

CodeArts PerfTest

User Guide

Issue 01
Date 2024-04-08



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

1 CodeArts PerfTest Use Process.....	1
2 Permissions Management.....	3
2.1 Creating a User and Assigning CodeArts PerfTest Permissions.....	3
3 Test Resource Management.....	5
3.1 Creating a Private Resource Group.....	5
3.2 Managing a Private Resource Group.....	7
4 PerfTest Project Management.....	12
4.1 Creating a Test Project.....	12
4.2 Setting Intelligent Analysis.....	13
4.3 Configuring Resource Tags.....	14
4.4 Managing Test Projects.....	15
5 PerfTest Case Management.....	19
5.1 Test Case Description.....	19
5.1.1 Introduction to Test Cases.....	19
5.1.2 Pressure Test Engines for Tens of Millions of Concurrent Users.....	20
5.1.3 Eight Pressure Test Modes.....	21
5.2 Directory Management of Test Cases.....	22
5.3 Creating a Test Case.....	24
5.4 Configuring a Test Case (High-Performance Test Projects).....	25
5.4.1 Filling in Case Details.....	25
5.4.2 Configuring a Case Script.....	26
5.4.3 Adding Request Information (Packet).....	30
5.4.4 Adding Request Information (Think Time).....	41
5.4.5 Adding Request Information (Response Extraction).....	41
5.4.6 Adding Request Information (Checkpoint).....	48
5.4.7 Adding a Data Instruction, Cycle Controller, Condition Judgment, or Rendezvous Point.....	50
5.4.8 Adding Concurrency Weight Controllers and Throughput Controllers.....	54
5.4.9 Pressure Configuration.....	56
5.4.10 Advanced Configuration.....	62
5.4.11 SLA Configuration.....	62
5.5 Configuring a Test Case (Scalable Test Projects).....	62
5.5.1 Case Details.....	63

5.5.2 Configuring a Case Script.....	64
5.5.3 Add Step.....	64
5.5.4 Pressure Configuration.....	67
5.5.5 Advanced Configuration.....	73
5.5.6 SLA Configuration.....	74
5.5.7 Action Word Library Management.....	74
5.6 Setting Global Variables (High-Performance Test Projects).....	75
5.6.1 Overview of Global Variables (High-Performance Test Projects).....	75
5.6.2 Adding an Integer, Text, or Enumerated Global Variable.....	76
5.6.3 Adding a .csv or .xlsx Global Variable File.....	77
5.6.4 Variable Read Rules.....	78
5.6.5 Inserting a Variable.....	79
5.7 Setting Global Variables (Extensible Test Projects).....	81
5.7.1 Overview of Global Variables (Extensible Test Projects).....	81
5.7.2 Creating and Deleting Global Variables.....	81
5.7.3 Global Variable Management.....	84
5.8 Binding a Domain Name.....	87
5.9 Plug-in Management.....	87
5.9.1 Summary.....	87
5.9.2 Signature Conventions.....	88
5.9.3 Guidelines for Compiling Go Plug-ins.....	92
5.9.4 JSON Description Statement.....	94
5.9.5 Custom Examples.....	95
5.9.6 Plug-in Usage.....	101
5.10 Resetting Configurations.....	104
5.11 Deleting a Test Case.....	104
5.12 Debugging a Case.....	104
5.13 Batch Operations.....	105
6 PerfTest Task Management.....	107
6.1 Creating a Test Task.....	107
6.2 Starting a Test Task.....	108
6.3 Managing Test Tasks.....	108
6.4 Associating Analysis Objects.....	110
7 PerfTest Report Management.....	112
7.1 Test Report Description.....	112
7.2 Viewing a Real-Time Test Report.....	116
7.3 Viewing an Offline Test Report.....	118
7.4 Report Comparison.....	120
8 Transaction Management.....	121
8.1 Creating a Transaction.....	121
8.2 Importing Recorded Scripts.....	122

8.3 Importing a Swagger File.....	125
8.4 Debugging Transactions.....	125
8.5 Managing Transactions.....	126
8.6 Managing Transaction Request Information.....	128
9 JMeter Test Project Management.....	130
9.1 Introduction to the JMeter Project.....	130
9.2 Managing JMeter Projects.....	131
9.3 Managing JMeter Test Plans.....	132
9.4 Managing JMeter Test Reports.....	137
9.5 Configuring Intelligent Analysis for JMeter Projects.....	145
9.6 Configuring Resource Tags.....	146
10 Crontask.....	148
10.1 Creating a Crontask.....	148
10.2 Managing a Crontask.....	150
11 Configuring SLAs.....	151
12 Auditing.....	154
12.1 CodeArts PerfTest Operations That Can Be Recorded by CTS.....	154
12.2 Querying Real-Time Traces.....	155
13 Reference.....	159
13.1 Header Description.....	159
13.2 Regular Expression Metacharacters.....	162
13.3 Modifying an Exported Project File.....	165
13.4 Mapping Between JMeter and PerfTest Fields.....	170

1 CodeArts PerfTest Use Process

CodeArts PerfTest provides performance test services for cloud applications that are built based on HTTP, HTTPS, TCP, UDP, HLS, RTMP, WebSocket, MQTT, or HTTP-FLV. CodeArts PerfTest quickly simulates service peaks with a large number of concurrent users. It allows you to define the contents and time sequences of packets and supports complex combinations of multiple transactions. After tests are complete, CodeArts PerfTest provides professional test reports to evaluate your service quality.

You can complete a performance test in four steps listed as in the following table.

Table 1-1 Process

1. Prepare a Resource Group	2. Create a Test Project	3. Create a Test Task	4. View a Test Report
Prepare a test resource group for running the performance test. NOTE Test resource groups are classified into shared resource groups and private resource groups. Shared resource groups are provided by the system by default, and private resource groups need to be created.	CodeArts PerfTest manages your test projects. Transactions, test tasks, and test reports are shared in the same project.	Use predefined test transactions to create performance test scenarios.	CodeArts PerfTest provides real-time and offline performance test reports for result analysis.

Constraint


To use CodeArts PerfTest for a pressure test on a public website, the user must be on the whitelist of this website. Otherwise, the user must bear any and all consequences and legal liabilities.


Prerequisites

1. You have [registered a Huawei account and enabled Huawei Cloud services](#).
2. Your account has permission to use CodeArts PerfTest. For details, see [Creating a User and Assigning CodeArts PerfTest Permissions](#).

Logging In to the CodeArts PerfTest Console

Step 1 Log in to the [management console](#).

Step 2 Click  and select a region.

Step 3 Click  in the upper left corner and click **CodeArts PerfTest** in the service list.

- If you log in for the first time, click **Authorize** on the displayed service authorization page to authorize CodeArts PerfTest to use the services on which it depends. Then, the CodeArts PerfTest console is displayed.
- If this is not your first login, the CodeArts PerfTest console is displayed directly.

----End

2 Permissions Management

2.1 Creating a User and Assigning CodeArts PerfTest Permissions

This section describes how to use [IAM](#) to implement fine-grained permissions control for your CodeArts PerfTest resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing CodeArts PerfTest resources.
- Grant only the permissions required for users to perform a specific task.
- Entrust other Huawei Cloud accounts or cloud services to perform efficient O&M on your CodeArts PerfTest resources.

If your Huawei Cloud account does not require individual IAM users, skip this chapter.

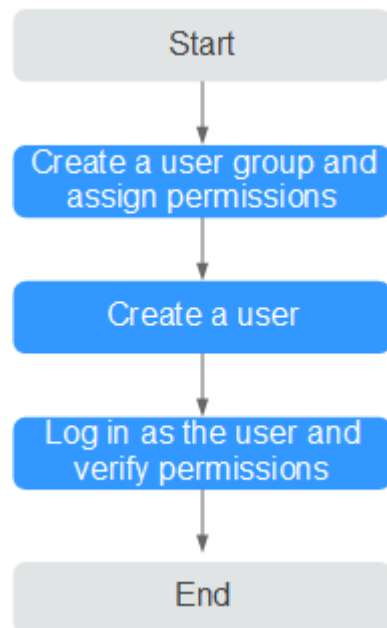
This section describes the procedure for granting permissions (see [Figure 2-1](#)).

Prerequisites

Before assigning permissions to user groups, you should learn about the system permissions listed in [Permissions Management](#). For system-defined policies of other services, see [System-defined Permissions](#).

Example

Figure 2-1 Process of granting CodeArts PerfTest permissions



1. **Create a user group and assign permissions** to it.
Create a user group on the IAM console and assign the **CodeArts PerfTest Administrator** permission to the group.
2. **Create an IAM user.**
Create a user on the IAM console and add the user to the group created in **1**.
3. **Log in** and verify permissions.
Log in to the CodeArts PerfTest console as the created user, switch to the authorized region, and verify the permissions. Choose **Service List > CodeArts PerfTest**. On the CodeArts PerfTest console, choose **PerfTest Projects** in the navigation pane, and click **Create Test Project** on the right. If a test project can be created, the permissions are successfully configured.

3 Test Resource Management

3.1 Creating a Private Resource Group

Test Resource Groups and Their Constraints

- Test resource groups are classified into shared resource groups and private resource groups. Shared resource groups are provided by the system by default, and private resource groups need to be created.
- Execution nodes of the shared resource group have been bound with an elastic IP address (EIP). When the tested application has network access restrictions, use a private resource group.
- A shared resource group supports a maximum of 1,000 concurrent users and 100 Mbit/s bandwidth. If higher concurrency or bandwidth is required, use a private resource group.
- JMeter test tasks can use only private resource groups. A JMeter test task allows a maximum of 2 million concurrent requests.

Procedure

NOTE

- When creating test resources for the first time, you need to be authorized by your account. An agency is automatically created so that CodeArts PerfTest can perform operations on Cloud Container Engine (CCE).
- To use CodeArts PerfTest, you must have the **CodeArts PerfTest Administrator** or **CodeArts PerfTest Developer** permission (you can only view the projects you created).
- To manage private resource groups, you must have the **CodeArts PerfTest Administrator** permission or both **CodeArts PerfTest Developer** and **CodeArts PerfTest Resource Administrator** permissions.
- To use private resource groups, you must have the **CodeArts PerfTest Administrator** permission or both **CodeArts PerfTest Developer** and **CodeArts PerfTest Resource Developer** permissions.
- For details about the permissions required for using CodeArts PerfTest and their application scenarios, see [What Are the Permissions Required for Using CodeArts PerfTest?](#).

