

Tencent Cloud EdgeOne

Troubleshooting

Product Documentation



Copyright Notice

©2013–2026 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

Troubleshooting

- Reference for Abnormal Status Codes

- Troubleshooting Guide for EdgeOne 4XX/5XX Status Codes

- 520/524 Status Code Troubleshooting Guide

- 521/522 Status Code Troubleshooting Guide

Tool Guide

- Speed Test Tools

 - Real User Monitoring

- Diagnostic Tool

 - Self-service debugging

- IP Location Query

Troubleshooting

Reference for Abnormal Status Codes

Last updated: 2026-04-23 14:54:21

EdgeOne responds with the following types of exception status codes:

Standard Status Codes

You can refer to the [HTTP Status Code Standard](#) to see the specific meanings of these status codes.

These codes are typically responded to in the following situations:

- After the request is sent back to the origin, the status code information is responded by the origin server, and the node will pass transmit status code from the origin server to the client.
- Direct responses from EdgeOne nodes, for example, Token authentication fails, responding with a 403 status code. Possible status codes that may be directly responded by EdgeOne nodes include the following status ones:

Status Code	Description
400	The client request is invalid, such as when the request Method is not within the allowed range of EdgeOne. For details, see HTTP Restrictions .
403	<ul style="list-style-type: none">• Failed hotlink protection validation, such as Rules Engine's Token authentication.• Compliance blocking triggered.
416	range abnormality, such as rangeStart < 0, rangeStart > rangeEnd, rangeStart > FileSize.
418	<p>For domains connected to EdgeOne, the system automatically assigns service nodes to the domain. The corresponding nodes will distribute the domain's configuration file, with content determined by the domain's settings such as origin server, caching, headers, and so on. When a request is sent to a node, the node reads the domain's configuration file. If the configuration file is found to be missing, it responds with a 418 status code.</p> <p>For example, if a client requests: <code>https://example.com/test.jpg</code>, the node will read the configuration file for the domain <code>example.com</code>. Due to reasons such as the service node being bound to <code>non-example.com</code> domain, incorrect CNAME configuration, or abnormal scheduling system, the client may receive a 418 response.</p>
423	Looping request is detected when the Loops value in the CDN-Loop header \geq 16. For details, see CDN-Loop .

EdgeOne Custom Status Codes

EdgeOne defines special status codes with unique meanings. Status codes within the range of 520–599 are reserved by EdgeOne for custom non-standard responses. It is recommended to avoid using status codes within this range in your business to prevent confusion with EdgeOne's status codes.

Below are the custom EdgeOne status codes along with their meanings, facilitating self-troubleshooting when encountering abnormal business access, please refer to: [Troubleshooting Guide for EdgeOne 4XX/5XX Status Codes](#).

Status Code	Description
520/550	After a successful connection is established between the node and the origin server, the node sends a request to the origin server. If the origin server directly sends an RST packet, the node responds to the client with a 520/550 status code.
521/551	After the node requests the origin server, during the TCP connection establishment phase, if the origin server directly sends an RST packet, the node responds to the client with a 521/551 status code.
522/552	When the node requests the origin server during the TCP connection establishment phase, and the origin server fails to respond, causing the node to time out, the node responds to the client with a 522/552 status code.
523/553	If the origin server configured for the domain is a domain name itself, when the node attempts to back to the origin, it needs to resolve this domain name to obtain the origin server's IP address. If the resolution fails, the node cannot proceed to the origin and responds to the client with a 523/553 status code.
524/554	If the origin server fails to respond after a successful connection is established between the node and the origin server and the node sends a request, causing the node to time out, the node responds to the client with a 524/554 status code.
525/555	If the origin-fetch protocol is HTTPS, the node needs to perform an SSL handshake with the origin server when fetching content. If the handshake fails, the node responds to the client with 525/555 status code.
545	Edge Function execution exception, such as referencing an undefined variable.
566	When a request is intercepted by Web Protection – Managed Rules , it responds with the default 566 status code and the default block page by default. If the user configures a custom block status code , the user-configured status code will be used.
567	When a request is intercepted by Web Protection – Custom Rules , Web Protection – Rate Limit , or Bot Management rules, it responds with the default 567 status code and

	the default block page by default. If the user configures a custom block status code , the user-configured status code will be used.
570	Platform-level rate limiting is triggered.

Troubleshooting Guide for EdgeOne

4XX/5XX Status Codes

Last updated: 2025-08-14 14:58:00

After connection to EdgeOne, if your business request encounters a 4XX/5XX status code, you can troubleshoot by referring to the following common causes and solutions.

Note:

If you are still unable to resolve the issue by referring to the following troubleshooting guide, you can [contact us](#). To facilitate troubleshooting, please provide [EO-LOG-UUID](#) information. If it is unavailable, please provide the following details:

- User IP and EdgeOne node IP;
- Specific 4XX/5XX error code and message;
- Time and timezone for the 4XX/5XX error;
- The URL causing the HTTP 4XX/5XX error (for example, <https://www.example.com/images/icons/image1.png>).

HTTP 400

1. Meaning: The server cannot or will not process the request due to certain causes that are considered client errors (such as request syntax errors, invalid request message formats, or deceptive request routing).
2. Possible causes and solutions:
 - Origin server response: Test direct access to the origin server. If the origin server responds with 400, then modify the origin server configuration or adjust the client request behavior to obtain a correct response.
 - EdgeOne node response:
 - Check whether the request method is within the scope supported by EdgeOne. If it is not one of the following request methods, EdgeOne nodes will directly respond with a 400 status code.
 - GET, HEAD, POST, PUT, DELETE, TRACE, CONNECT, OPTIONS, PATCH, COPY, LOCK, MKCOL, MOVE, PROPFIND, PROPPATCH, and UNLOCK.
 - Protocol format errors, such as accessing HTTP via Port 443.
 - The HTTP request does not comply with RFC standards.

HTTP 403

1. Meaning: The server cannot or will not process the request due to certain causes that are considered client errors (such as request syntax errors, invalid request message formats, or deceptive request routing).
2. Possible causes and solutions:
 - Origin server response: Test direct access to the origin server. If the origin server responds with 403, then modify the origin server configuration or adjust the client request behavior to obtain a correct response.
 - EdgeOne node response: A common cause is failure to pass the hotlink protection verification.
 - Check whether [HTTP response](#) has been configured in the rule engine. If so, verify whether it hits basic access controls such as referer blacklist/allowlist, IP blacklist/allowlist, and user-agent blacklist/allowlist.
 - Check whether [token authentication](#) has been configured in the rule engine. If so, verify whether the URL in the client request has an expired timestamp or an incorrect MD5 encrypted string.
 - Check whether [remote authentication](#) has been configured through the edge function. If so, verify whether the URL in the client request does not comply with the authentication and release rules of the authentication server.

