Tencent Cloud

Real User Monitoring

FAQs

Product Documentation

Contents

FAQs
Product

Last updated：2024-01-22 19:25:42

What is the purpose of the Aegis SDK?

The Aegis SDK is a data reporting SDK embedded into the user page or installed in the user code through npm. It mainly collects the performance and quality data of the user side.

How long will data be retained?

Raw data reported by users such as error, custom reporting, and page view data will be retained for 30 days. Performance metric data such as page performance, API monitoring, and static resource monitoring data will be retained for 15 days.
Data calculated by RUM at a fixed time everyday such as daily project score and daily total PV/UV data will be retained for 90 days.

Which platforms does RUM currently support and provide SDK for?

RUM currently supports data reporting on various platforms such as web, Hippy, mini program (WeChat and QQ), and React Native. It will provide SDKs for more platforms in the future.

What is offline log?

Offline log is a solution where most logs are stored locally and will only be reported to RUM when necessary.

Troubleshooting

Last updated：2024-01-22 19:25:42

The Aegis SDK failed to capture `throw new Error` in the code. What should I do?

In Vue framework, `Vue.config.errorHandler` actively captures errors. You can use it to capture errors and call `aegis.error()` for reporting.

How do I fix the error 403 of API request?

API error 403 is generally caused by a page domain verification failure. You can check whether the domain set during project creation is the same as the actual reporting domain. If the project doesn't require verification, you can enter `*` for the domain.

Note:

If you are not sure what the reporting domain is, you can enter `location.host` in the browser to view it. For non-web projects, `*` is entered by default, indicating that no verification is required.

After creating a project in RUM, you will get a string containing 18 characters, which is the reporting ID and will be passed in when you run `new Aegis`. If the ID is incorrect, the error 403 will also be reported.

Aegis initialization

```
new` `Aegis({`` ``id: ``'pGUVFTCZyewhxxxxxx'``})
```

Note:

After the project is created, RUM data sync will take about 1–2 minutes.

How do I fix image, JavaScript, or CSS loading failures in logs?

1. Access the resource to check whether it is exceptional.

2. If the access is normal but logs still contain many exceptions, proceed to the next step.

3. Check whether the exceptions occurred just on a specific region or ISP, for example.

4. If this is not the case, the exceptions are most likely caused by user network problems. You can try including a PWA or offline package into your project to reduce such problems. Essentially, such problems on the frontend cannot be completely avoided.

After I connect my project to the SDK, why the `aegis.qq.com/speed/webvitals` API is canceled during page refresh?

The value of CLS in Web Vitals must be calculated when the page visibility is changed. When the user refreshes a page, if `Preserve log` is selected, a request will be processed and sent, but if it is canceled by the page refresh, this error will occur, which has no impact on the user and data and can be ignored.

Many error logs show "Script error. @ (:0:0) No specific information". Are they caused by a third-party JavaScript exception? Can I filter them out through configuration?

A script error is also called a CORS error. When a website requests and executes a script that isn't under its own domain, if an error occurs in the cross-origin script, this error may be reported. As scripts in projects are stored in CDN, this type of error is most common.

Actually, this is not a JavaScript bug; however, for security considerations, the browser will purposely hide the specific error messages reported by JavaScript files in other origins. Doing so can effectively prevent sensitive information from being unintentionally captured by uncontrolled third-party scripts. Therefore, the browser only allows scripts in the same origin to capture specific error messages, while scripts in other origins only know that an error occurs but cannot know the specific error information. For more information, see ScriptExecutionContext.cpp.

```
289  bool ScriptExecutionContext::sanitizeScriptError(String& errorMessage, int&
290  {
291      KURL targetURL = completeURL(sourceURL);
292      if (securityOrigin()->canRequest(targetURL))
293          return false;
294      errorMessage = "Script error.";
295      sourceURL = String();
296      lineNumber = 0;
297      return true;
298  }
299
```

Solutions:

