

# Application Performance Management 2.0

## Best Practices

**Issue** 01  
**Date** 2024-06-14



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2024. All rights reserved.**

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Cloud Computing Technologies Co., Ltd.

## **Trademarks and Permissions**



HUAWEI and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

## **Notice**

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

---

# Contents

---

|  |           |
|--|-----------|
| <b>1 Locating the Causes of Request Errors.....</b>  | <b>1</b>  |
| <b>2 Searching for Span Information.....</b>         | <b>3</b>  |
| <b>3 Connecting On-premises Services to APM.....</b> | <b>9</b>  |
| <b>4 Associating Traces with Logs.....</b>           | <b>12</b> |

# 1 Locating the Causes of Request Errors

## Background

When the number of external requests increases sharply or the load changes abruptly, application performance problems occur frequently, for example, requests cannot be quickly responded or properly handled. Quickly identifying, locating, and handling these problems are required in routine inspection.

APM has powerful analysis tools for cloud application diagnosis. It displays application statuses, call processes, and user operations through topologies and tracing, so that you can quickly locate and resolve faults and performance bottlenecks.

For example, you can view the call relationships between services and quickly locate abnormal instances through topologies. You can also drill down to services and determine root causes based on method tracing.

## Applicable Scenarios

- Routine inspection, covering application metrics such as latency, throughput, and number of errors
- Quick locating of error calls

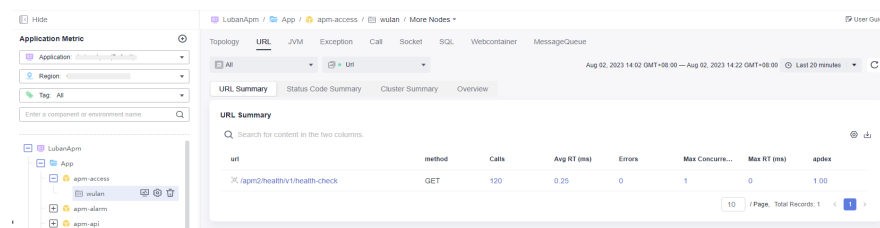
## Procedure

**Step 1** Log in to the APM console.

**Step 2** In the navigation pane, choose **Application Monitoring > Metrics**.

**Step 3** Click the **URL** tab. On the page that is displayed, view metrics such as the number of calls, number of errors, and latency.

**Figure 1-1** Viewing URLs



**Step 4** Click the abnormal URL to go to the tracing page.

**Figure 1-2** URL details

| url                   | method | Number of L... | Average ... | umber of... | Maximu... | Slowest L... | 0ms-10ms | 10ms-10... | 100ms-5... | 500ms-1s | 1s-10s | 10s-n |
|-----------------------|--------|----------------|-------------|-------------|-----------|--------------|----------|------------|------------|----------|--------|-------|
| /user/login           | POST   | 14             | 128271.36   | 14          | 4         | 128332       | 0        | 0          | 0          | 0        | 0      | 14    |
| <b>/user/validate</b> | POST   | 14             | 127265.21   | 14          | 4         | 127354       | 0        | 0          | 0          | 0        | 0      | 14    |

**Step 5** Locate error or slow traces.

**Figure 1-3** Viewing traces

The screenshot shows the 'Tracing' interface with search criteria on the left and a list of 18 records on the right. The records are filtered by 'POST /user/validate'. The response times range from 127288 ms to 127310 ms, and all records show a 500 error code.

**Step 6** Click the corresponding URL to obtain the trace details and determine the root cause.

**Figure 1-4** Trace details

The screenshot shows the 'Trace details' interface. At the top, there is a call graph showing the sequence of calls between 'user', 'email-product-service', 'email-user-service', and 'email-dao-service'. Below the graph is a table of actions with columns for 'Action', 'Response Time', 'Application', 'API Type', 'Call Par.', and 'More In...'. The actions include 'invoke', 'connect', and 'invoke' for various services.