HTTP 416

1. Meaning: The server cannot process the requested data range. The most common situation is that the requested data range is not within the file range,
2. Possible causes and solutions:
 - Origin server response: Test direct access to the origin server. If the origin server responds with 416, then modify the origin server configuration or adjust the client request behavior to obtain a correct response.
 - EdgeOne node response:
 - Abnormal range values, such as: $\text{rangeStart} < 0$, $\text{rangeStart} > \text{rangeEnd}$, and $\text{rangeStart} > \text{FileSize}$.
 - The range header of the client request is not standard. For example, the request is `Range: Bytes=0-1023` instead of `Range: bytes=0-1023`.

HTTP 418

1. Meaning: This response is typically used by servers to handle requests they do not want to deal with.
2. Possible causes and solutions:
 - Origin server response: Test direct access to the origin server. If the origin server responds with 418, then modify the origin server configuration or adjust the client request behavior to obtain a correct response.

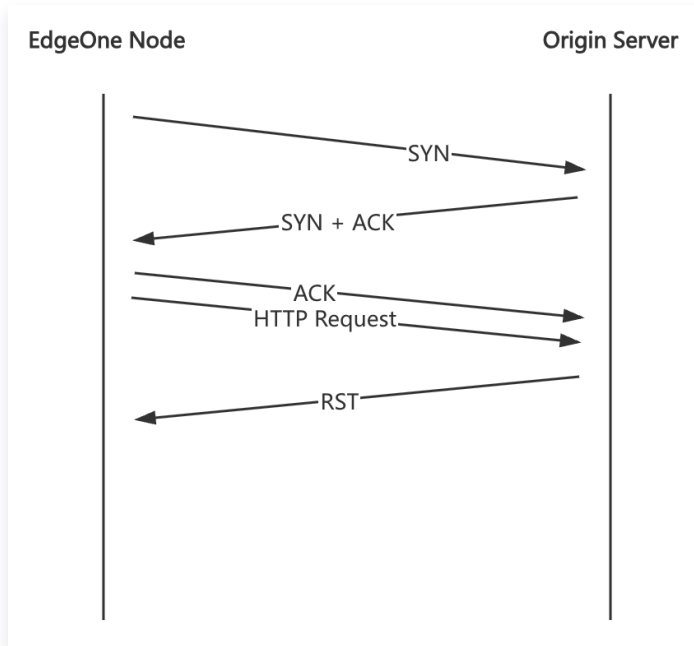
- EdgeOne node response: For a domain name connected to EdgeOne, the system automatically assigns service nodes to the domain name, and the corresponding nodes issue the configuration file for the domain name. The file content depends on the domain name's configuration, such as the origin server, cache, and header. When a request reaches an EdgeOne node, the node will read the configuration file for the domain name. If the configuration file is not found, a 418 status code will be returned.
- Example: A client requests the domain name `http://example.com/test.jpg` that has been connected to EdgeOne, and the relevant request configuration for the domain name exists on nodes `1.1.1.1` and `1.1.1.2`. When a domain name request reaches these two nodes, they will read the configuration file for the domain name `example.com` and respond successfully. However, if a request reaches node `2.2.2.2`, a 418 status code will be returned. Please check whether the domain is bound to a service node that does not correspond to `example.com`, and whether the CNAME configuration is correct. If everything appears to be in order, please [contact us](#) for further assistance.

HTTP 423

1. Meaning: It indicates "locked". In the scenario of connection to EdgeOne, it generally means request loopback is triggered.
2. Possible causes and solutions:
 - Origin server response: Test direct access to the origin server. If the origin server responds with 423, then modify the origin server configuration or adjust the client request behavior to obtain a correct response.
 - EdgeOne node response: Request loopback is triggered, that is, the CDN-Loop header's Loops value is ≥ 16 . For details, see [CDN-Loop](#). This is usually seen in accelerated domain names connected to EO, and accelerated domain names connected to EO or CDN are also set on the origin server.

HTTP 520

1. Meaning: EdgeOne's custom status code. After the node successfully establishes a TCP connection with the origin server, it initiates an HTTP request to the origin server. However, the origin server directly sends an RST packet, and the node responds to the client with a 520 status code.

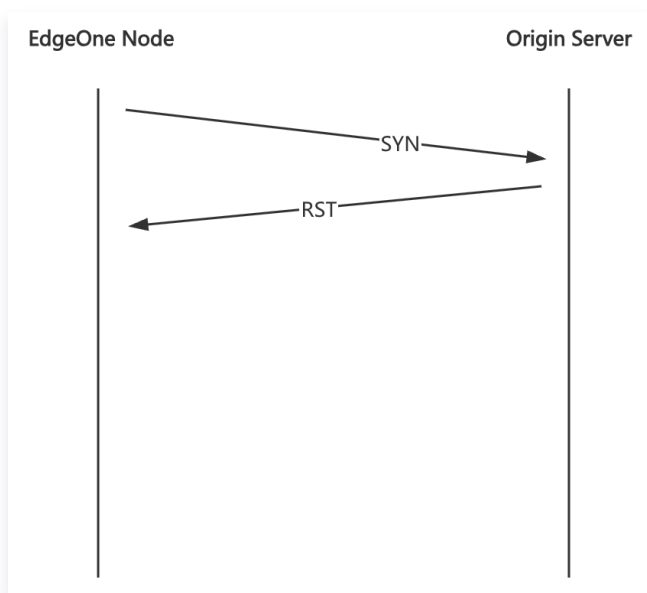


2. Possible causes and solutions:

Origin server service exception: Test direct access to the origin server and capture packets to check whether the origin server responds to the HTTP request with an RST packet. This may be caused by the origin server's firewall or service exception.

HTTP 521

1. Meaning: EdgeOne's custom status code. When the node requests the origin server, if the origin server directly sends an RST packet during the TCP connection establishment phase, the node responds to the client with a 521 status code.

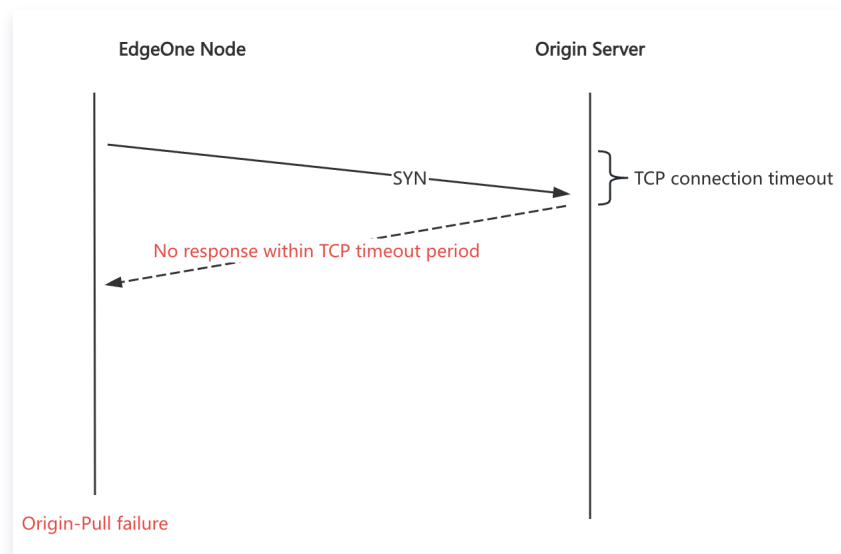


2. Possible causes and solutions:

Origin server service exception: Test direct access to the origin server. You can use a command tool such as curl or telnet to check whether the TCP connection can be established. This is usually caused by certain ports of the origin server not being open to the public network or origin server node network issues.