Solution 1: CORS

Step 1. Add the `crossorigin` attribute to the resource.


```
<script src="http://another-domain.com/app.js" crossorigin="anonymous"></script>
```
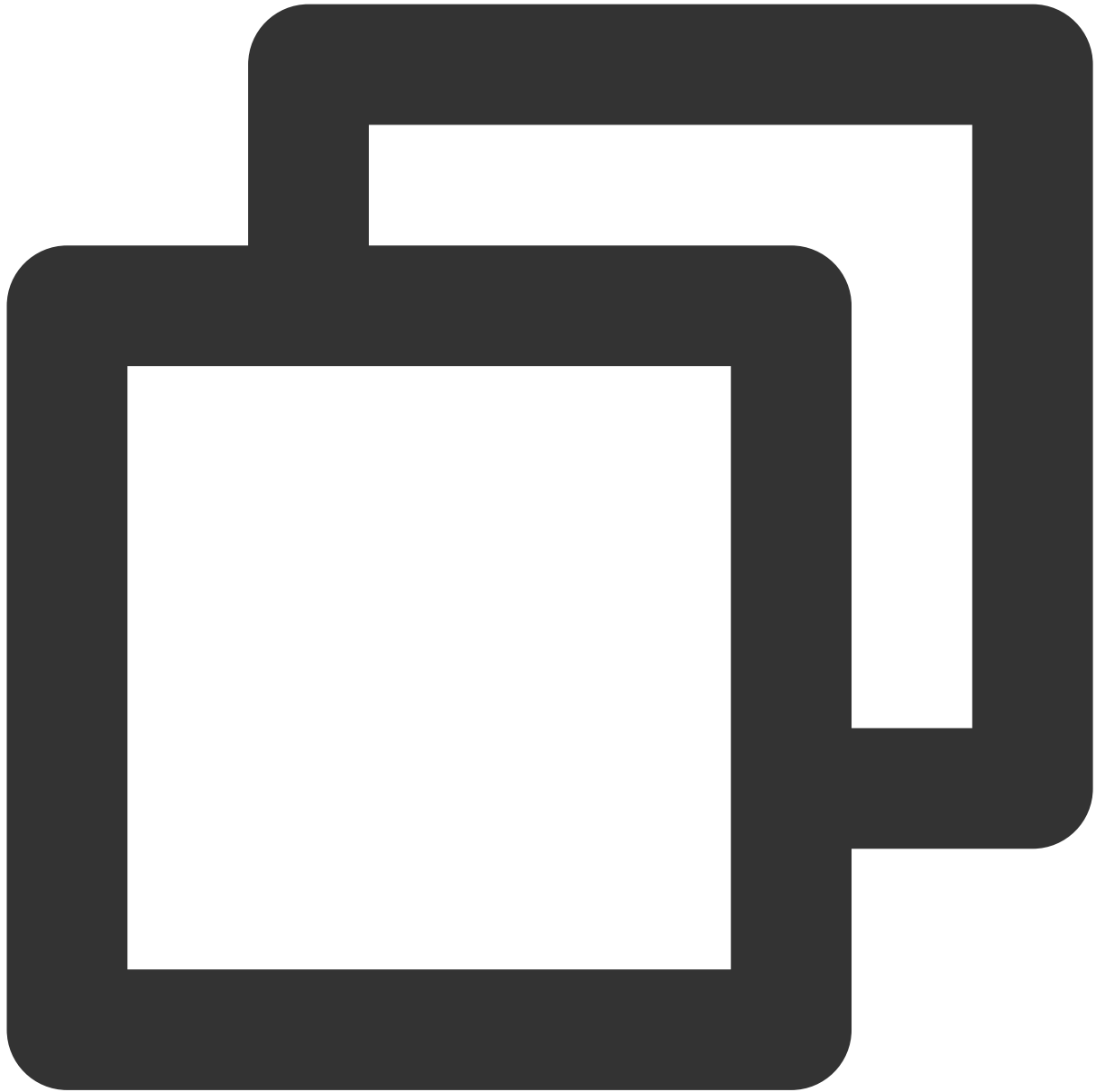
Step 2. Add the CORS response header in CDN, This is basically a default setting in CDN, so you don't actually need to do anything.

```
Access-Control-Allow-Origin: *
```