- Step 1** Log in to the CodeArts PerfTest console, choose **Resource Groups** in the navigation pane, and click **Create Resource Group**.
- Step 2** (Optional) If this is the first time you create a private resource group, grant CodeArts PerfTest permissions necessary for creating private resource groups.
- Step 3** If you do not have a CCE cluster, perform **Step 4** to create one before creating a resource group. If a CCE cluster is available, go to **Step 5** to create a resource group.
- Step 4** Create a cluster.

Click **Create Cluster**. The page for buying CCE clusters is displayed. For details about how to create a cluster, see [Buying a CCE Cluster](#).

 **NOTE**

- The cluster management scale is related to the number of execution nodes. Create nodes of the corresponding specifications based on the number of concurrent users for pressure testing. For example, if 20 execution nodes are required, you can set the cluster scale to 50 nodes.
- You are advised to select **Tunnel network** as the network model. **Container CIDR Block** and **Service CIDR Block** must be the same as those of the tested object.

Click **Next: Add-on Configuration**. Retain the default settings.

Click **Next: Confirm**. After confirming that the cluster configuration information is correct, select **I have read and understood the preceding instructions** and click **Submit**. It takes about 6 to 10 minutes to create a cluster.

After the cluster is created, return to the cluster management page and click **Create Node**. For details about how to create a node, see [Creating a Node](#).

 **NOTE**

- A node must have at least four vCPUs and 8 GB memory.
- Select EulerOS as the operating system.
- At least two nodes (one debugging node and one execution node) are required. The number of nodes depends on the specifications of the pressure test object. For example, 21 execution nodes (one debugging node and 20 execution nodes) are required for 100,000 concurrent users and 4 vCPUs | 8 GB memory.
- If the CCE cluster node and the application to be tested are not in the same VPC network, bind an EIP to the CCE cluster node. You can use existing EIPs. If no EIP is available, click **Auto create** to create one. When EIPs are automatically created, you are advised to pay by traffic and set the bandwidth to a large value to avoid affecting the pressure test. Additionally, an EIP with specified configurations is automatically assigned to each node. If the number of EIPs is less than the number of nodes, the EIPs are randomly bound to the nodes.

Click **Next: Confirm**. After confirming that the node configuration information is correct, select **I have read and understood the preceding instructions** and click **Submit**. After the node is created, return to the CodeArts PerfTest console.

- Step 5** Create a resource group.

Choose **Resource Groups** in the navigation pane, and click **Create Private Resource Group**.

Set the basic information by referring to [Table 3-1](#).

Table 3-1 Creating a private resource group

Parameter	Description
Resource Group Name	Name of the private resource group to be created.
Debugging Cluster	Select a CCE cluster from the drop-down list.
Advanced Configuration	This parameter is optional. For details, see Advanced Configuration . NOTE <ul style="list-style-type: none">This feature is for whitelisted users. To use this feature, contact background support personnel to enable it and approve your request.Modifying advanced configuration parameters may cause task execution failures. Exercise caution when performing this operation.
Debugging Node	Node used for debugging. The debugging node cannot be changed after the resource group is created.
Execution Node	Target machine that a pressure test will be performed on and can provide performance data during testing.

Click **Create**.

----End

3.2 Managing a Private Resource Group

After a private resource group is created, you can perform the following operations on it.

Custom JMeter Installation Package


To switch the JMeter version, customize the JMeter installation package to meet service requirements.

- Step 1** Log in to the CodeArts PerfTest console and choose **Resource Groups** in the navigation pane.
- Step 2** Download [JMeter installation package](#) of the required version.
- Step 3** In the resource group list, click **Customization JMeter Installation package**, import the installation package, and wait until a message is displayed indicating that the import is successful.

NOTE

The JMeter installation package must be in .tgz, .tar.gz, or .zip format and smaller than 100 MB. The version must be 5.2.1 or later. Ensure that the installation package can be executable locally.

Step 4 (Optional) Delete the JMeter installation package.

If a message is displayed indicating that the upload fails, or you need to upload the JMeter installation package again, click  to delete the current package.

----End

Advanced Configuration

NOTE

- This feature is for whitelisted users. To use this feature, contact background support personnel to enable it and approve your request.
- Modifying advanced configuration parameters may cause task execution failures. Exercise caution when performing this operation.

When the private resource group is idle (the number of running executors is 0), you can adjust the CPU and memory specifications of a single executor.

Step 1 Log in to the CodeArts PerfTest console and choose **Resource Groups** in the navigation pane.

Step 2 In the resource group card, click **Advanced Config**. Set parameters by following the instructions in [Table 3-2](#).

Table 3-2 Advanced configuration

Parameter	Description
Single Executor CPU	The value is an integer ranging from 1 to 128. The default value is 4 (unit: core).
Single Executor Memory	The value is an integer ranging from 1024 to 524,288. The default value is 8192 (unit: MB).
Container/ Node IP Address	Select whether the container IP address is consistent with the node IP address.
DNS Policy	Select Cluster DNS or Node DNS.

Step 3 Click **OK**.

Step 4 Click **Reset** to restore the default settings.

----End

Adding a Cluster

When you create a private resource group, a node cluster is selected. If many clusters or clusters in different regions are required to run a pressure test task, add clusters and configure the pressure percent to allocate the task to different clusters.

Step 1 Log in to the CodeArts PerfTest console and choose **Resource Groups** in the navigation pane.

Step 2 In the resource group card, click **Resource Adjust**.

Step 3 Add a cluster.

Click **Add Cluster**, set the basic information by referring to [Table 3-3](#), and click **OK**.

Table 3-3 Adding a cluster

Parameter	Description
Region	Select the region of the created CCE node cluster from the drop-down list. The region can be different from the current CodeArts PerfTest region.
Node Cluster	Select a CCE cluster from the drop-down list.
Execution Node	Select a pressure target machine that provides performance data during a pressure test. NOTE If the region of the new cluster is different from the current CodeArts PerfTest region, use a cross-region node. Bind an EIP to the node or use NAT to allow the node to access the public network.

You can repeat this step to add more clusters.

Step 4 Configure the pressure percent.

After clusters are added, you can view all node clusters of the current resource group in the **Resource Adjust** dialog box, and adjust the **Pressure Percent** of different clusters.

Step 5 (Optional) Delete a node cluster.

In the **Resource Adjust** dialog box, click **Delete**.

----End

Scaling a Node Cluster

Step 1 Log in to the CodeArts PerfTest console and choose **Resource Groups** in the navigation pane.

Step 2 In the resource group card, click **Resource Adjust**.

Step 3 Scale a node cluster.

- Scale out the resource group: Click **Scale**, select existing nodes and new nodes in the **Select Execution Node** dialog box, and click **OK**.
- Scale in the resource group: Click **Scale** under the resource group, select the execution nodes you want to keep, and click **OK**.

----End

Viewing Information About a Private Resource Group

- Step 1** Log in to the CodeArts PerfTest console and choose **Resource Groups** in the navigation pane.
- Step 2** In a resource group card, you can view the information listed in the following table.

Table 3-4 Private resource group information

Name	Description
Private Resource Group Name	Name entered when you were creating a private resource group.
Private Resource Group Status	Running Abnormal Upgradable Deploying
Actuators	In-use/Total Actuators Concurrency supported by a single executor of high-performance test projects is 5,000 (HTTP/HTTPS), 5,000 (WebSocket), 5,000 (MQTT), 1,000 (JMeter), or 1,000 (HLS/RTMP/HTTP-FLV). Concurrency supported by a single executor of scalable test projects is 500. NOTE Scalable test projects are for whitelisted users. To use this feature, contact background support personnel to enable it and approve your request.
Current Concurrency	Sum of concurrent users of all node clusters in the current private resource group.
Debug Node	Debugging nodes selected when you were creating a private resource group.
Execution Node	Execution nodes of all node clusters in a resource group. Click More to view all execution nodes of the resource group.
Created At	Time when the private resource group was created.
Modified At	Last time when the private resource group was modified.

----End

Upgrading a Private Resource Group

- Step 1** Log in to the CodeArts PerfTest console and choose **Resource Groups** in the navigation pane.

Step 2 In a resource group card, click **Upgrade**.

----End

Deleting a Private Resource Group

Step 1 Log in to the CodeArts PerfTest console and choose **Resource Groups** in the navigation pane.

Step 2 In a resource group card, click **Delete**.

 **NOTE**

When you delete a resource group, nodes will not be used in pressure tests again instead of being deleted. To delete a node, go to the corresponding service to perform the deletion.

----End

4 PerfTest Project Management

4.1 Creating a Test Project

CodeArts PerfTest helps you manage test projects. Transaction modes, test cases, test tasks, real-time reports, intelligent analysis results, and offline reports are shared in a test project. You can create different test projects for different test programs.

You can customize or use a template to create a test project. CodeArts PerfTest defines templates for the following scenarios:

- **All-in-one network office system scene:** You can quickly construct a pressure model by simulating system access requests to detect service performance bottlenecks under different pressure models and prevent service breakdown.
- **Stream scene:** Supports streaming push and pull with common streaming protocols. Construct corresponding pressure test scenarios and simulate typical audio and video scenarios to perform performance pressure tests to identify risks in advance.
- **Seckill scene:** Simulate a large number of users performing operations on offerings at a specified time point to check the reliability of e-mall services when the access pressure increases exponentially.
- **E-commerce scene:** Simulate a large number of merchants logging in to the e-commerce system and perform operations such as viewing and maintaining offering information. Verify that related operations are not interrupted and services can run properly.
- **Quick Create:** Quickly create customized pressure test scenarios and orchestrate test cases on the test case page, greatly improving test case design efficiency.

Manually Creating a Test Project

Step 1 Log in to the CodeArts PerfTest console, choose **PerfTest Projects** in the left navigation pane, and click **Create Test Project**.

Step 2 Set the basic information by referring to [Table 4-1](#).

Table 4-1 Creating a test project

Parameter	Description
Project Name	Name of a new test project. NOTE The project name can contain a maximum of 128 bytes, including letters, digits, hyphens (-), underscores (_), slashes (/), and periods (.).
Type	<ul style="list-style-type: none">• High performance: efficient utilization of computing resources with massive concurrency and cost effectiveness• Scalable: Custom scripts are supported for customization requirements. NOTE <ul style="list-style-type: none">- After a test project is created, the project type cannot be modified.- Scalable test projects are for whitelisted users. To use this feature, contact background support personnel to enable it and approve your request.
Description	Description of a new test project.