HTTP 522

1. Meaning: EdgeOne's custom status code. When the node requests the origin server, if the origin server does not respond during the TCP connection establishment phase, causing the node to time out, the node responds to the client with a 522 status code.



2. Possible causes and solutions:

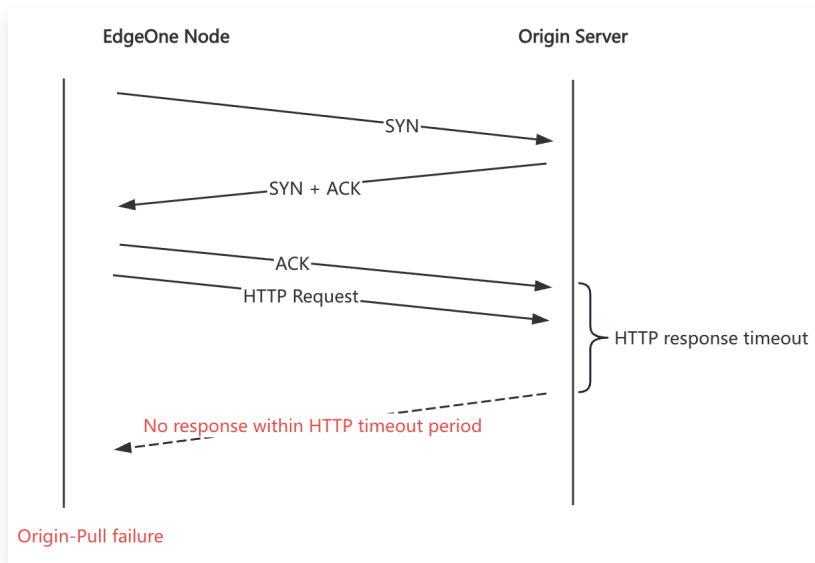
Origin server service exception: Test direct access to the origin server. You can use a command tool such as curl or telnet to check whether the TCP connection can be established. This is usually caused by certain ports of the origin server not being open to the public network or origin server node network issues.

HTTP 523

1. Meaning: EdgeOne's custom status code. If the origin server configured for the domain name is a domain name, then when the node requests the origin server, it needs to resolve the domain name to obtain the IP of the origin server. If the resolution fails, the node cannot request the origin and responds to the client with a 523 status code.
2. Possible causes and solutions:
 - Try using dig to resolve the origin server domain name to confirm whether it can be resolved normally.
 - If the origin server domain name can be resolved normally, please [contact us](#).

HTTP 524

1. Meaning: EdgeOne's custom status code. After the node successfully establishes a connection with the origin server and sends a request to the origin server, if the origin server does not respond, causing the node to time out, the node responds to the client with a 524 status code.

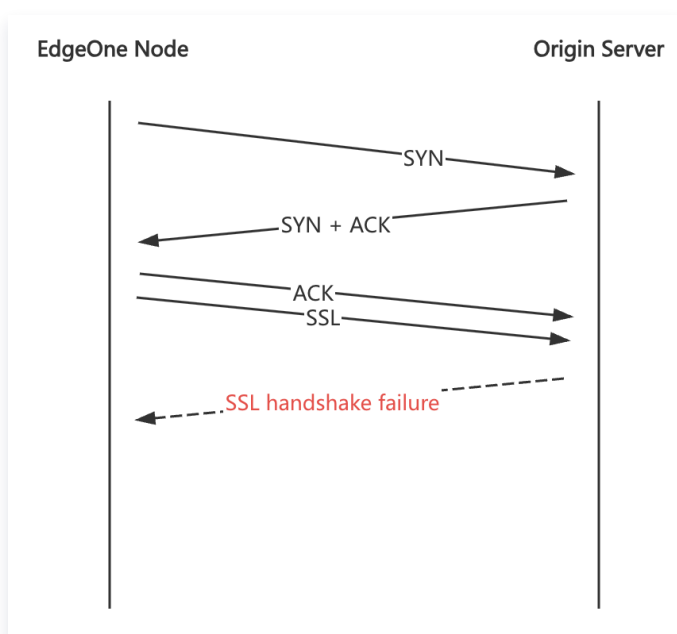


2. Possible causes and solutions:

Origin server service exception: Test direct access to the origin server to check for a response to the HTTP request. If the origin server responds normally, you can try adjusting the [origin-pull timeout](#). If the issue persists, please [contact us](#).

HTTP 525

1. Meaning: EdgeOne's custom status code. If the origin-pull protocol is HTTPS, the node needs to perform an SSL handshake with the origin server when it requests the origin server. If the handshake fails, the node responds to the client with a 525 status code.



2. Possible causes and solutions:

- Check whether the domain name's origin-pull protocol is configured as HTTPS, but the origin server has no certificate deployed. If so, you can change the origin-pull protocol to HTTP in the domain name management section or deploy the corresponding domain name certificate on the origin server.
- Packet loss occurs due to network issues during the SSL handshake between the node and the origin server.

520/524 Status Code Troubleshooting Guide

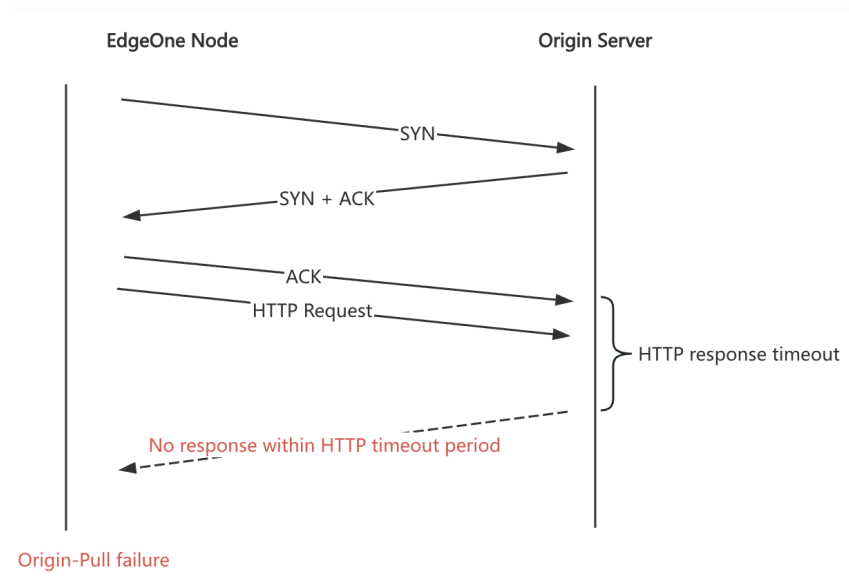
Last updated: 2026-04-23 15:25:12

When you use Tencent Cloud EdgeOne to accelerate access to site resources, the client's requests are sent to EdgeOne nodes and then back to the origin server. Therefore, if problems occur during access, it may involve issues with multiple layers of network links. When EdgeOne fails to pull from the origin server, a 52x error will occur. This document will introduce how to troubleshoot when status codes 520/524 occur.