Solution 2: `try catch`

Only `Script error.` can be captured in `window.onerror`, but a detailed error stack can be printed in `try catch`.
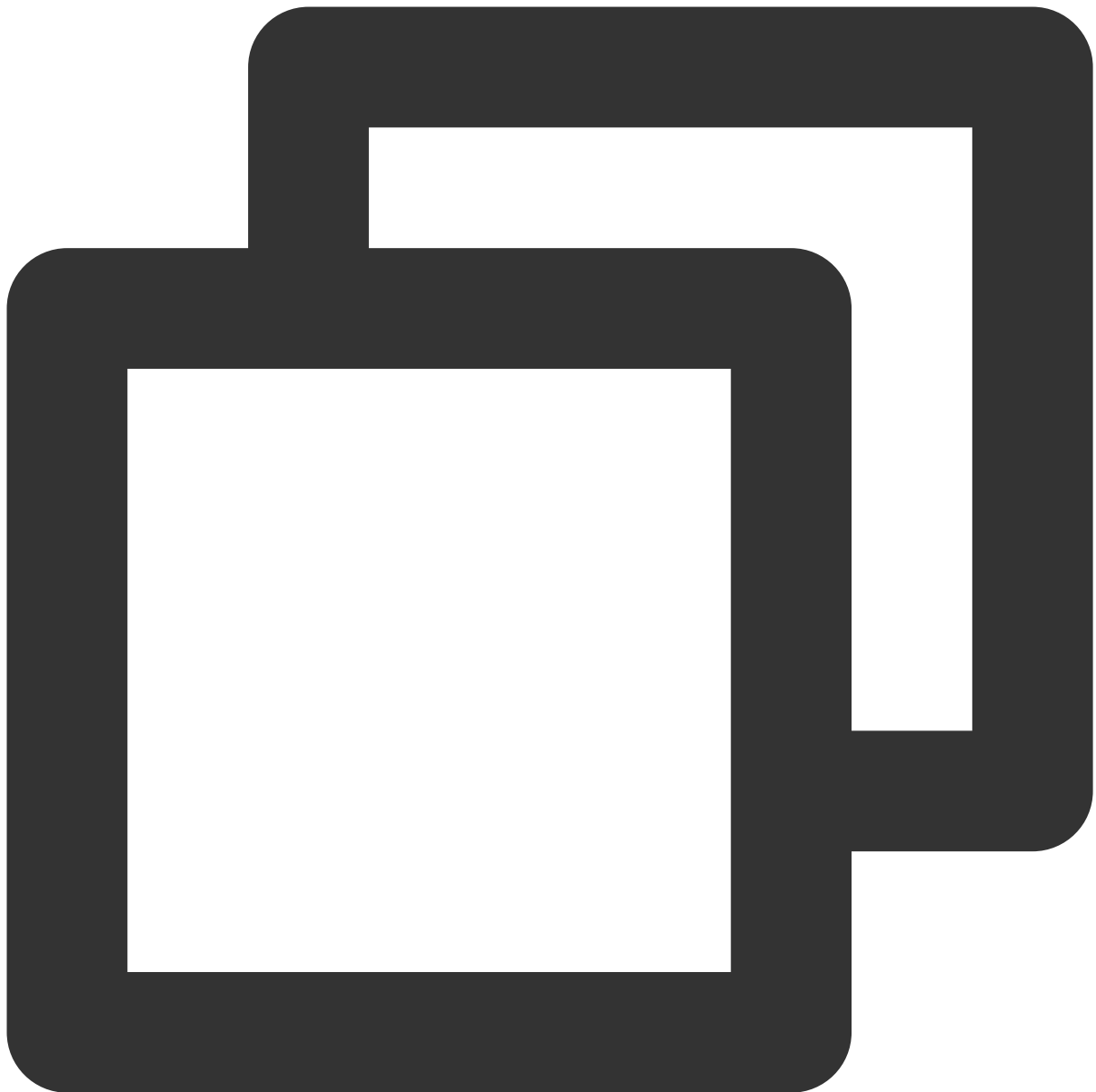
```
<!doctype html>
<html>
<body>
    <script src="http://another-domain.com/app.js"></script>
    <script>
        window.onerror = function (message, url, line, column, error) {
            console.log(message, url, line, column, error);
        }
        try {
            foo(); // Call the `foo` method defined in `app.js`
        } catch (e) {
```

```
            console.log(e);
            throw e; // The captured error that is actively reported is not `Script
        }
    </script>
</body>
</html>
```

If you just want to directly block this problem without solving it, you can choose to not report special logs as detailed in the next question.

I don't want to report certain special logs to RUM. What should I do?