----End

# 2 Searching for Span Information

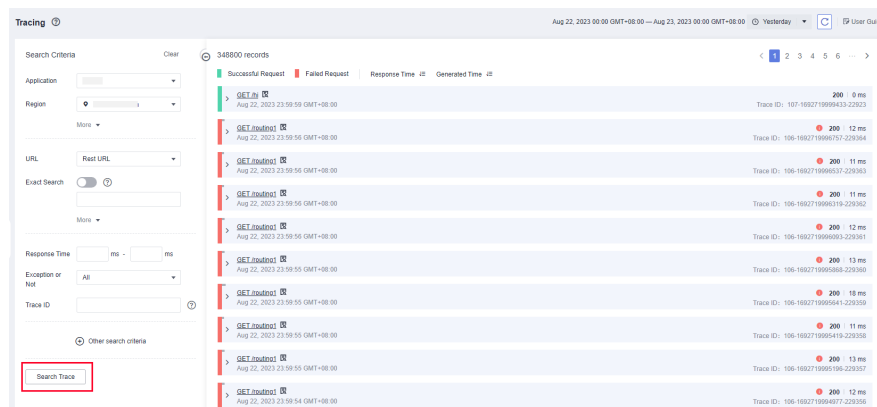
## Background

In the distributed architecture, the calls between microservices are complex. If it takes much time to respond to external requests or some requests become abnormal, you can specify a trace ID or set other criteria on the **Tracing** page to view trace details.

## Procedure

- Step 1** Log in to the APM console.
- Step 2** In the navigation pane, choose **Application Monitoring** > **Tracing**.
- Step 3** Enter the following search criteria and click **Search Trace**.

**Figure 2-1** Tracing search result



**Table 2-1** Search criteria of traces

| Search Criterion | Description                             | Mandatory |
|------------------|---|-----------|
| Application      | Application to which the trace belongs. | Yes       |
| Region           | Region where the trace is located.      | Yes       |

| Search Criterion | Description  | Mandatory |
|------------------|--|-----------|
| Component        | Component to which the trace belongs.  | No        |
| Environment      | Environment to which the trace belongs.  | No        |
| Instance         | Instance to which the trace belongs.   | No        |
| URL              | Trace URL, which can be a REST or real URL. A REST URL contains a variable name, for example, <code>/apm/get/{id}</code> . A real URL indicates an actual URL. | No        |
| Exact Search     | Whether to perform exact match on URLs. If this option is selected, exact match is performed. If this option is not selected, fuzzy match is performed.        | No        |
| Call Method      | HTTP method of the trace.  | No        |
| Status Code      | HTTP status code returned by the trace.  | No        |
| Response Time    | Response time range of the trace. You can specify the minimum and maximum response time to search for traces or leave them empty.                              | No        |
| Exception or Not | Whether to filter the traces that are regarded as exceptions.  | No        |
| Trace ID         | ID of a trace. If you specify this parameter, other search criteria become invalid and the search will be performed based on the trace ID you specify.         | No        |

**Step 4** Click **Other search criteria**. **Custom Parameter**, **Global Trace ID**, and **Application Code** are displayed.

**Figure 2-2** Other search criteria

Search Criteria Clear

Application

Region

More ▾

---

URL

Exact Search  ?

More ▾

---

Response Time  ms -  ms

Exception or Not

Trace ID  ?

---

Custom Parameter

Global Trace ID

Application Code

**Table 2-2** Search criteria of traces

| Search Criterion | Description   | Mandatory |
|------------------|---|-----------|
| Custom Parameter | If you have configured <b>Key for Header Value Interception</b> , <b>Key for Parameter Value Interception</b> , and <b>Key for Cookie Value Interception</b> for URL monitoring, you can set <b>key=value</b> to search.                      | No        |
| Global Trace ID  | Global ID of a trace. If you specify this parameter, other search criteria become invalid and the search will be performed based on the trace ID you specify.   | No        |
| Application Code | If you have configured <b>Service Code Length</b> , <b>Key for Service Code Interception</b> , and <b>Normal Service Code</b> , corresponding application codes will be collected. You can search information based on the application codes. | No        |