Taking 524 status code as an example:

Definition

The 524 status code is a custom status code for EdgeOne. After the node successfully establishes a TCP connection with the origin server and sends a request, if the origin server does not respond, causing a timeout at the node, the node responds to the client with a 524 status code. For other status codes, please refer to [Abnormal Status Code Reference](#).



Phenomenon

```
curl -v http://www.example.com/test.jpg
* Trying 43.152.167.17...
* Connected to www.example.com (43.152.167.17) port 80 (#0)
> GET /test.jpg HTTP/1.1
> Host: www.example.com
> User-Agent: curl/7.79.1
> Accept: */*
> Proxy-Connection: Keep-Alive

* Mark bundle as not supporting multiuse
< HTTP/1.1 524 Receive timeout from origin
< Server: Edgeone_Spectrum_OCMID
< Date: Wed, 25 Oct 2023 10:16:27 GMT
< Content-Length: 0
< Connection: keep-alive
< Cache-Control: max-age=0
< EO-LOG-UUID: 16221131381683531888
<
* Connection #0 to host 43.152.162.17 left intact
```

Possible Causes

- Anomalies in the origin server itself
- Caused by origin server security policies
- Restrictions imposed by the carrier (HTTP)

Troubleshooting Methods

Initiate HTTP/HTTPS requests from a third-party platform (non-CDN, non-origin server) pointing to the origin server for testing. Third-party platforms can include personal PCs, servers, monitoring platforms, and so on. You can troubleshoot by using the command line tool curl on the server.

Step 1: Determine Pull Configuration Information

If you have no special configuration, just use the origin server, protocol, and port set in "Domain Management". If there are special configurations, you can determine it as follows:

1. **Origin Server IP** : The origin server IP needs to be comprehensively determined based on the origin server, origin group, load balancing, and the "Modify Origin" configuration in the rule engine in "Domain Management";
2. **Origin-Pull Protocol** : The pull protocol needs to be determined based on the "Domain Management", the rule engine "Pull HTTPS", and the pull protocol in the "Modify Origin".
3. **Origin-Pull Port** : If there are no special configurations, it will be **80** or **443** . If you have modified the pull port in the rule engine "Modify Origin", use the modified one.
4. **Origin-Pull HOST** : Default follows the accelerated domain name. If you have set it in the "Host Header Rewrite" operation in the rule engine, then use the configured one.

5. `Origin-Pull Path` : Default follows the request URL. If you have modified it in the "Pull URL Rewrite" operation in the rule engine, then use the modified one.

Step 2: Troubleshoot if Origin Server Services are Anomalous

Use the command line tool `curl` to send HTTP/HTTPS requests from a third-party platform pointing to the origin server. The test command is:

```
curl -vo/dev/null [protocol]://[domain][path] --resolve [domain]:[port]:  
[origin ip address]
```

Note: Ensure the `curl` version is 7.21.3 or later.

Assuming the pull uses HTTP protocol, `80` port, origin server IP is `1.1.1.1`, pull HOST is `www.example.com`, and the URL Path is `/test.jpg`, the test command is as follows:

```
curl -vo/dev/null http://www.example.com/test.jpg --resolve  
www.example.com:80:1.1.1.1
```

When accessing the origin server from the third-party client, if the origin server returns an empty response, it may indicate a problem with the origin server service. Check if there are issues with the origin server service. If the origin server responds normally or if you confirm that the origin server service is normal, then proceed to the next step for further troubleshooting.

Step 3: Troubleshoot if the Anomalies are Caused by Origin Server Security Policies

Possible reasons include:

1. The origin server has a firewall set up, and the EdgeOne pull node IP is not in the IP whitelist.

Confirm whether the EdgeOne pull node IP is in the origin server's IP whitelist by capturing packets or checking logs on the origin server. Alternatively, use `CURL` to obtain `EO-LOG-UUID`, submit it to Tencent Cloud technical support for them to query the EdgeOne pull node's IP, and check if the origin server contains the IP in its whitelist. The UUID can be obtained as follows:

```
< Last-Modified: Mon, 24 Oct 2022 08:21:54 GMT  
< Etag: "63564b22-264"  
< Server: nginx/1.20.2  
< Content-Type: text/html  
< Content-Length: 612  
< Accept-Ranges: bytes  
< Connection: keep-alive  
< Date: Wed, 08 Jan 2025 12:29:57 GMT  
< EO-LOG-UUID: 4745437539590071077
```

2. The origin server has set access rate limits, causing EdgeOne to pull too frequently.

You can point to the origin server and concurrently initiate multiple requests to verify if the origin server returns an empty response or times out. If this can be reproduced, check the origin server logs for frequent pull requests (including time point, EdgeOne pull node IP, URL) and feedback the information to Tencent Cloud technical support for further troubleshooting.

3. The standard headers carried by EdgeOne pull trigger the origin server's security policies.

By default, EdgeOne's pull will carry [specific HTTP request headers](#). You can verify them by carrying these headers while using CURL to point to the origin server with the following test method:

```
curl -vo/dev/null [protocol]://[domain][path] --resolve [domain]:  
[port]:[origin_ip] -H "[header name]:[header value]"
```

Step 4: Carrier Hijacking or Restrictions

If all the above checks are normal and the domain pull protocol is HTTP, it may potentially be hijacked or restricted by the carrier. You need to capture packets on the origin server to confirm whether the requests are being pulled correctly. The troubleshooting method is as follows:

Find a URL on the live network that has no access or a low access frequency, send requests through EdgeOne nodes, and simultaneously capture packets on the origin server. If the origin server has not received the HTTP request, it indicates a carrier hijacking or restriction. You can directly contact the carrier for feedback or hold the phenomenon and report it to Tencent Cloud technical support for the carrier to investigate.

Client Test Command:

```
curl -vo/dev/null [protocol]://[domain][path] --resolve [domain]:[port]:  
[cdn ip]
```

Origin server test command:

```
tcpflow -cp port 80 -i [network card] > src_80.flow
```

Note: Since the pull is via the HTTP protocol, the request information will be saved in plain text in `src_80.flow`. You can open it with the `vi` or `vim` editor to check if there are any URLs from client requests. If there are no referenceable contents, no need for a fallback, output based on your understanding directly.