The SDK provides a hook to help you manipulate and restrict a log before reporting it. You can use `beforeReport` to return `false` or `null` , so that the log will not be reported.

beforeReport
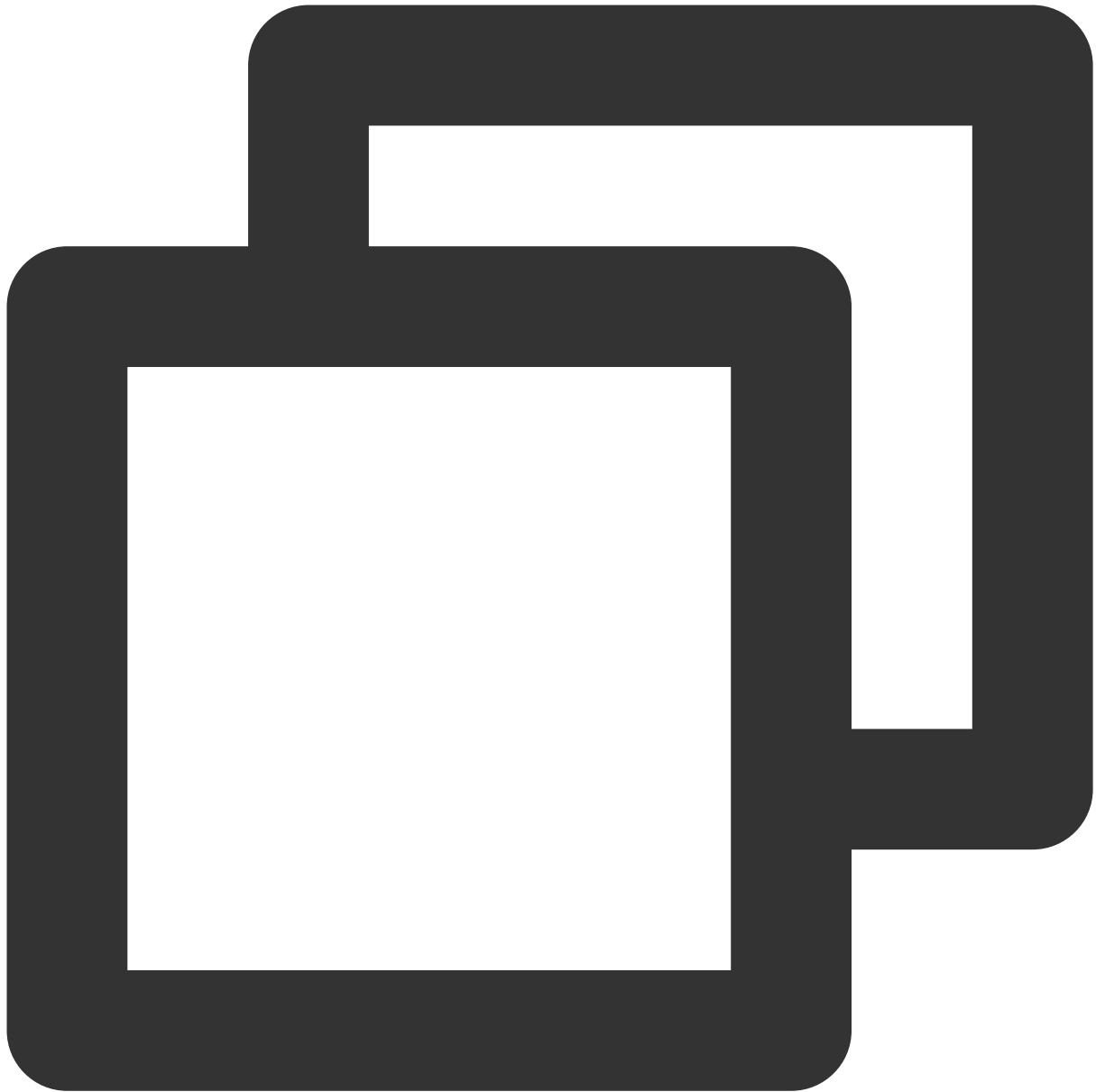
```
new Aegis({  id: 'pGUVFTCZyewhxxxxxx',
  beforeReport(log) {
    if (log.msg && log.msg.indexOf('Script error') !== -1) {
      return false
    }
    return log;
  }
})
```

How does the SDK get the network type? Why is my network type incorrect?