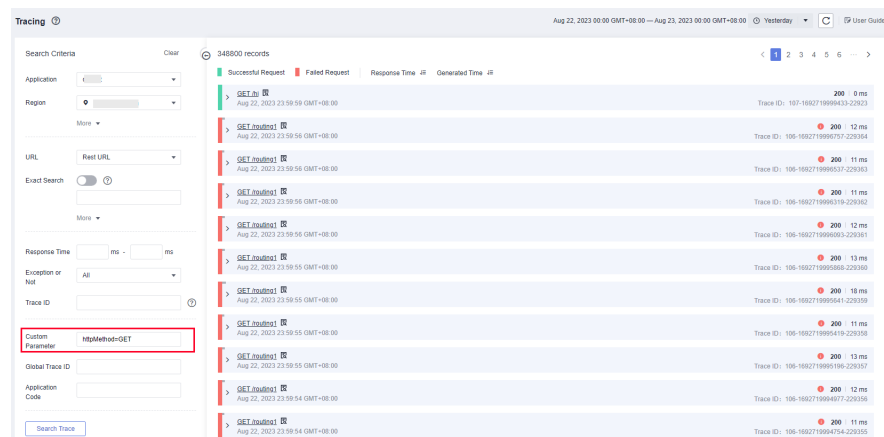


- **Custom Parameter**

Usage Instructions


- a. Configure **Key for Header Value Interception**, **Key for Parameter Value Interception**, and **Key for Cookie Value Interception** for URL monitoring. For details, see [Configuring the URL Monitoring Item](#).
- b. In the **Custom Parameter** text box, set the parameters and values.
- c. Click **Search Trace**. The results are displayed on the right.

**Figure 2-3** Results of querying traces based on the custom parameters

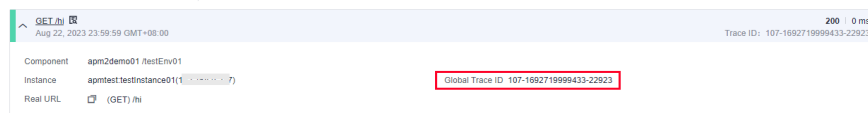


- **Global Trace ID**

Usage Instructions

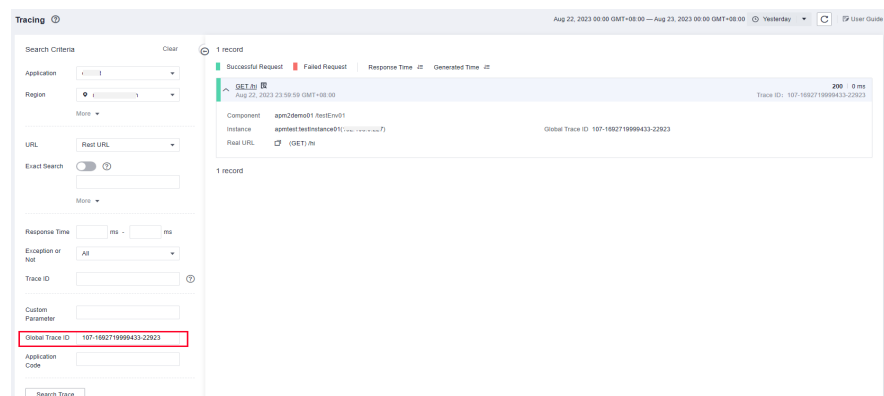
- a. Click  next to the target trace to view the global trace ID.

**Figure 2-4** Obtaining the global trace ID



- b. In the **Global Trace ID** text box, enter the global trace ID.
- c. Click **Search Trace**. The results are displayed on the right.