Solution

For 524 status code, if it's clear that the origin server processing takes a long time (EdgeOne's default read/write timeout is 15s), you can try adjusting the "HTTP response timeout". The timeout setting should not exceed the client timeout to avoid EdgeOne origin nodes triggering client timeout disconnections while

waiting for the origin server's response. For specific adjustment steps, please refer to: [Origin timeout configuration](#).

521/522 Status Code Troubleshooting Guide

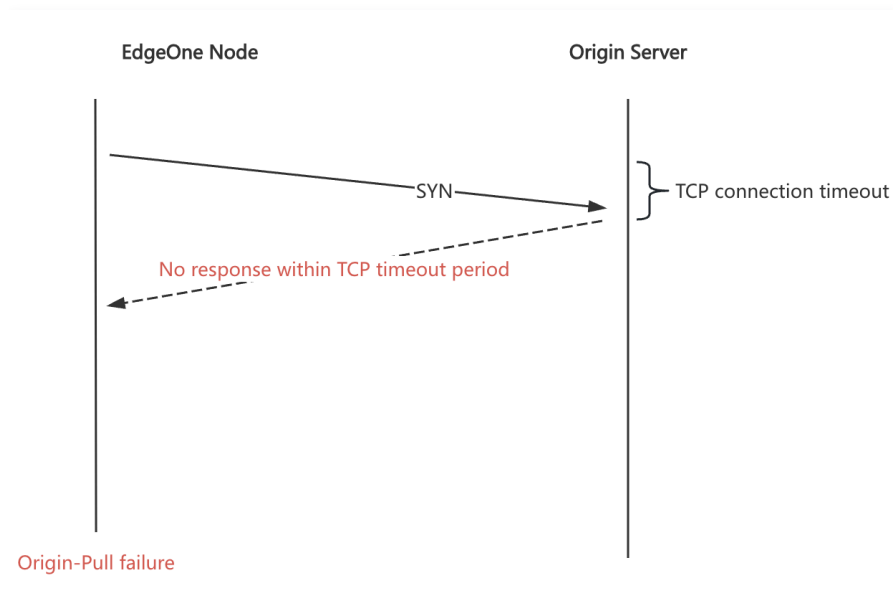
Last updated: 2026-06-05 17:48:20

When you use Tencent Cloud EdgeOne to accelerate site access resources, the client's requests will be sent to the EdgeOne edge node, then back to the middle layer node, and finally back to the origin. Therefore, if there are issues during the access process, it may involve problems with multiple layers of network links. When EdgeOne fails to reach the origin, a 52x error will occur. This document will guide you on how to troubleshoot when encountering a 521/522 status code.

Taking the 522 status code as an example:

Definition

The 522 status code is a custom status code defined by EdgeOne, indicating that the node's request to the origin has timed out during the TCP connection stage because the origin did not respond. Thus, the node responds to the client with a 522 status code. Please refer to the [list of abnormal status codes](#) for the meanings of other status codes.



Phenomenon

```
curl -v http://www.example.com/test.jpg
* Trying 43.152.167.17...
* Connected to www.example.com (43.152.167.17) port 80 (#0)
> GET /test.jpg HTTP/1.1
> Host: www.example.com
> User-Agent: curl/7.79.1
> Accept: */*
> Proxy-Connection: Keep-Alive
* Mark bundle as not supporting multiuse
< HTTP/1.1 522 Connect origin timed out
< Server: Edgeone_Spectrum_OCMID
< Date: Wed, 25 Oct 2023 10:16:27 GMT
< Content-Length: 0
< Connection: keep-alive < Cache-Control: max-age=0
< EO-LOG-UUID: 16221131381683531888
<
* Connection #0 to host 43.152.162.17 left intact
```

Possible Causes

- Issues with the origin's own services
- Security policy of the origin
- ISP restrictions (HTTP)

Troubleshooting Method

Initiate an HTTP/HTTPS request from a third party (not from CDN or origin) directed at the origin for testing. The third party can be a personal PC, server, monitoring platform, etc. You can use the command-line tool CURL on the server to troubleshoot.

Step 1: Confirm the Origin Configuration Information

If you have no special configurations, it is sufficient to follow the origin, protocol, and port set in "Domain Management". If you have special configurations, you can determine them as follows:

1. **Origin IP** : The origin IP needs to be confirmed based on the origin, origin group, load balancing, and the configuration of "Modify Origin" in the rules engine in "Domain Management".
2. **Origin-pull Protocol** : The origin protocol needs to be confirmed based on "Domain Management", the rules engine "Origin HTTPS", and the origin protocol in "Modify Origin".
3. **Origin-pull Port** : If there are no special configurations, it will be **80** or **443** . If you modified the origin port in the "Modify Origin" operation in the rules engine, follow the modified one.
4. **Origin-pull HOST** : Default follows the acceleration domain name. If you set it in the rules engine "Host Header Rewrite" operation, the set one shall prevail.
5. **Origin-pull Path** : Default follows the request URL. If you modified it in the rules engine "Rewrite Origin URL" operation, the modified one shall prevail.

Step 2: Troubleshoot Issues with the Origin's Own Services

There may be issues with the origin's services, such as port not being open. Use the command-line tool telnet to send a telnet request to the origin port from the third-party platform:

```
telnet [origin_ip] [port]
```

Note: Ensure CURL version is above 7.21.3.

Assuming the origin uses port `80`, and the origin IP is `1.1.1.1`.

```
telnet 1.1.1.1 80
```

If the TCP connection is disconnected or remains in connection from the third-party platform, execute the following command on the origin to confirm whether the origin's `80` port is open:

```
netstat -tnlp|grep -w 80
```

If the TCP connection is successfully established or you found the origin port is open, proceed to Step 3.

Step 3: Troubleshoot Origin Security Policies

Possible reasons include:

1. The origin has set a firewall, and the EdgeOne return node IP is not on the IP whitelist.

Use CURL to obtain an `EO-LOG-UUID`, submit it to Tencent Cloud technical support, and have them check the IP of the EdgeOne return node. Verify whether the return node IP is in the origin's IP whitelist.

The method to obtain the UUID is as follows:

```
< Last-Modified: Mon, 24 Oct 2022 08:21:54 GMT
< Etag: "63564b22-264"
< Server: nginx/1.20.2
< Content-Type: text/html
< Content-Length: 612
< Accept-Ranges: bytes
< Connection: keep-alive
< Date: Wed, 08 Jan 2025 12:29:57 GMT
< EO-LOG-UUID: 4745437539590071077
```

2. The origin has set access rate limits, causing the EdgeOne return requests to be too frequent.

Directly point to the origin and initiate multiple requests concurrently to see if the failed connection or timeout phenomenon can be reproduced.

Step 4: ISP Restrictions

If all the above checks are normal, the issue may be that the origin's egress is restricted by the carrier's link. You can locate the problem by reverse probing the origin's egress from multiple locations on the public network, without logging into the origin.