Step 3 When the configuration is complete, click **OK**.

After creating a test project, you can add a test case for the test project. For details, see [PerfTest Case Management](#).

----End

Creating a Test Project Using a Template

Step 1 Log in to the CodeArts PerfTest console and choose **Dashboard** in the left navigation pane.

Step 2 Select a template to automatically create test tasks. CodeArts PerfTest provides the following templates:

- **All-in-one Scene**
- **Stream Scene**
- **Seckill Scene**
- **E-commerce Scene**
- **Quick Create**

You can modify a test case as required. For details, see [PerfTest Case Management](#).

----End

4.2 Setting Intelligent Analysis

You can add analysis objects for test projects you created.


 **NOTE**

Currently, CodeArts PerfTest in the Singapore Region has been connected to APM 2.0, and the APM 1.0 configuration in CodeArts PerfTest is invalid. Use APM 2.0 when setting intelligent analysis.

Prerequisites

- Application Performance Management (APM) has been enabled, a probe has been deployed, and an application has been generated.
- Application Operations Management (AOM) has been enabled, and a probe has been deployed.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the PerfTest project to which analysis objects are to be added to go to the details page.
- Step 3** On the **Intelligent Analysis** tab page, click **Create Analysis Object**.
- Step 4** In the displayed dialog box, the system automatically generates the analysis object name, but you can change it. Select the analysis object to be added. The analysis object can be either an application group, that is **Application Monitor (APM)**, or a node, that is **Node Monitor (AOM)**. The application group is the monitoring group that is set when APM probes are installed on the application server. The node is the VM where AOM probes are installed.
- Step 5** Click **OK** after the configuration.
- Step 6** Edit or delete analysis objects.
 - Edit an analysis object: On the **Intelligent Analysis** tab page, click  next to the analysis object to modify and edit it as prompted.

 **NOTE**

If an analysis object is associated with a test task, once its parameters are edited, parameters of its associated object in the test task are also changed.

- Delete an analysis object: On the **Intelligent Analysis** tab page, click  next to the analysis object to delete and delete it as prompted.

 **NOTE**

Analysis objects cannot be deleted if they are being used in a test task.

----End

4.3 Configuring Resource Tags

Add tags to the created PerfTest project. Each PerfTest project can be regarded as a resource. You can tag resources under your account for classification. In scenarios such as cost management, you can quickly view details about a type of resources by tag.

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.



Step 3 On the **Resource Tags** tab, click **Add Tag**. Select a tag key, and a new key-value pair text box is displayed. You can add multiple tags at a time.

NOTE

- A test project can have a maximum of 20 tags. A tag key must be unique and can have only one value.
- Tag keys of different test projects can be invoked by each other. For example, add tag keys **a** and **b** to Test Project 1. When you add tags to Test Project 2, **a** and **b** will be automatically displayed in the tag key drop-down list.

Step 4 Click **OK** after the configuration.

Step 5 You can edit and delete tags.

- Editing a tag: On the **Resource Tags** tab, click  next to the desired tag, enter another value, and click **OK**.
- Deleting a tag: On the **Resource Tags** tab, click  next to the desired tag, and click **OK**.


----End

4.4 Managing Test Projects

After a test project is created, you can modify, delete, import or export it.

Modifying a Test Project

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 In the PerfTest project list, choose , enter the project name or description, and click **OK**.

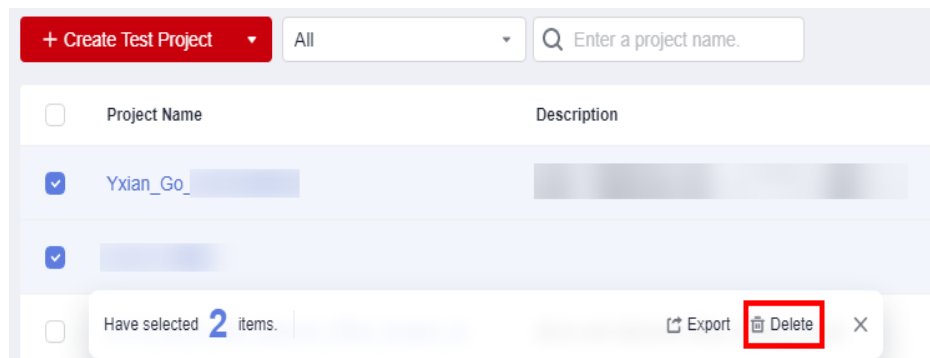
----End

Deleting a Test Project

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.


Step 2 In the PerfTest project list, choose  > **Delete Project**. Delete the test plans as prompted.

Step 3 You can also select one or more test projects, click **Delete**, and delete them as prompted.



----End

Modifying a Tag

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** In the PerfTest project list, choose  > **Modify Tag**. The **Resource Tags** page is displayed. For details, see [Configuring Resource Tags](#).

----End

Exporting a Test Project

You do not need to write a test project from scratch. Instead, find a project similar to the service model, export the test project, slightly modify it, and import the test project.

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Select one or more projects to be exported from the PerfTest project list, and click **Export**. Then save the exported test project file **ProjectList(serial number).json** to the local host.

NOTE

- If SLA rules have been configured for a test project, the SLA rules will not be exported when you export the project.
- If a file is uploaded for a global variable, the corresponding file will not be exported when you export the test project.
- Scalable test projects cannot be exported.
- Projects that use the MQTT protocol cannot be exported.

- Step 3** Modify parameters and the project name (the name must be unique) in the file as required, and save the file.

NOTE

- Do not change the format of the **ProjectList(serial number).json** file.
- The project name can contain a maximum of 128 bytes, including letters, digits, hyphens (-), underscores (_), slashes (/), and periods (.).

----End

Importing a Test Project

Test projects of PerfTest and JMeter can be imported to CodeArts PerfTest.

Step 1 Log in to the CodeArts PerfTest console, choose **PerfTest Projects** in the left navigation pane, and click **Import**.

Step 2 Import a PerfTest project.

1. Set **Project Type** to **PerfTest project**.
2. Click **Select File**, select a project file in JSON format, and click **Import**.
The name of the imported project must be unique in CodeArts PerfTest. Otherwise, the project cannot be imported.

NOTE

Projects that use the MQTT protocol cannot be imported.

Step 3 Import a JMeter project.

NOTE

- Ensure that the fields in the JMeter script are supported in CodeArts PerfTest. For details, see [Mapping Between JMeter and PerfTest Fields](#).
- JMeter parses the following contents: thread group, HTTP request, HTTP header manager, regular expression extractor, user parameter, user-defined variable, default HTTP request value, and fixed timer.
- Random variables are not supported.
- To ensure the accuracy of data import, do not use variables with the same name.
- A JMeter script cannot exceed 5 MB in size.

1. Set **Project Type** to **JMeter project**.
2. Click **Select File**, select a project file in **.jmx** format, and click **Import**.


Step 4 After the project is imported, click **Close**. You can modify the test project by referring to [Modifying a Test Project](#) and [Managing Transaction Request Information](#).

----End

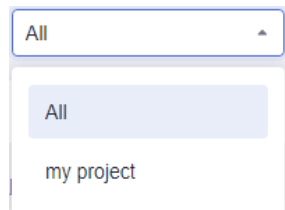
Viewing a Test Project

You can view the existing projects.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Enter a test project name in the search box  in the upper right corner. You can search test projects by name.

You can also select a range to view test projects from the drop-down list



- **All:** Displays all projects within the current user's permission scope, including projects created by subaccounts.
- **my project:** Displays only projects created by the current account.

----End

5 PerfTest Case Management

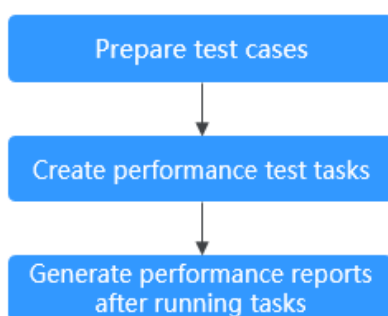
5.1 Test Case Description

5.1.1 Introduction to Test Cases

A PerfTest project consists of test cases, test tasks, and performance reports. Transactions and intelligent analysis are enhanced capabilities.

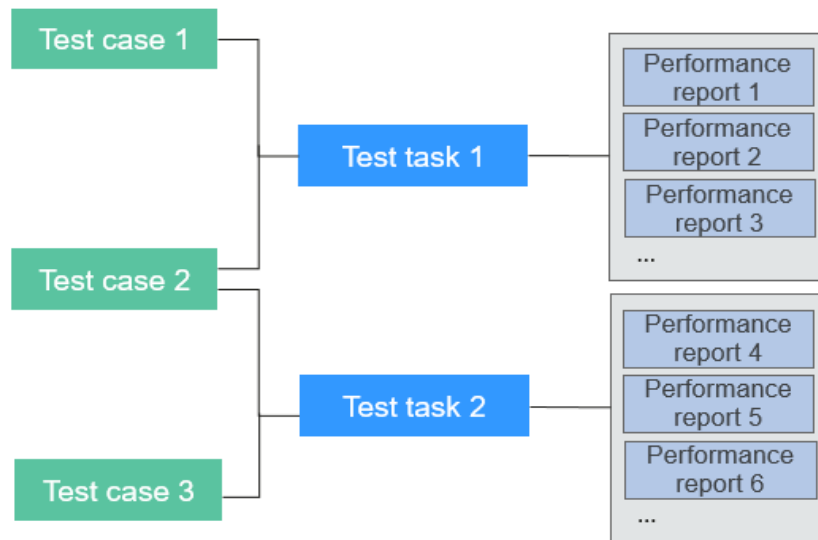
Test cases, test tasks, and performance reports are the three main steps of performance testing.

Figure 5-1 Performance testing process



A test case is a test model established based on a performance pressure test scenario. A test task is a performance test based on defined test models. A performance report is the execution result of a test task and displays the performance indicators of the tested system in high-concurrency scenarios.