**Figure 2-5** Results of querying traces based on the global trace ID

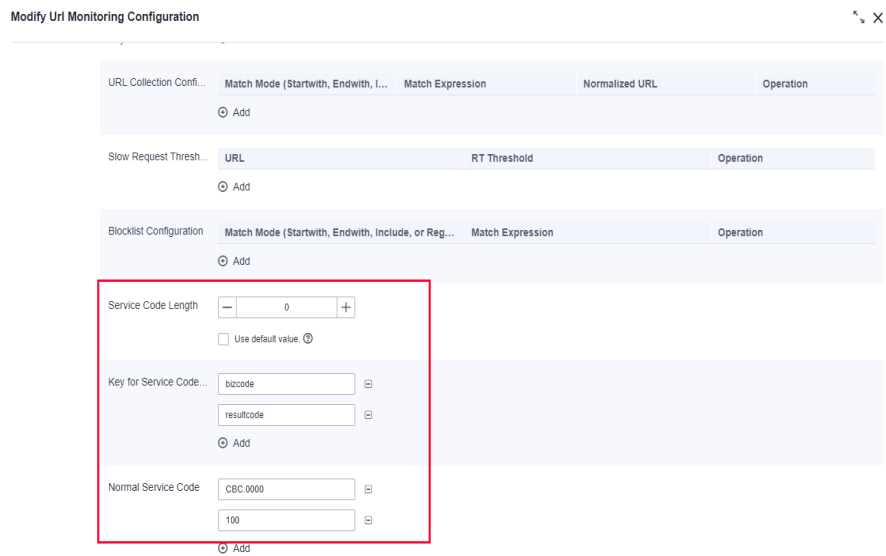


- **Application Code**

Usage Instructions

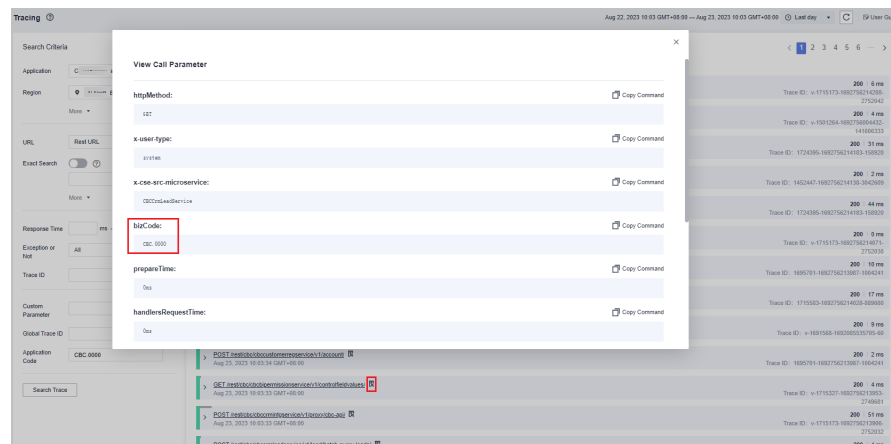
- Configure **Service Code Length**, **Key for Service Code Interception**, and **Normal Service Code** for URL monitoring. For details, see [Configuring the URL Monitoring Item](#).

**Figure 2-6** URL monitoring



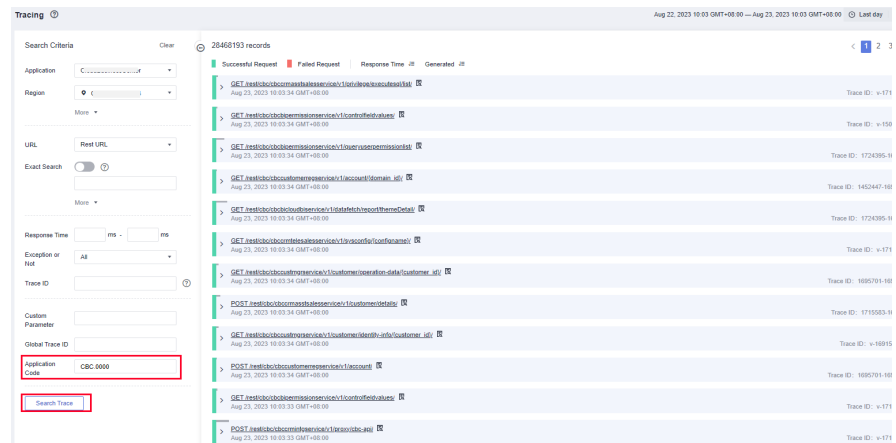
- In the navigation pane, choose **Application Monitoring > Tracing**.
- Click to view the value of the service code, which corresponds to the application code.

**Figure 2-7** Viewing the service code



- In the **Application Code** text box, enter the application code.

**Figure 2-8** Searching for the traces corresponding to the code



e. Click **Search Trace**. The results are displayed on the right.

----End

# 3 Connecting On-premises Services to APM

---

## Background

You cannot connect on-premises services to APM using Direct Connect. To access APM, configure a proxy.

## Configuration Method

If the network between your host and APM is disconnected, configure a proxy.

### Step 1 Configure a proxy.

1. Log in to the AOM 2.0 console.
2. On the menu bar, choose **Collection Management**.
3. In the navigation tree on the left, choose **UniAgent > Proxy Areas**. The **Proxy Areas** page is displayed.
4. Click **Add Proxy** and set related parameters.