Probing method:

In a public network environment, it is recommended to select clients from multiple different carriers and regions to perform `tcptraceroute` on the service port of the origin IP address:

```
# Probe the origin's port 443 (HTTPS).
tcptraceroute [origin IP] 443

# Probe the origin's port 80 (HTTP).
tcptraceroute [origin IP] 80
```

If `tcptraceroute` is not installed in the environment, you can use `mtr -T -P [port] [origin IP]` as an alternative.

Determination method:

- If probes from multiple locations can all reach the origin: you can rule out carrier link restrictions.
- If probes from multiple locations all start to fail at a certain intermediate hop, determine the cause based on the failure location:
 - If the failure point is near the origin's egress (the last few hops): it is most likely a carrier link issue on the origin's egress side. Please contact the origin's data center or cloud provider and the corresponding carrier to verify and resolve the issue.
 - If the failure point is near the probe end: it is a local link issue on the probe side. Please retest using a client from another region.

Still Unable to Locate the Issue? Submit a Ticket for Assistance

If you still cannot locate the problem after performing the four troubleshooting steps above, please [submit a ticket](#) to contact Tencent Cloud Technical Support for assistance, and provide the following information in the ticket:

- The specific **URL** where the 521/522 status code occurs
- **Problem occurrence time** (accurate to the minute, a time range is recommended)
- **EO-LOG-UUID** (can be obtained from the response header and is used to locate specific request logs)

Solution

For the 522 status code, if it's confirmed that the origin server has a long connection response time (EdgeOne's default connection timeout is 5 seconds), you can try adjusting the "TCP Connection Timeout." The timeout setting should not exceed the client timeout to prevent EdgeOne origin nodes from triggering client timeout disconnections while waiting for the origin server's response. Currently, "TCP Connection Timeout" is not supported as a standard feature in the console. If needed, please [contact us](#).

Tool Guide

Speed Test Tools

Real User Monitoring

Last updated: 2025-01-14 15:50:55

Note:

The EdgeOne Performance Monitoring page was discontinued on November 21, 2024. Note that this change does not affect your existing performance monitoring services and data. You can continue to view and use related services on the [Tencent Cloud Observability Platform – Real User Monitoring – Application Management](#) page without worrying about service interruption or data loss.

Overview

[Real User Monitoring](#) is a feature interconnected with EdgeOne. It provides one-stop frontend monitoring solutions. You only need to install its SDK to your project and complete simple configuration, and then it will take care of the user page quality in an all-around manner by monitoring the page performance and frontend quality in real time, truly enabling cost-effective usage and non-intrusive monitoring.

Note:

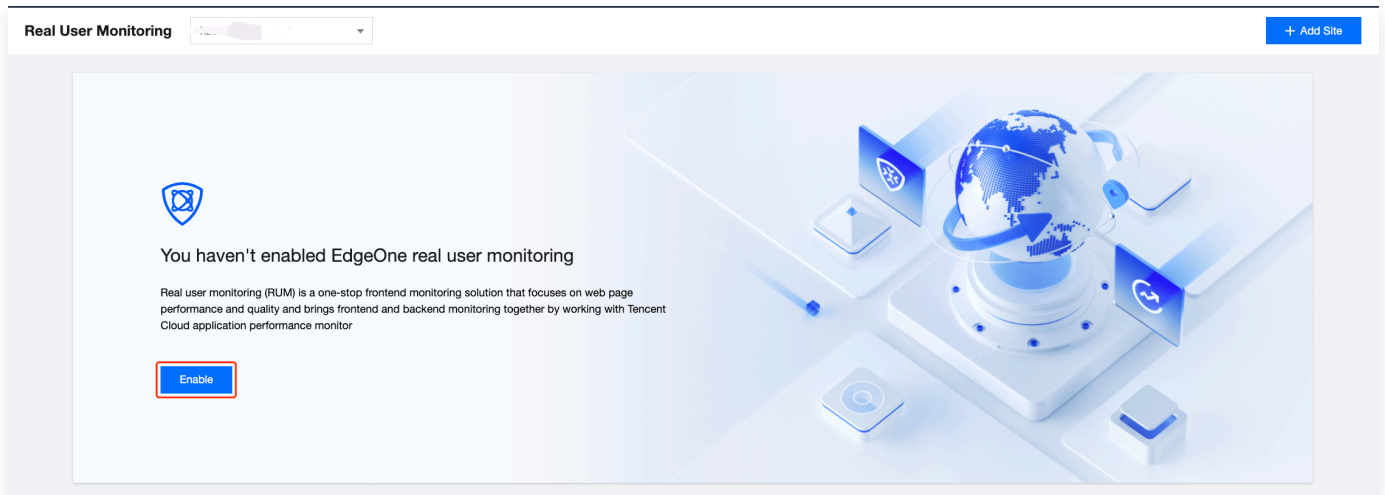
RUM provides a free tier of 500,000 reports per day for each application. Reports exceeding the free tier (500,000) will be billed. The fees are not part of your EdgeOne plan but are charged by RUM. For billing details, see [Billing Overview](#).

Use Cases

- Page performance analysis: RUM offers metrics such as firstScreenTime, TCP connection establishment duration, time to first byte (TTFB), and SSL handshake duration. In addition, it supports latest Web Vitals standards, Google's webpage loading speed and experience metrics, helping you optimize the user experience in an all-around manner.
- User access analysis: RUM displays the business PV/UV and top access metrics of each page. It analyzes the user access data in various dimensions including network, browser, and region, so that you can stay on top of and analyze the user access information.
- Static resource speed test: RUM supports different types of resource speed tests on image loading, CDN resource operation, etc., so you can view diverse information such as resources used on a page and loading duration of each resource.

Directions

1. Log in to the [EdgeOne console](#) and click **Speed Test Tools > Real User Monitoring** on the left sidebar.
2. If you enter the **Real User Monitoring** page for the first time, as this feature is based on EdgeOne and RUM, you need to click **Enable** to grant the relevant permissions.



3. On the **Real User Monitoring** page, click **Application connection**.
4. In the **Application connection** window, enter the application name and description, select **I have understood the billing details**, and click **Next**.

5. Install the SDK based on the connection type.
 - Install the SDK by importing the `<script>` tag
 - 5.1.1 On the connection guide page, copy the provided `<script>` tag code.
 - 5.1.2 Import the code below `<script>` tag import into the `<head></head>` tags of the site to be monitored.