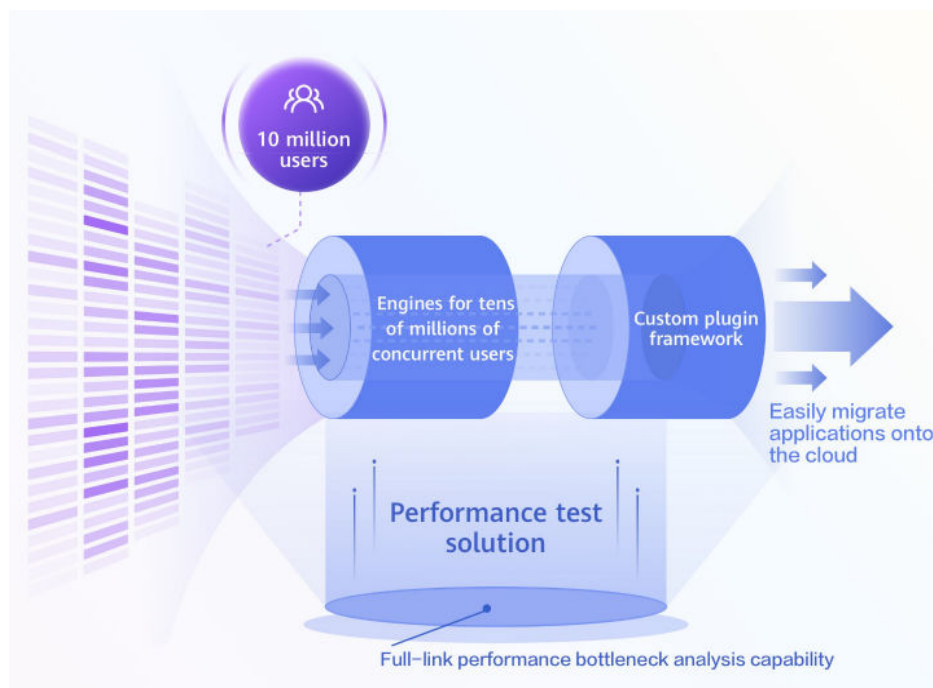
Figure 5-2 Relationships between test cases, test tasks, and performance reports



5.1.2 Pressure Test Engines for Tens of Millions of Concurrent Users

With the rapid increase of enterprise services, some large-scale service systems now have hundreds of millions of daily active users, which presents significant challenges to live network stability. Identifying performance challenges brought by high-concurrency services in advance has become a key focus for enterprises.

CodeArts PerfTest supports ultra-high concurrency of tens of millions of clusters with ultra-high instantaneous concurrency, gradient pressure increase, and dynamic pressure adjustment, meeting pressure test requirements of applications with hundreds of millions of daily active users. It supports custom plug-ins to interconnect with proprietary protocols and functions, and meet the requirements of performance pressure tests based on various protocols and complex scenarios. Therefore, enterprises can conduct high-concurrency tests on demand to proactively identify performance issues and ensure the quality of their products before they hit the market.



5.1.3 Eight Pressure Test Modes

Distilled from 30 years of high-concurrency test engineering solutions and practices, CodeArts PerfTest offers eight modes, such as surge (to simulate traffic bursts), intelligent peak load (to simulate system performance), oscillation (to simulate high and low peaks), and TPS mode (to customize pressure). You can create realistic scenarios quickly and increase the coverage of pressure test scenarios by 50%.



5.2 Directory Management of Test Cases

A test case directory is a display structure for organizing and managing test cases in a test project.

Prerequisites

A [PerfTest project](#) has been created.

Creating a Test Case Directory

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

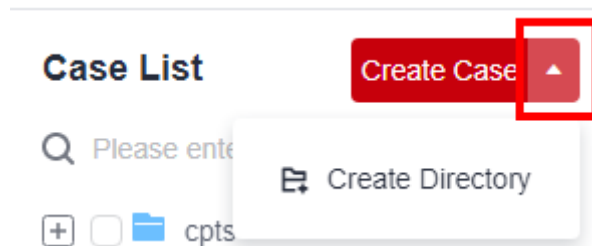
Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, create a directory.

- Create a case directory in the root directory.

Click the arrow on the right of **Create Case**. Click **Create Directory** from the drop-down list and enter the directory name.

Figure 5-3 Creating a case directory in the root directory



- Create a subdirectory in an existing directory.


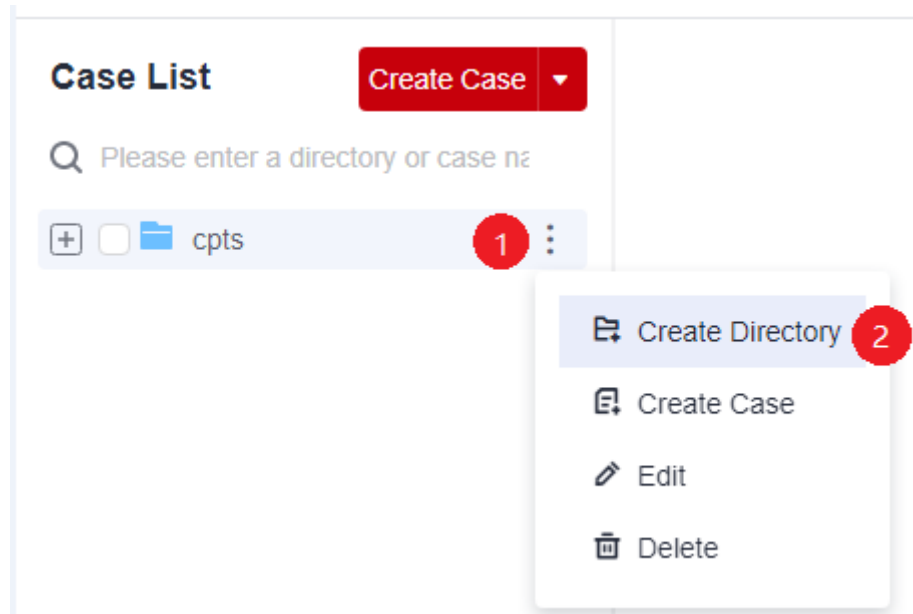

Click  next to an existing directory, choose **Create Directory** from the drop-down list, and enter the directory name.

Figure 5-4 Creating a subdirectory in an existing directory




----End

Modifying a Test Case Directory

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, view the created case directory structure on the left of the page.
- Step 4** Click  next to the directory to be modified, and choose **Edit** from the drop-down list.
- Step 5** Enter the new directory name in the text box and click another position on the page. The directory name is automatically saved.

----End

Deleting a Test Case Directory

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, view the created case directory structure on the left of the page.
- Step 4** Click  next to the directory to be deleted, choose **Delete** from the drop-down list, and delete the directory as prompted.

----End

5.3 Creating a Test Case

A test case is a test model established based on a performance pressure test scenario.

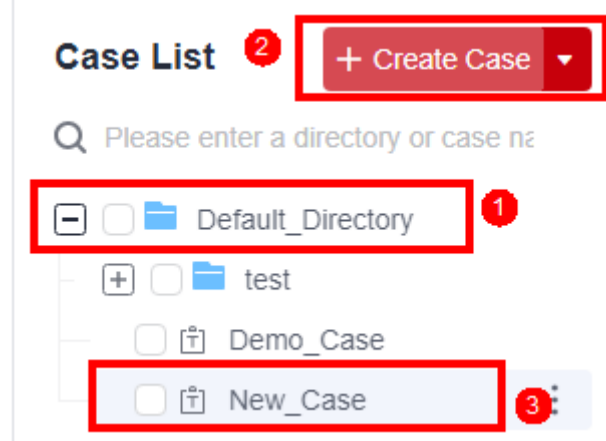
Prerequisites

- A **PerfTest project** has been created.
- A **test case directory** has been created.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, view the created case directory structure on the left of the page. You can create a case in either of the following ways:
 - To create a case in the current directory, click **Create Case**.

Figure 5-5 Creating a case in the current directory




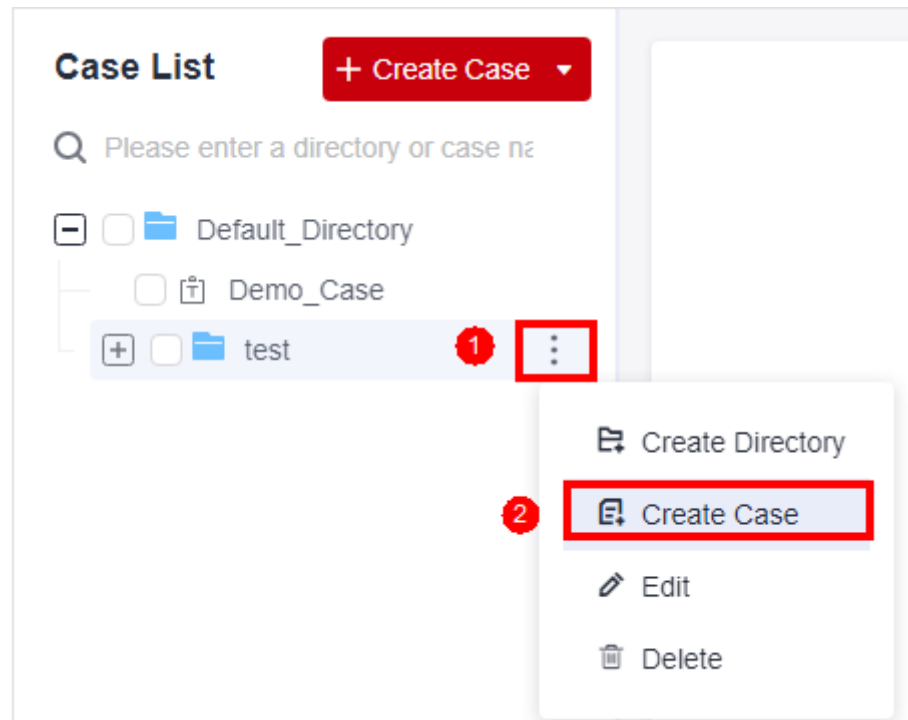
- To create a case in a specified directory, click  next to the directory, and choose **Create Case** from the drop-down list.

Figure 5-6 Creating a case in the specified directory



Step 4 Click **Save**.

----End

5.4 Configuring a Test Case (High-Performance Test Projects)


5.4.1 Filling in Case Details

Case details describe the background, preconditions and test steps of a test case to record its basic information for reuse. Case details are optional.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** Click **Case Details** on the right of the page. On the case details page that is displayed, enter the test case description, prerequisites, and procedure, and set basic case information based on section [Table 5-1](#).