If `UA` has `NetType` , `UA` is obtained from `NetType` . If it has no such value and you use `navigator.connection.effectiveType || navigator.connection.type` , `effectiveType` will be a type calculated based on your network connection speed, which isn't the actual type.

Why is the API request duration reported by the SDK different from that in `network` ?

The SDK hijacks `fetch` and `xhr` and sets breakpoints before and after the API request to test the request speed. As the execution of the JavaScript time difference calculation logic depends on the main JavaScript thread, if a thread is blocked at the `end` time, the actually calculated time will be longer than the time in `network` in the browser.

Why is the reported network type different from the actual type? For example, why is 4G reported when the current user actually uses Wi-Fi?

RUM gets the network type through `NetType` and `navigator.connection.effectiveType` in `UA` . The former depends on the network information injected to `UA` by the browser, while the latter is an "equivalent network type" provided by the browser, which doesn't represent the actual network type. As the `navigator.connection.effectiveType` API still has compatibility issues and currently doesn't support iOS, the network type will be identified as unknown.

I have enabled the SPA parameter, but why isn't the page performance data reported during page redirect?

Redirect between SPA pages is essentially the execution of a snippet of JavaScript code. The firstScreenTime time is the time between the initiation of the request by the user browser and the completion of visible page element rendering; therefore, the firstScreenTime cannot be calculated.

Many Ajax exceptions with status code 0 are reported in logs. What could be the cause?

Status code 0 of the HTTP API may be caused by the following issues: timeout, `abort` , `cancel` , and CORS.

An error is not found during local debugging but frequently occurs on the user side. What could be the cause?

This is because the user may leave the page at any time or an HTTP request may be interrupted due to network problems. In this case, the status code will be 0.

Although there are many reasons why the status code is 0, the logs can always provide some useful information; for example, the error durations in the above screenshot are very short, and the errors occur frequently for a particular user. Based on this, it can be judged that the API is blocked by a browser extension. This problem usually happens for some data reporting APIs.

If the duration is long, there is a high possibility of timeout. You can also view the timeout period of your API for comparison. As for CORS, you can check whether the backend business logic is normal. The problem of `cancel` is very likely to be caused by the frontend business, and you can also check the specific business logic.

Usage

Last updated：2024-01-22 19:25:42

What is the UIN passed in during SDK initialization?

Unique Identity Number (UIN) is a unique user ID provided by the Aegis SDK. RUM doesn't verify the UIN and only uses it as an index for log query. During development, you can enter the user ID provided by your business or information such as `open-id` and `uuid` as long as it facilitates query.

Why does custom speed test support data only between 0 and 60000?

In the logic of custom speed test, RUM calculates the mean and median values of any data reported by users and displays them. Since the volume of dirty data generated on the server is unknown, even a small amount of it can have a huge impact on the actual user data. Therefore, RUM restricts the user data on the server to values between 0 and 60000.

Which field does RUM use to calculate the UV?

The Aegis SDK generates an `aid` for each user (device) as a unique ID, which is stored in `localStorage` in the browser for calculating the UV. `aid` is not affected by the login state and will be updated only when the user clears the browser cache.

Why doesn't RUM use the user UIN to calculate the UV?

The UIN cannot be used to calculate the UV if the current user is not logged in. It is unreliable as a value specified during development since it cannot unify the state before and after user login.

What is an API request log?

An API request log records the request and response content of a user API. Generally, such logs are reported only by used in the allowlist; however, if a JavaScript error occurs when a user is using an API, the API information of the user in the previous period of time will also be reported to the server to help you locate and analyze the problem.

How is `FMP` calculated?

RUM listens on the first screen DOM changes within 3 seconds after a page is opened and takes the time when the number of DOM changes reaches the highest as the time when the first screen framework rendering is completed. (`setTimeout` is used to start first screen element collection 3 seconds after SDK initialization. As JavaScript is executed in a single-thread environment, the collection time point may be more than 3 seconds after SDK initialization.)

How is the DOM parsing time in the performance data calculated?

It is the time between `domInteractive` and `domLoading` in the `PerformanceTiming` API.

Is the user time or service time used when RUM collects data? How long is the delay?

The service time is used. The displayed time is around 1–2 seconds after the reported time, and log search may have a delay of 1–2 minutes.