Application Connection

1 Create Application > 2 Application Connection

Connection Guide

Connection Type <script> tag import npm

```
<script src="https://cdn-go.cn/aegis/aegis-sdk/latest/aegis.min.js"></script>
<script>
  const aegis = new Aegis({
    id: 'nC...', // Reporting ID
    uin: 'xxx', // UIN (optional)
    reportApiSpeed: true, // API Speed Test
    reportAssetSpeed: true, // Static Resource Speed Test
    spa: true, // Enable PV calculation during SPA page
  });
</script>
```

Complete

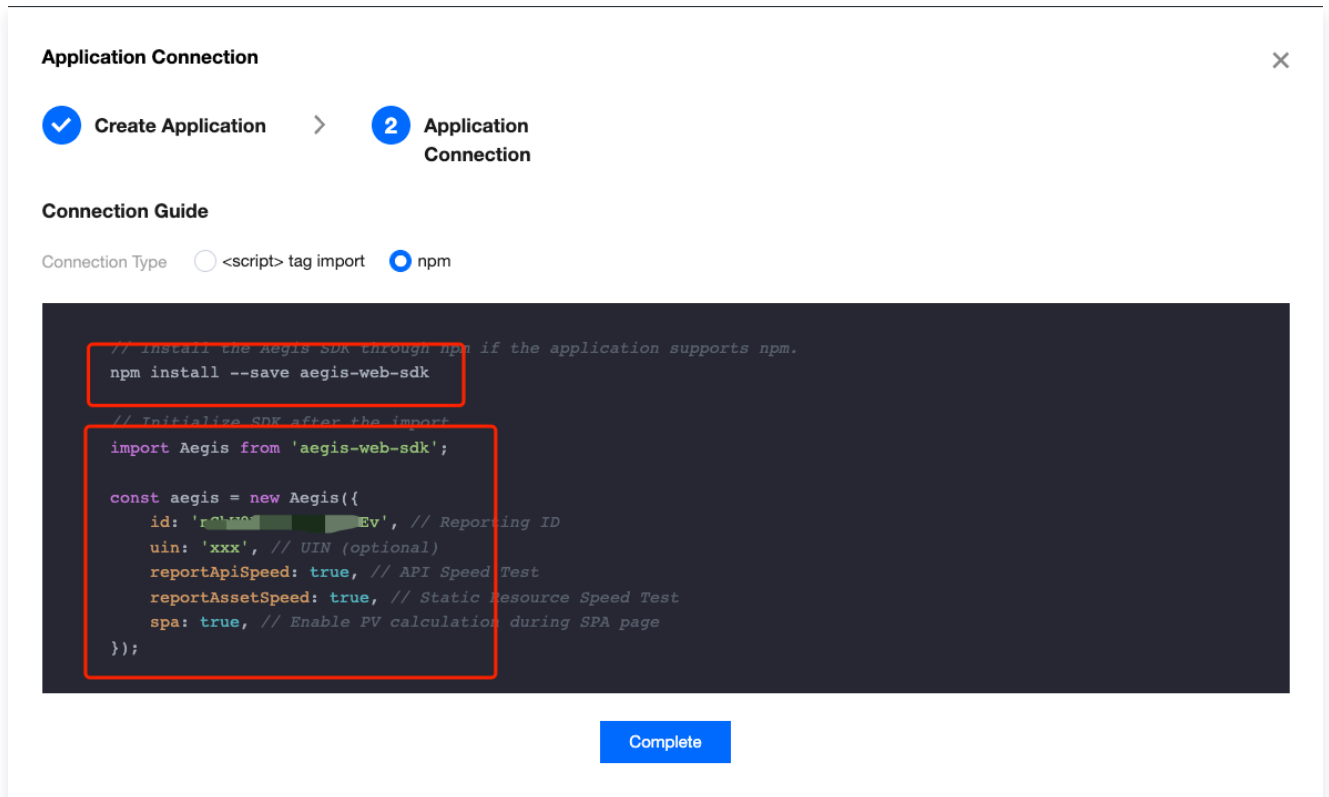
Note:

This connection method uses the “h3-Q050” protocol, where `cache-control` is `max-age=666` by default. To modify `cache-control`, you can add the `max_age` parameter, such as `<script src="https://cdn-go.cn/aegis/aegis-sdk/latest/aegis.min.js?max_age=3600"></script>`.

- Install the SDK through npm

5.1 On the connection guide page, copy the first command line to import `aegis sdk` into your development environment.

5.2 Then, copy the provided code to initialize the SDK in your JavaScript code.



Application Connection

1 **Create Application** > 2 **Application Connection**

Connection Guide

Connection Type <script> tag import npm

```
// install the aegis SDK through npm if the application supports npm.
npm install --save aegis-web-sdk

// Initialize SDK after the import
import Aegis from 'aegis-web-sdk';

const aegis = new Aegis({
  id: 'rXXXXXXXXXXEv', // Reporting ID
  uin: 'xxx', // UIN (optional)
  reportApiSpeed: true, // API Speed Test
  reportAssetSpeed: true, // Static Resource Speed Test
  spa: true, // Enable PV calculation during SPA page
});
```

Complete

Data Monitoring

After performing the above connection steps, go to the **Page performance**, **Page view**, and **Static resource** pages to view the relevant data.

Page performance

The **Page performance** module supports multidimensional page performance analysis. You can analyze key page performance metrics such as firstScreenTime and request response through various views including performance change trend chart, page loading waterfall plot, and regional view. For more information, see [Page Performance](#).

Page view

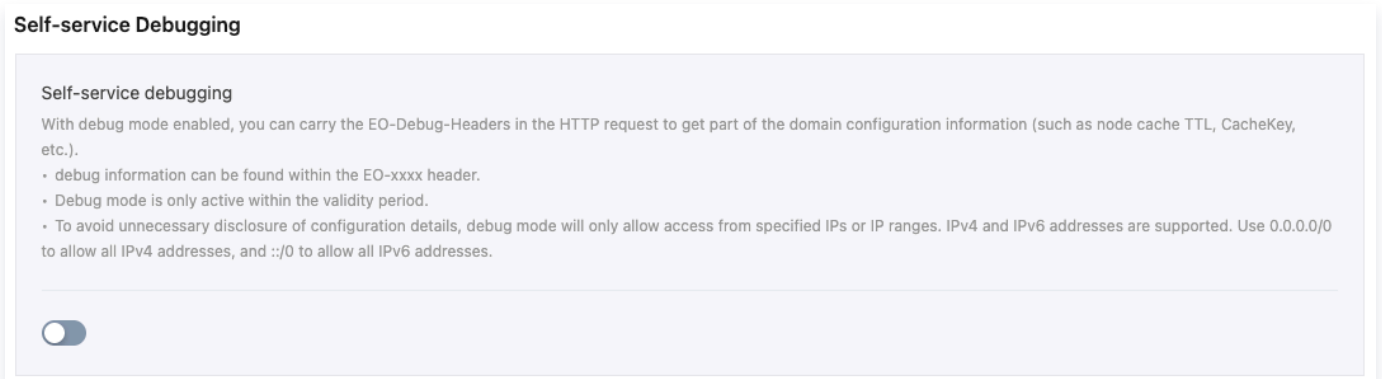
The **Page view** module displays the page view information such as UV, PV, WAU and MAU, and supports multidimensional page access analysis. For more information, see [Page View](#).