Table 5-1 Basic case info

Parameter	Description
Status	Status of a test case, such as Creating, Designing, Testing, and Finished .
Case ID	ID of the test case.
Case Level	<ul style="list-style-type: none">• L0: verification of underlying functions. Each module should have 10 to 20 test cases. L0 test cases account for 5% of all test cases.• L1: basic function verification for inherited features or before iteration acceptance. L1 test cases account for 20% of all cases.• L2: important feature verification for manual tests in non-regression versions. L2 test cases account for 60% of all cases.• L3: verification of minor and non-important functions, and exception tests on basic and important functions. L3 test cases account for 10% to 15% of all cases.• L4: verification of special input, scenarios, and threshold conditions. L4 test cases account for less than 5% of all cases.
Label	A custom parameter that is used to identify a case. <ul style="list-style-type: none">• Enter a label: Click the label text box, enter a label, and press the space bar or Enter to generate a label. Each case can contain a maximum of 10 labels.• Delete a label: Click the label text box and click  next to a generated label to delete it.
Created at	Time when a test case was created. The value is automatically generated after the test case is saved.
Modified at	Time when a test case was last modified. The value is automatically generated after the test case is saved.

Step 5 After configuring the case details, click **Save**.

----End

5.4.2 Configuring a Case Script

After a test case is created, you can modify it as required.

Creating a Case Script

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 Click **Case Script** and select **Pre-Steps** or **Steps**.

Pre-Steps are executed only once in each execution thread. This configuration is optional.

Steps are executor of a test case, which needs to be configured.

Step 5 Add concurrency weight controllers, throughput controllers, requests, transactions, data sets, cycle controllers, condition judgments, and rendezvous points as required.

You can add both requests and transactions, or add either of them. In practice, data sets, cycle controllers, condition judgments, and rendezvous points are used together with requests.

- For details about how to add requests, see [Adding Request Information \(Packet\)](#), [Adding Request Information \(Think Time\)](#), [Adding Request Information \(Response Extraction\)](#), and [Adding Request Information \(Checkpoint\)](#).
- Add a transaction. A transaction mode already exists on the **Transactions** tab page. Click **Add Transaction**, select a transaction name, and click **OK**. For details about how to create a transaction, see [Creating a Transaction](#).
- For details about how to add data sets, cycle controllers, condition judgments, or rendezvous points, see [Adding a Data Instruction, Cycle Controller, Condition Judgment, or Rendezvous Point](#).
- For details about how to add concurrency weight controllers or throughput controllers, see [Adding Concurrency Weight Controllers and Throughput Controllers](#).

Step 6 When the configuration is complete, click **Save**.

----End

Skipping a Request Upon Failure


When a request fails during case execution, the case does not report an error to terminate the execution. Instead, the case continues to execute subsequent requests. The failure of a request does not affect the execution of subsequent key steps.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

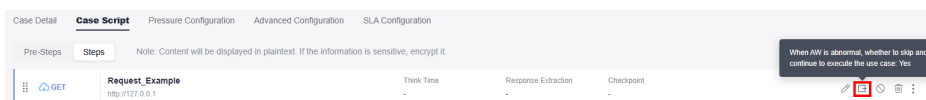
Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

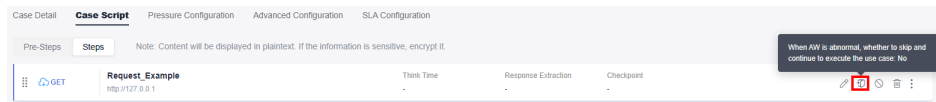
Step 4 On the **Case Script** tab page, find the desired request (AW).

Step 5 Click the second button () on the right of the request to configure whether to skip the AW request upon failure. The default value is **No**.

- Yes: Skip a failed request. The case execution is not interrupted.



- No: By default, a failed request is not skipped. The request failure does not interrupt case execution.

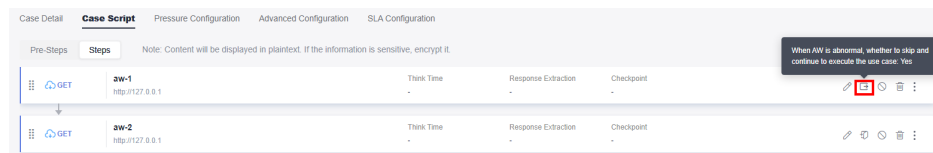


----End

Samples

If request **aw-1** is configured to be skipped upon failure, the system will skip the request failure and continue to execute the subsequent request **aw-2**.

Figure 5-7 AW that is configured to be skipped upon failure



Retrying a Request Upon Failure

When a request fails during case execution, the system retries the request based on the configuration, including the retry times and retry interval. If a request fails to be executed due to network or performance errors, configure retries for fault tolerance.

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** On the **Case Script** tab page, find the desired request (AW).
- Step 5** On the **Checkpoint** tab, enable **Enable Result Check** to configure retry upon failure.

----End

Samples

aw-1 is configured to be retried upon failure.

- If the **aw-1** request is successfully executed, the **aw-2** request will be executed without retrying **aw-1**.
- If the **aw-1** request fails, it will be retried three times at an interval of 3000 ms based on the checkpoint.
 - If the **aw-1** request is successful during the three retries, the **aw-2** request will be executed.
 - If all the three retries fail, the **aw-1** request fails, and the **aw-2** request will be interrupted.

Figure 5-8 Checkpoint enabled for the aw-1 request

GET aw-1
http://127.0.0.1

Packet Think Time Response Extraction **Checkpoint**

Enable Result Checkpoint

Retry Times 3

Retry Interval 3000 ms


At least one of the following three parameters must be set: response code, header, and content.

Response Code 200


Disabling/Enabling a Case Request

If you want to retain a request of a case but do not want to use it, you can disable it. If you want to resume the use of the request, you can enable it again.

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** Disable a case request.


On the **Case Script** tab, click  next to the desired request to disable it. Other requests are not affected.


- Step 5** Enable a case request.

On the **Case Script** tab, click  next to the desired request to enable it. Other requests are not affected.

----End

Copying/Pasting a Case Request

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** On the **Case Script** tab, click  > **Copy** next to the desired request. **The copy is successful!** is displayed in the upper right corner of the page.

Step 5 You can select a request in any project under **PerfTest Projects**, click  > **paste to front** or **paste to behind** next to the request.

----End

5.4.3 Adding Request Information (Packet)

Packets are data blocks transmitted between HTTP/HTTPS/TCP/UDP/HLS/RTMP/WebSocket/HTTP-FLV/MQTT-based applications. These data blocks start with text metadata that describes the packet content and meaning. The text metadata is followed by optional data. Packets are transmitted among clients, servers, and agents.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** On the **Case Script** tab page, select a case, and click **Add Request**.
- Step 5** On the **Packet** tab page, select a protocol type based on the service protocol type. Currently, HTTP, HTTPS, TCP, UDP, HLS, RTMP, WebSocket, MQTT, and HTTP-FLV are supported. Set the basic information according to the following packet parameter tables.

Table 5-2 HTTP and HTTPS packet parameters

Parameter	Description
Request mode	GET, POST, PATCH, PUT, or DELETE.
Request URL	URL for sending a request, for example, http://domain name/path , or http://domain name/path?key1=value1&key2=value2 . Variables can be inserted. For details, see Inserting a Variable .
Response Timeout (ms)	Timeout period for waiting for a response from the server when a request is sent. The timeout value ranges from 20 ms to 5 min. If this parameter is not set, the default response timeout interval (5,000 ms) is used.

Parameter	Description
Carried cookie	<p>NOTE</p> <p>At least one request exists in the case. Set parameters when adding new requests.</p> <ul style="list-style-type: none">● Obtain automatically: Use the cookie set in the response.● Obtain manually: Use the cookie in the header of the current request.
Automatic Redirection	Whether to redirect to the redirection address. The function is enabled by default, that is, the redirection address is automatically redirected to.
Request Parameter	<p>Set the parameters in the request URL.</p> <p>Click Add Request Parameter and set Name and Value. Select Url Encode if necessary. By default, Url Encode is not selected. If Url Encode is selected, URL encoding is performed on the values of request parameters.</p> <p>If you do not need to add a request parameter, click Delete to delete it.</p>
RequestHeader	<p>Add headers and contents based on the headers to be verified or used by a pressure test server. Headers are not mandatory. Only user-defined headers are transparently transmitted to the pressure test server. For details about Header, see Header Description.</p> <p>Click Add Header and set Header and Value. If you do not need to add header information, click Delete to delete the packet header.</p> <p>You can click Bulk Edit to edit multiple headers at a time in the text box. You can click Key-Value Edit to return to the key-value pair editing mode.</p> <p>NOTE</p> <p>If the request mode is POST or PUT and Header is Content-Type, variables can be inserted, and Value can be one of the three following types:</p> <ul style="list-style-type: none">● Customize: Enter a value in the text box.● application/x-www-form-urlencoded: The request body is a key value pair that can be added. The value is text.● multipart/form-data: The request body is a key-value pair that can be added. The value can be text or a file. If the value is a file, the body can be imported.

Parameter	Description
Body	<p>The body of an entity contains a data block consisting of random data. Not all packets contain the body of an entity.</p> <p>If global variables have been set or local variables have been set for response extraction, variables can be used in the packet content. During a pressure test, variables in the packet content will be replaced with specified values.</p> <ol style="list-style-type: none">1. Enter \$ in the request content input box.2. In the Insert Variable dialog box, set parameters. For details, see Inserting a Variable.3. Click Select.

Table 5-3 TCP packet parameters

Parameter	Description
IP	IP address of the tested server to which requests are sent.
Port Number	Port number of the tested server to which requests are sent.
Connection Timeout	Timeout duration for the server's response after a connection is initiated. The connection timeout value ranges from 20 ms to 60s.
Response Timeout	Timeout duration for waiting for the server's response after a connection is established. The response timeout value ranges from 20 ms to 60s.
Connection Settings	<ul style="list-style-type: none">● Repeated use: When a request response is complete, the connection remains and is used to send and receive the next request response.● Close: When a request response is complete, the connection is closed and re-established next time.

Parameter	Description
Return Settings	<p>Used to judge whether a request response has been received.</p> <ul style="list-style-type: none">• Length of returned data: length of the returned data, in bytes. When a response of this length is received, data receiving is complete.• End-of-text character: ending mark of the returned data. When an end-of-text character is received, data receiving is complete. <p>NOTE It is recommended that a unique end-of-text character be set. If there are multiple such characters in a response, when the first character is received, the response is considered to have been received. As a result, the received response data is incomplete.</p>
Content Format	<p>Select String or Hexadecimal code stream data based on the service request content of the tested server.</p> <p>NOTE A hexadecimal code stream can contain only numbers 0–9 and letters a–f. The total number of characters must be an even number.</p> <p>If global variables have been set or local variables have been set for response extraction, variables can be used in the packet content. During a pressure test, variables in the packet content will be replaced with specified values.</p> <ol style="list-style-type: none">1. Enter \$ in the request content input box.2. In the Insert Variable dialog box, set parameters. For details, see Inserting a Variable.3. Click Select.

