authentication certificate", verify the signature data using the corresponding public key
34	SECURITY	Web scraping detection	Detects and prevents web scraping programs from automatically accessing and capturing content.

#	Plugin Type	Name	Description
35	SECURITY	IP Allowlist/Denylist	Intercepts IP addresses by setting up an IP denylist. Only allows specific IP addresses to access resources by setting up an IP allowlist.
36	SECURITY	Universal Encryption	API authentication method that uses "Digital Certificate Authentication". Encrypts API interface request and response data.
37	ANALYTICS&MONITORING	Service Health Check	Monitors the health status of load instances and triggers alarms when exceptions occur.
38	LOGGING	Log Collection - Kafka	Sends log information to a Kafka server.
39	SYNCHRONIZATION	Global Configuration	General parameter configuration for the gateway.
40	TRANSFORMATIONS	Basic Request Converter	Converts the format of the request message and headers.
41	TRANSFORMATIONS	Mock	Returns a fixed mock test message.
42	TRANSFORMATIONS	Basic Response Converter	Converts the format of the response message and headers.
43	TRANSFORMATIONS	HTTP to gRPC Protocol Conversion	Converts HTTP requests to gRPC requests.
44	TRANSFORMATIONS	Request Proxy Forwarding	Sends backend service interface responses to the context for use by other strategies.
45	TRANSFORMATIONS	RFC Proxy Adaptation	
46	TRANSFORMATIONS	Logic Orchestration	
47	TRAFFIC_CONTROL	Request Rate Limiting (Cluster)	Limit the frequency of cluster request access.
48	TRAFFIC_CONTROL	Request Rate Limiting (Node)	Limit the request access frequency of each node in the cluster.