**Figure 3-1** Adding a proxy

**Table 3-1** Parameters for adding a proxy

| Parameter        | Description  | Example Value |
|------------------|--|---------------|
| Proxy Area       | Select the created <b>proxy area</b> .                   | region        |
| Host             | Select a host where the UniAgent has been installed.     | -             |
| Proxy IP Address | Set the IP address of the proxy.                         | -             |
| Port             | Enter a port number, which cannot be greater than 65535. | -             |

5. Click **OK**.

**Step 2** Configure the JavaAgent.

1. Download the JavaAgent package to any directory of the host to be connected to APM.

Example command:

```
curl -O https://xxx/apm-javaagent-x.x.x.tar
Download Agent 2.4.1: curl -k https://apm2-javaagent-cn-north-4.obs.cn-north-4.myhuaweicloud.com/apm_agent_install2.sh -o apm_agent_install.sh && bash apm_agent_install.sh -ak {APM_AK} -sk {APM_SK} -masteraddress https://xx.xx.xx.xx:41333 -obsaddress https://apm2-javaagent-cn-north-4.obs.cn-north-4.myhuaweicloud.com -version 2.4.1; history -cw; history -r
```

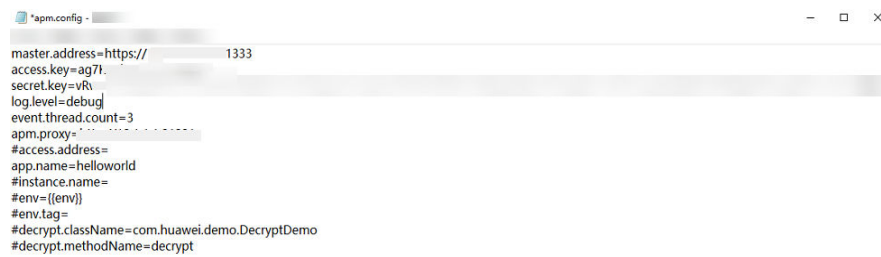
2. Run the **tar** command to decompress the JavaAgent package.

Example command:

```
tar -xvf apm-javaagent-x.x.x.tar
```

3. Modify the **apm.config** file in the JavaAgent package. Add **apm.proxy** to the configuration file, as shown in the following figure.

**Figure 3-2** Configuration file



**NOTE**

- Agents of 2.4.1 and later support access through a proxy. Format: **apm.proxy=ip:port** (Obtain *ip:port* from the AOM console.)
- To obtain an AK/SK, see [Access Keys](#).
- To obtain the **master.address**, see [Access Address \(master.address\)](#).

**Step 3** Restart the application.

1. Modify the startup script of the Java process.

Add the path of the **apm-javaagent.jar** package and the component name of the Java process to the end of the Java command in the service startup script.

Example of adding **-javaagent** parameters:

```
java -javaagent:/xxx/apm-javaagent/apm-javaagent.jar=appName={appName}
```

2. Restart the application.

----End

# 4 Associating Traces with Logs

## Application Scope

Common log frameworks, such as Logback and Log4j.

## Example

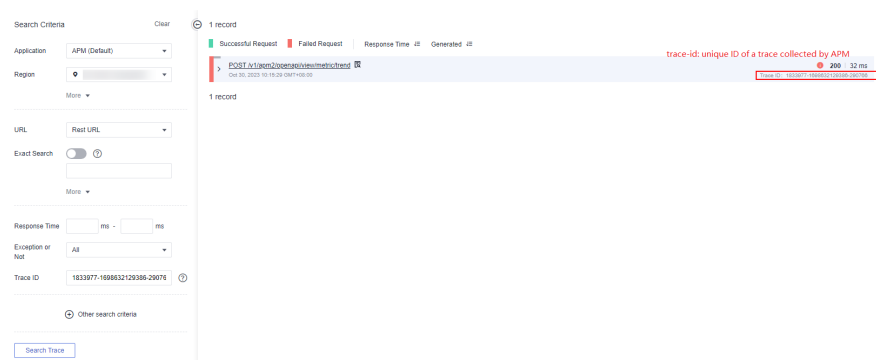
```
<property name="LOG_PATTERN" value="%d{yyyy-MM-dd HH:mm:ss.SSS}} |  
gtraceid: %X{apm-gtraceid} | traceid: %X{apm-traceid} | spanId: %X{apm-  
spanid}">
```

```
</property>
```

## Trace Parameters

1. **apm-traceid**: unique ID of a trace collected by APM.

Figure 4-1 Unique ID of a trace



2. **apm-gtraceid**: unique ID of a trace which is not sampled.

### NOTE

APM has a certain sampling ratio. The **apm-gtrace-id** parameter is used to uniquely identify a trace that is not sampled.

3. **apm-spanid**: ID of a microservice called in a trace. Example:

Figure 4-2 Calls between microservices