Table 5-4 UDP packet parameters

Parameter	Description
IP	IP address of the tested server to which requests are sent.
Port Number	Port number of the tested server to which requests are sent.

Parameter	Description
Return Settings	<p>Used to judge whether a request response has been received.</p> <ul style="list-style-type: none"> • Length of returned data: length of the returned data, in bytes. When a response of this length is received, data receiving is complete. • End-of-text character: ending mark of the returned data. When an end-of-text character is received, data receiving is complete. <p>NOTE It is recommended that a unique end-of-text character be set. If there are multiple such characters in a response, when the first character is received, the response is considered to have been received. As a result, the received response data is incomplete.</p>
Content Format	<p>Select String or Hexadecimal code stream data based on the service request content of the tested server.</p> <p>NOTE A hexadecimal code stream can contain only numbers 0–9 and letters a–f. The total number of characters must be an even number.</p> <p>If global variables have been set or local variables have been set for response extraction, variables can be used in the packet content. During a pressure test, variables in the packet content will be replaced with specified values.</p> <ol style="list-style-type: none"> 1. Enter \$ in the request content input box. 2. In the Insert Variable dialog box, set parameters. For details, see Inserting a Variable. 3. Click Select.

Table 5-5 HLS packet parameters

Parameter	Description
Streaming media address	Address of the video source, for example, http://domain name/path .
Play Duration (s)	<p>Duration of simulated playback.</p> <p>NOTE During case debugging, the playback duration is set to 3 seconds.</p>
Retry Delay (ms)	Interval for obtaining new data in live streaming when the video playback source does not have new data.

Parameter	Description
Retry Times	Maximum number of attempts to obtain new data in live streaming when the video playback source does not have new data. If the number of attempts exceeds the maximum, the system determines that the attempt fails.

Table 5-6 RTMP packet parameters

Parameter	Description
Request Mode	<ul style="list-style-type: none">• Stream pushing: a process of transmitting onsite video signals to the network.• Stream pulling: a process of pulling streaming media video files from a server using a specified address.
When Request Mode is Stream pushing , set the following parameters.	
Streaming server address	IP address of the tested streaming media server.
Port Number	Listening port number of the tested streaming media server. The default value is 1935 for RTMP.
App name	AppName in the stream URL, that is, the path for storing live streaming media files, for example, live .
Streaming name	StreamName in the stream URL, which uniquely identifies a live stream, for example, livestream .
Push duration (s)	Duration for simulating stream pushing, in seconds.
Video source address	OBS address for storing video source files for simulating live stream pushing. Currently, only .flv files are supported. Non-standard H.265-encoded .flv files are not supported.
When Request Mode is Stream pulling , set the following parameters.	
Streaming server address	IP address of the tested streaming media server.
Port Number	Listening port number of the tested streaming media server. The default value is 1935 for RTMP.
App name	AppName in the stream URL, that is, the path for storing live streaming media files, for example, live .
Streaming name	StreamName in the stream URL, which uniquely identifies a live stream, for example, livestream .

Parameter	Description
Play Duration (s)	Simulated duration of live stream watching, in seconds.

Table 5-7 WebSocket packet parameters

Parameter	Description
Request mode	Connect: Establish a WebSocket connection with the pressure test website. Disconnect: Close the connection to the pressure test website. Pong: Send pong to the pressure test website. Ping&Pong: Send pong to the pressure test website and expect to return ping . Request: Send a request to the pressure test website and receive a response. ReadOnly: Only receive information sent by the pressure test website. WriteOnly: Only send information to the pressure test website.
Request connection address	URL of the WebSocket connection, for example, ws://domain name/path . Encrypted requests are supported, such as wss://domain name/path .
Connection Timeout (ms)	Timeout duration for the server's response after a connection is initiated. The connection timeout value ranges from 20 ms to 60s. NOTE This parameter needs to be set only when the request mode is Connect .
Status Code	Status code for closing a connection. The default value is 1,000 . The value range is 0–4,999. NOTE This parameter needs to be set only when the request mode is Disconnect .
Request Type	Select Text or Binary based on the service request content of the tested server. NOTE This parameter needs to be set only when the request mode is Request or WriteOnly .
Bodys	Enter the packet content of request information based on the specified request type. NOTE This parameter needs to be set only when the request mode is Request or WriteOnly .

Parameter	Description
Response Type	Select Text or Binary based on the service response content of the tested server. NOTE This parameter needs to be set only when the request mode is Request or ReadOnly .
Response Timeout (ms)	Timeout duration for waiting for the server's response after a connection is established. The response timeout value ranges from 20 ms to 60s.
Request Header	Add headers and contents based on the headers to be verified or used by a pressure test server. Headers are not mandatory. Only user-defined headers are transparently transmitted to the pressure test server. For details about Header , see Header Description . Click Add Header and set Header and Value . If you do not need to add header information, click Delete to delete the packet header. You can click Bulk Edit to edit multiple headers at a time in the text box. You can click Key-Value Edit to return to the key-value pair editing mode.

Table 5-8 HTTP-FLV packet parameters

Parameter	Description
Streaming media address	Address of the video source, for example, http://{Video source IP address}:8080/live/stream.flv .
Play Duration (s)	Duration of simulated playback. NOTE During case debugging, the playback duration is set to 3 seconds.

Table 5-9 MQTT packet parameters

Parameter	Description
Request mode	Connect : Establish an MQTT connection with the pressure test website. Publish : Publish a message. Subscribe : Subscribe to sent messages. Disconnect : Close the MQTT connection to the pressure test website.
When the request mode is Connect , set the following parameters.	

Parameter	Description
MQTT Version	MQTT version. Currently, 3.1 and 3.1.1 are supported.
Transport Address	MQTT server address, which supports the TCP, SSL, WS, and WSS protocols.
Timeout interval	Timeout interval for the client to perform operations such as establishing a connection and sending a message.
Client ID	MQTT client ID. You can set whether to add a random suffix to the client ID by enabling or disabling Add a random suffix .
Username	Username set for the MQTT client. This parameter is sensitive. Set it using global variables in sensitive parameter mode.
Password	Password set for the MQTT client. This parameter is sensitive. Set it using global variables in sensitive parameter mode.
Connection keepalive	Active heartbeat interval, in seconds.
Clear Sessions	<ul style="list-style-type: none">• If this option is toggled on, the reserved session information of the last connection will be cleared when a connection is established between the client and the server.• If this option is toggled off, the reserved session information of the last connection will not be cleared when a connection is established between the client and the server.

Parameter	Description
Add Will	<ul style="list-style-type: none">You can toggle on this option to add a will. In this way, when the client is disconnected abnormally, the MQTT agent automatically publishes the will message to other subscribers. Set the following parameters when adding a will:<ul style="list-style-type: none">Theme: topic of the will.Type: Text or Binary. You can also set whether to add a timestamp.Content: content of the will.QoS Level: quality of service (QoS) level, which is used to describe the reliability of message transmission. Send at least once: applicable to real-time data transmission. Send at least once: applicable to applications that require reliability but allow slight delay. Only once: applicable to applications that require high reliability and long delay.Keep your will: If this option is toggled on, the client receives the set will message immediately after subscription. If this option is toggled off, the subscribing client receives the will message only after the client that publishes the will message is disconnected unexpectedly.If you toggle off Add Will, no will message will be sent.
If the request mode is Publish , set the following parameters:	
Theme	Topic of the message to be published.
QoS Level	QoS level, which is used to describe the reliability of message transmission. Send at least once: applicable to real-time data transmission. Send at least once: applicable to applications that require reliability but allow slight delay. Only once: applicable to applications that require high reliability and long delay.
Request Type	Text or Binary. You can also set whether to add a timestamp.
Request Content	Detailed content of the request.
Timeout interval	Timeout interval for the client to send a message.

Parameter	Description
Reserved Message	<ul style="list-style-type: none">• If this option is toggled on, messages to be published when the client is disconnected are stored on the server.• If this option is toggled off, messages to be published when the client is disconnected are cleared.
When the request mode is Subscribe , set the following parameters.	
Subscribe to Topic	Topic of the subscribed message, which must match the topic of the published message.
QoS Level	QoS level, which is used to describe the reliability of message transmission. Send at least once : applicable to real-time data transmission. Send at least once : applicable to applications that require reliability but allow slight delay. Only once : applicable to applications that require high reliability and long delay.
Response Type	Text or Binary . You can also set whether to add a timestamp.
Timeout interval	Timeout interval of the subscription topic.
End Condition	When this condition is met, the subscription ends. <ul style="list-style-type: none">• Maximum triggering time: The subscription ends when the time reaches the upper limit.• Maximum Number of Received Messages Triggered: The subscription ends when the number of received messages reaches the upper limit.
If the request mode is Disconnect , you do not need to set parameters.	

Step 6 Click **Save**.

----End

Filling in a Packet

Packets refer to all click operations performed by users on the website interface. The process of sending a packet: A click operation is edited to a code stream complying with protocol specifications and carrying a user's request before the code stream is sent to a third party, leading to a correct or failed response.

Press **F12** or use a packet-capturing tool (such as Wireshark) to check how a packet is requested and fill in the packet to be tested according to the actual service.

5.4.4 Adding Request Information (Think Time)

To better simulate user behavior, insert a waiting time between different operations. For example, when a user receives data from the server, the user may wait several seconds before viewing data and providing responses. This period of time is called think time.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** On the **Case Script** tab page, select a case, and click **Add Request**.
- Step 5** On the **Think Time** tab page, set the basic information by referring to [Table 5-10](#).

Table 5-10 Think time parameters

Parameter	Description
Enable Think Time	After think time is enabled, you may wait for several seconds before viewing the data received from the server, and then respond.
Duration (ms)	Think time refers to the waiting time between two consecutive operations of a user, such as the interval between login and search. For example, the response time for each case execution is 0.5s. <ul style="list-style-type: none">• To execute two case requests per second, do not add a case request whose type is think time.• To execute only one case request per second, set Duration to 1s. If the think time is set to 1s and the response time is longer than 1s, the think time will not take effect, and pressure test requests are sent according to the response time.

- Step 6** When the configuration is complete, click **Save**.

----End

5.4.5 Adding Request Information (Response Extraction)

If multiple packets exist in the same case, use regular expressions or JSON to extract the output of the previous packet for the input of the next packet.