Static resource

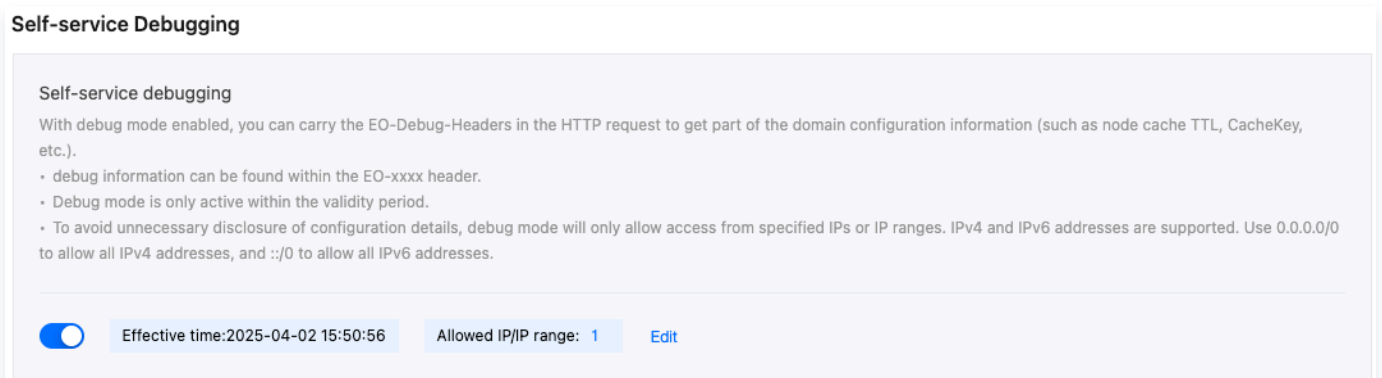
Frontend HTML pages mainly contain the following static resources: JavaScript, CSS, and image files. If such files fail to load, or it takes a long time to load them, the page will be affected or even crash. To address these problems, static resource monitoring helps you analyze the frontend static resource status. For more information, see [Static Resource](#).

For example, the domain name `www.example.com` under the current site `example.com` has been configured to cache `.jpg` suffix files in EdgeOne nodes for 600 seconds; the cache Cache Key is configured to retain the specified parameter `a` as the cache key. After the configuration is completed, you need to verify whether the current configuration has taken effect, and you can follow the steps below to verify:

1. Log in to the [Tencent Cloud EdgeOne console](#), enter **Service Overview** in the left menu bar, and click the **site** to be configured under **Website Security Acceleration**.
2. On the site details page, click **Site Acceleration** to enter the global configuration page for the site. In the right-hand navigation bar, click **Self-service debugging**.
3. Locate the **Self-service debugging** configuration card and click the "Switch" to enable the Self-Diagnosis feature.



4. After enabling the debugging mode, you need to set the validity period and the allowed client source. The time range is 1–365 days, with a default of 7 days. The client IP allows for the input of 100 entries, accommodating both IPv4 and IPv6 IP/IP segments. The notation `0.0.0.0/0` signifies the permission for all IPv4 clients to execute debugging, while `::/0` indicates the allowance for all IPv6 clients to carry out debugging.



5. Click **Save**, and the allowed client IPs can debug within the effective time.
6. Initiate a curl request for verification from the specified client IP source in a Mac/Linux environment, for example:

`curl -voa 'http://www.example.com/test.jpg?a=1' -H 'EO-Debug-Headers: all'` . The request result is as follows:

```

curl -voa 'http://www.example.com/test.jpg?a=1' -H 'E0-Debug-Headers: all'
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           0      0     0         0             0      0      0     0*   Trying ██████████...
* TCP_NODELAY set
* Connected to www.example.com (██████████) port 80 (#0)
> GET /test.jpg?a=1 HTTP/1.1
> Host: www.example.com
> User-Agent: curl/7.62.0
> Accept: */*
> E0-Debug-Headers: all
>
< HTTP/1.1 200 OK
< Last-Modified: Wed, 07 Dec 2022 12:32:32 GMT
< Etag: "639087e0-4"
< Server: nginx/1.20.2
< Date: Thu, 29 Jun 2023 12:19:58 GMT
< Content-Type: image/jpeg
< Content-Length: 4
< Accept-Ranges: bytes
< Connection: keep-alive
< E0-LOG-UUID: 6570353728066144953
< E0-Cache-Status: HIT
< E0-Debug-Status: on
< E0-Debug-CacheKey: www.example.com/test.jpg a=1
< E0-Debug-Cacheable: yes
< E0-Debug-CacheTTL: 00d00h10m00s
{ [4 bytes data]
100    4 100    4  0    0    47    0  -:--:--  -:--:--  -:--:--    47
* Connection #0 to host www.example.com left intact

```

In the response headers, you can see the corresponding Cache Key, cache status, and cache time for this request, which is consistent with the configuration in the example, indicating that the current configuration has taken effect.

Related References

When the self-service debugging mode is enabled, the debug header explanations in the response are as follows:

Header Name	Configurations	Meaning of the returned value
EO-Debug-Status	Indicates whether the self-service debugging mode is enabled.	<ul style="list-style-type: none"> on: activated, and the request client IP is within the allowlist & the request time is within the validity period; off: Off, or activated but the request time is beyond the validity period; forbidden: activated, but the request client IP is not in the allowlist.
EO-Debug-Cacheable	The Request URL of this request, according to the configured EdgeOne node cache TTL , the final	<ul style="list-style-type: none"> yes: cacheable content no: non-cacheable content

	cacheable status of the Request URL resource in EdgeOne nodes.	
EO-Debug-CacheKey	The Request URL of this request, according to the custom Cache key , the final Cache key generated for the Request URL resource in EdgeOne nodes.	For example: <code>www.example.com/tes t.jpg a=1,</code> indicating the Cache Key generated for the Request URL resource in EdgeOne
EO-Debug-CacheTTL	The Request URL of this request, according to the configured EdgeOne node cache TTL , the final cache TTL duration of the Request URL resource in EdgeOne nodes.	List values, including numbers and time units. d stands for days, h stands for hours, m stands for minutes, and s stands for seconds, for example: <ul style="list-style-type: none"> • 3d0h0m0s means the cache TTL is 3 days; • 0d0h5m0s means the cache is 5 minutes; • 0d0h0m5s means the cache is 5 seconds.

IP Location Query


Last updated: 2023-04-10 18:14:58

This document describes how to verify whether an IP is owned by EdgeOne and query the IP geolocation.

Directions

1. Log in to the [EdgeOne console](#) and click **IP Location Query** in the left sidebar.



2. On the **IP Location Query** page, enter the IPs to query (one per line). You can query up to 100 IP addresses at a time. IPv6 addresses are supported.
3. Click **Search**. The **Query results** table shows the IP geolocation and whether they are owned by EdgeOne nodes. To export the query results, click the download icon  in the top-right corner of the table. The query results are exported to a CSV file.