For example, in an e-commerce flash sale scenario, after you search for a product and purchase it, you can use the response extraction function to extract the product ID obtained from packet search and use the ID as the input parameter of the next purchase packet.

 NOTE

- Cross-case extraction is not supported. That is, the output extracted from case A's packet cannot be used for the input of case B's packet.
- The variables from a response extraction are local variables.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** On the **Case Script** tab page, select a case, and click **Add Request**.
- Step 5** On the **Response Extraction** tab page, set the basic information by referring to [Table 5-11](#).

Table 5-11 Response extraction parameters

Parameter	Description
Enable Response Extraction	After response extraction is enabled, if multiple packets exist in the same case, you can use regular expressions or JSON to extract the output of the previous packet for the input of the next packet.
Variable Name	Must be unique. The value of the response extraction is assigned to this variable.
Expected Value	Expected variable value of response extraction. If you enable this option, the extracted value is compared with the expected value. If they are different from each other, the URL verification will fail.
Extraction Range	Range of the content to be extracted. The packet content, header, and URL can be extracted using regular expressions.
Regular Expression	<p>A regular expression specifies a type of logical expressions performed on strings. In a regular expression, you use certain predefined strings and a combination of these strings to form a rule string that is used to specify a specific filtering logic.</p> <p>A complete regular expression consists of two types of characters: special characters (also called meta characters) and literal or common text characters (such as letters, numbers, and underscores). For details about meta characters, see Regular Expression Metacharacters.</p> <p>NOTE () indicates extraction. Enclose the content to be extracted in parentheses. The content between each () indicates a sub-expression.</p>

Parameter	Description
Sequence Number of Matching Item	This parameter indicates the sequence number of the matched content extracted through a regular expression. Value range: positive integers NOTE This parameter cannot be set to 0 .
Expression Value	This parameter indicates the sequence number of the parsed sub-expression. Value range: natural numbers NOTE <ul style="list-style-type: none"> Value 0 indicates that the entire regular expression is matched. Value 1 indicates that the first sub-expression of the regular expression is matched, that is, the content extracted by the first "()". After extracting content through Regular Expression and Sequence Number of Matching Item , use Expression Value to obtain the final content.
Key Name	Enter the key name to be obtained. For example, to extract v42 from {"key":{"key1":"v1","key2":{"key3":"v3"},"key4":[{"key41":"v41","key42":"v42"}, {"key41":"v43","key42":"v44"}]}}, enter key.key4[0].key42 . NOTE This parameter is valid only when Extraction Range is set to Parameter values in .json files .
Default Value	Value extracted when a regular expression match or JSON extraction fails.
Condition Expression	This parameter is used together with the Key Name to be obtained. For example, {"key":{"key1":"v1","key2":{"key3":"v3"},"key4":[{"key41":"v41","key42":"v42"}, {"key41":"v43","key42":"v44"}]}}, If you want to describe when key42 is equal to v42, extract the target value v41 , you can enter key.key4[].key42 = v42 for Condition Expression , and enter key.key4[].key41 for Key Name .

Table 5-12 Common regular expressions

Regular Expression	Description	Example
(\d+)	Matches non-negative integers.	String: bTivm2wu9jih1LBKR4osZGrjil

Regular Expression	Description	Example
		Match results: 2 9 1 4
([A-Za-z]+)	Matches a string of letters.	String: bTivm2wu9jih1LBKR4osZGrjll Match results: bTivm wu jih LBKR osZGrjll
([A-Za-z0-9]+)	Matches a string of digits and letters.	String: bTivm2wu9jih1LBKR4osZGrjll:asd22 Match results: bTivm2wu9jih1LBKR4osZGrjll asd22
(\w+)	Matches a string of digits, letters, or underscores (_).	String: bTivm2wu9jih1LBKR4osZGrjll:asd22 Match results: bTivm2wu9jih1LBKR4osZGrjll asd22
([\w-]+(\.[\w-]+)+)*@([\w-]+(\.[\w-]+)+)	Matches email addresses.	String: bTivm2wu9jih1LBKR4osZGrjll:abc@abc.com Match results: abc@abc.com

Step 6 (Optional) A request can contain one or more variables. If you need to add multiple variables for a request, click **Add Variable** and set related parameters.

Step 7 When the configuration is complete, click **Save**.

----End

Examples of Regular Expression Response Extraction

Example 1

The response content of the previous packet is as follows:

```
"baseInfo" : {
"mobilephone": "xxxxxxxxxx",
"Telephone": "xxxxxxxxxx",
"unitGuid": "xx",
"unitMame": null,
"address": "xxx",
"sex" : 0,
"imageUr1" : nul1
},
"UserNotices": null
},
```

```
"msgId" : "64xxxxxxxxxxxxxxxxxxxxf5",
"isUsed" : "1"
"token": "eyJxxxxxxxJ9.ejzdW1101JYZMiLCJcUbdGUkIJeZMT!!zz#z!
20TxxxxxxxxxxxxxxxxS1611dYQJSHTFKifVOslmd1aWQiOiIiODkiLCJleHAiOiJleIMzliZnZyY1MjZ9.myU5idiASM-1
1@EP7YQTfTsR_8zsq7?sbYJYoxfRsuf6OZhgL-
XWmjndaviGauhSdw16ImWOFevbACSHMXGT1U0ijS5z6ezX@sZePruzFncvIMgShF8xNPN6zVokQp-
uwbyS3W6NpZpDuwsjuiZ7DZTNpKqCkGHwvPjrHBOvWFR_u6-FBbTiFiqdhQb95U-1gLiLvoZHY_rguzwyrZ-
leGRdCG_ZASreoWC-
uH)HnqltpgltrChWQToHQyxOABdMSbBShhNctBBZHgQPMESqQQQTbBiPGvbQDprB7ZBFMUB_ShynS_evtyfE
ladGEddhOBn-fxxxxxxxx"
},
```

To extract the token value, use the regular expression **"token"\s*:\s*(.*)"**. The configuration of response extraction is shown as follows:

Example 2

The response content of the previous packet is as follows:

```
javawind:9javawind:12
javawind:16javawind:17
javawind:46javawind:22
```

To extract the value 16, see the following figure.

1. The following data is extracted using the regular expression **javawind:(.*)javawind:(.*)**.
9 12
16 17
46 22
2. The following data is extracted using the second matching item.
16 17
3. The following data is extracted using the first expression.
16

Example 3

The packet content is **ababdacac**.

Response extraction settings:

Variable Name: x Expected Value: Enter an expected value. Delete

Extraction Range: Packet content Header URL Response code Parameter values in .json files

Regular Expression: (ab|ac)+ Sequence Number of Matching Item: 1

Expression Value: 0 Key Name: Enter a key name.

Default Value: not found Condition Expression: Enter a condition express

Regular Expression: (ab|ac)+ indicates that multiple **ab** or **ac** values are matched.

Sequence Number of Matching Item: **1**, indicating that the matching item is matched by **ab**. **2**, indicating that the matching item is matched by **ac**. **1** may be used to obtain **abab** and a substring **ab**, and **2** may be used to obtain **acac** and a substring **ac**.

Expression Value: The value **0** indicates that the maximum matching string **abab** or **acac** is used. The value **1** indicates that the substring **ab** or **ac** is used.

Example 4

The extracted content is **Content-Type** in the HTTP response header.

Variable Name: x Expected Value: Enter an expected value. Delete

Extraction Range: Packet content Header URL Response code Parameter values in .json files

Regular Expression: Content-Type: (.*)\r\n Sequence Number of Matching Item: 1

Expression Value: 0 Key Name: Enter a key name.

Default Value: not found Condition Expression: Enter a condition express

Regular Expression: Content-Type: (.*)\r\n. Note that there is a space after the colon (:) and \r\n is at the end, which is generated based on HTTP specifications.

Sequence Number of Matching Item: **1**, because there is only one (.*) .

Expression Value: **1**, indicating that the expected value is obtained.

Examples of JSON Expression Response Extraction

The sample data is as follows:

```
{
  "name": {
    "first": "Tom",
    "last": "Anderson"
  },
  "age": 37,
  "children": ["Sara", "Alex", "Jack"],
  "fav.movie": "Deer Hunter",
  "friends": [{
    "first": "Dale",
    "last": "Murphy",
    "age": 44,
    "nets": ["ig", "fb", "tw"]
  },
  {
    "first": "Roger",
    "last": "Craig",
    "age": 68,
    "nets": ["fb", "tw"]
  }
],
  "first": "Jane",
```

```

    "last": "Murphy",
    "age": 47,
    "nets": ["ig","tw"]
  }}
}

```

Example 1

To extract the **first name**, set **Key Name** to **name.first**.

Configuration for Example 1:

- Variable Name: x
- Expected Value: Enter an expected value.
- Extraction Range: Parameter values in json files
- Regular Expression: Enter a regular expression
- Sequence Number of Matching Item: Enter the sequence number
- Expression Value: Enter an expression value
- Key Name: name.first
- Default Value: not found
- Condition Expression: Enter a condition expression

Example 2

To extract **the number of Tom's children**, set **Key Name** to **children.#**.

Configuration for Example 2:

- Variable Name: x
- Expected Value: Enter an expected value.
- Extraction Range: Parameter values in json files
- Regular Expression: Enter a regular expression
- Sequence Number of Matching Item: Enter the sequence number
- Expression Value: Enter an expression value
- Key Name: children.#
- Default Value: not found
- Condition Expression: Enter a condition expression

Example 3

To extract **the name of Tom's second child**, set **Key Name** to **children.1**. (Sequence number starts from 0.)

Configuration for Example 3:

- Variable Name: x
- Expected Value: Enter an expected value.
- Extraction Range: Parameter values in json files
- Regular Expression: Enter a regular expression
- Sequence Number of Matching Item: Enter the sequence number
- Expression Value: Enter an expression value
- Key Name: children.1
- Default Value: not found
- Condition Expression: Enter a condition expression

Example 4

To extract **the last names of Tom's friends older than 45**, set **Key Name** to **friends.#(age>45)#.last**.

Configuration for Example 4:

- Variable Name: x
- Expected Value: Enter an expected value.
- Extraction Range: Parameter values in json files
- Regular Expression: Enter a regular expression
- Sequence Number of Matching Item: Enter the sequence number
- Expression Value: Enter an expression value
- Key Name: friends.#(age>45)#.last
- Default Value: not found
- Condition Expression: Enter a condition expression

Example 5

To extract the first name of Tom's friend whose last name is Murphy, set Key Name to `friends.#(last=="Murphy").first`.

The screenshot shows the configuration interface for a variable named 'x'. The 'Extraction Range' is set to 'Parameter values in json files'. The 'Regular Expression' field is empty. The 'Sequence Number of Matching Item' field is empty. The 'Expression Value' field is empty. The 'Default Value' field is set to 'not found'. The 'Key Name' field is set to 'friends.#(last=="Murphy")'. The 'Condition Expression' field is empty.

Example 6

To extract the second nickname of Tom's first friend, set Key Name to `friends.0.nets.1`. (Sequence number starts from 0).

The screenshot shows the configuration interface for a variable named 'x'. The 'Extraction Range' is set to 'Parameter values in json files'. The 'Regular Expression' field is empty. The 'Sequence Number of Matching Item' field is empty. The 'Expression Value' field is empty. The 'Default Value' field is set to 'not found'. The 'Key Name' field is set to 'friends.0.nets.1'. The 'Condition Expression' field is empty.

Example 7

To extract the first name of Tom's friend whose nickname is fb, set Key Name to `friends.#(nets.#(=="fb")).first`.

The screenshot shows the configuration interface for a variable named 'x'. The 'Extraction Range' is set to 'Parameter values in json files'. The 'Regular Expression' field is empty. The 'Sequence Number of Matching Item' field is empty. The 'Expression Value' field is empty. The 'Default Value' field is set to 'not found'. The 'Key Name' field is set to 'friends.#(nets.#(=="fb"))'. The 'Condition Expression' field is empty.

5.4.6 Adding Request Information (Checkpoint)

Checkpoint uses customized verification information to check whether the content returned by a server is correct. For different protocol types, the checkpoint supports different contents. The HTTP and HTTPS protocols support the verification of response code, header, and content. The TCP, UDP, MQTT, and WebSocket protocols support only content verification.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab page, select a case, and click **Add Request**.

Step 5 On the **Checkpoint** tab page, set the basic information by referring to [Table 5-13](#).

Table 5-13 Checkpoint parameters

Parameter	Description
Enable Result Checkpoint	After this option is enabled, you can check whether the content returned by a server is correct through customized verification information.
Retry Times	Number of times to retry a failed request until it succeeds. The value ranges from 0 to 100. NOTE Statistics such as response time, RPS, and success rate of retried requests are not included in the report. The report counts only the last data.
Retry Interval	Interval for retrying a failed request. The unit is ms. The value ranges from 0 to 3,600,000, that is, the maximum interval is 1 hour.
Response Code	Set this parameter only when the protocol type is set to HTTP or HTTPS on the Packet tab page. This parameter is the HTTP or HTTPS response status codes, including 1XX , 2XX , 3XX , 4XX , and 5XX , carried in a packet.
Header	Set this parameter only when the protocol type is set to HTTP or HTTPS on the Packet tab page. Headers of HTTP and HTTPS. <ol style="list-style-type: none">1. Click Add Header.2. Enter a Header Name. For details, see Header Description.3. Set check conditions.4. Set contents.
Response Content	This parameter refers to the body parts of HTTP, HTTPS, TCP, UDP, MQTT, and WebSocket, namely the load parts of the requests and responses. <ol style="list-style-type: none">1. Click Add Content.2. Set check conditions.
Condition Rule	<ul style="list-style-type: none">● AND: The verification is successful only when all the check conditions are met.● OR: The verification is successful when any check condition is met.

Step 6 When the configuration is complete, click **Save**.

----End

5.4.7 Adding a Data Instruction, Cycle Controller, Condition Judgment, or Rendezvous Point

Data Instruction

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, choose **Add More > Data Set**.

Step 5 Click **Add Param**, and configure **Name**, **Logic**, and **Mode**.

- **Name**

Name of the parameter to which the parameter logic result will be assigned as the parameter value.

- **Logic**

- To assign a constant value to a parameter, enter a character string, for example, **1.23** or **abc**.
- To assign a variable value to a parameter, enter **\$** in the text box. In the displayed **Insert Variable** dialog box, select a variable type.

If you set **Type** to **Custom**, set the following parameters:

- Range**: variable range.
- Name**: name of a custom variable. Specifies the name of a variable when the variable is added.

If you set **Type** to **System**, select a function. For details, see [Table 5-14](#).

Table 5-14 Functions

Name	Description
ID card number	Randomly generates an ID number.
Mobile phone number	Randomly generates a phone number.
Random number in a range	Randomly generates an integer within your specified range.
Random character string	Randomly generates a string consisting of lowercase letters and numbers based on your specified string length (1 to 32 characters).
Time stamp	Generates a timestamp for the current time based on your specified unit. A 10-digit timestamp represents time in seconds whereas a 13-digit timestamp represents time in milliseconds.

Name	Description
Datetime Format	The current time is output in the specified format. Time offsets in seconds are supported.
UUID	Randomly generates a string of 32 characters.

If you set **Type** to **Four Fundamental rules**, select a function.


- i. Select **+**, **-**, **x**, **/**, or **Scale Mode** from the drop-down list box, and click **Select**.
- ii. Click . In the displayed **Four Functions Editing** dialog box, set the following parameters:
 - **Expression**: Enter operands for the arithmetic operation. An operand can be a number (such as 1.23), a global, local, or system variable. You can reselect the operation here.
 - **Mode**: Select how the expression result will be rounded. For details, see [Table 5-15](#).

Table 5-15 Rounding modes

Name	Description (Using Two Decimal Place Rounding as an Example)
Far-zero Rounding	Rounds the number in a way that the result heads away from zero. For example, 1.234 is rounded to 1.24, and -1.234 is rounded to -1.24.
Near-zero Rounding	Rounds the number in a way that the result heads towards zero. For example, 1.234 is rounded to 1.23, and -1.234 is rounded to -1.23.
Round Up	Rounds the number towards positive infinity. Positive numbers will be rounded up, whereas the unwanted digits of negative numbers will be simply discarded. For example, 1.234 is rounded to 1.24, and -1.234 is rounded to -1.23.
Round Down	Rounds the number towards negative infinity. The unwanted digits of positive numbers will be simply discarded whereas negative numbers will be rounded down. For example, 1.234 is rounded to 1.23, and -1.234 is rounded to -1.24.

Name	Description (Using Two Decimal Place Rounding as an Example)
Rounding	If the first digit to be discarded is smaller than 5, the digit and all the digits that follow are simply discarded. Otherwise, the last retained digit is increased by 1 in value. For example, 1.234 is rounded to 1.23 and -1.235 is rounded to -1.24.
Rounded to six	If the first digit to be discarded is smaller than 6, the digit and all the digits that follow are simply discarded. Otherwise, the last retained digit is increased by 1 in value. For example, 1.235 is rounded to 1.23 and -1.236 is rounded to -1.24.
Bankers	Rounds the number to its nearest even integer. For example. 1.2350000 is rounded to 1.24, 1.2250000 is rounded to 1.22, and 1.2250001 is rounded to 1.23.
No Rounding	The number is not rounded. NOTE If this mode is selected, no further action is required.

- **Precision:** The maximum number of decimal places is 32.
- iii. When the configuration is complete, click **OK**.

 **NOTE**

If you specify multiple logics for a parameter, all the logics are combined as a character string and the string is assigned to the parameter as its value.

When the variable type is the encryption function, you need to select the encryption type and the position where the variable is inserted. The encryption types are as follows:

- MD5: Encrypts the val data using MD5 and returns the encrypted data string.
- MD5-BASE64: Performs MD5-BASE64 encoding on val data and returns the encoded data string.
- SHA-1: Encrypts the input data using SHA-1 and returns the encrypted data string.
- SHA-224: Encrypts the input data using SHA-224 and returns the encrypted data character string.
- SHA-256: Encrypts the input data using SHA-256 and returns the encrypted data character string.
- SHA-384: Encrypts the input data using SHA-384 and returns the encrypted data string.

- SHA-512: Encrypts the input data using SHA-512 and returns the encrypted data character string.
- BASE64 encoding: Encodes the input data using BASE64 and returns the encoded data string.
- BASE64 decoding: Performs BASE64 decoding on the input data and returns the decoded data character string.
- **Mode**
 - If you select **Repeated assignment** from the drop-down list box, a value is assigned each time the parameter value is needed in a thread.
 - If you select **Assigned once** from the drop-down list, the parameter value is assigned only once in a thread.

Step 6 When the configuration is complete, click **Save**.

----End

Cycle Controller

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired case in the case list on the left.

Step 4 On the **Case Script** tab, choose **Add More > Cycle Controller**.

Step 5 Set parameters.

Number of Cycles: Enter a value between 1 and 1,000,000.

Exit Conditions: If the request meets the condition, the cycle ends. Both expressions and comparisons support variables and character strings. Currently, conditions cannot be combined. If you need to use a set or multiple comparisons, set a regular expression.

Step 6 Click **Add More** to specify more details for the cycle.

NOTE

A maximum of 10 layers of steps can be nested.

Step 7 When the configuration is complete, click **Save**.

----End

Condition Judgment

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired case in the case list on the left.

Step 4 On the **Case Script** tab, choose **Add More > Condition Controller**.

Step 5 Set parameters.

Judgment Condition: If a request meets the condition, the request, transaction, or other logic controller defined by the condition controller is executed. Both expressions and comparisons support variables and character strings. Currently, conditions cannot be combined. If you need to use a set or multiple comparisons, set a regular expression.

Step 6 Click **Add More** to add requests, transactions, or other logic controllers that are executed after conditions are met.

 **NOTE**

A maximum of 10 layers of steps can be nested.

Step 7 When the configuration is complete, click **Save**.

----End

Rendezvous Point

The purpose of a rendezvous point is to block concurrency until either X concurrent requests are blocked or the waiting time is reached, at which point all of them are released at a time. Therefore, a rendezvous point can create large instant loads at different points of the test case.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired case in the case list on the left.

Step 4 On the **Case Script** tab, choose **Add More > Rendezvous Point**.

Step 5 Set parameters.

Waiting Time (ms): waiting time at the rendezvous point.

Concurrent Number: When the number of users reaches this quantity, the users waiting at the rendezvous are released.

Step 6 When the configuration is complete, click **Save**.

----End

5.4.8 Adding Concurrency Weight Controllers and Throughput Controllers

Concurrency Weight Controller

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired case in the case list on the left.

Step 4 On the **Case Script** tab, choose **Add More > Concurrency Weight Controller**.

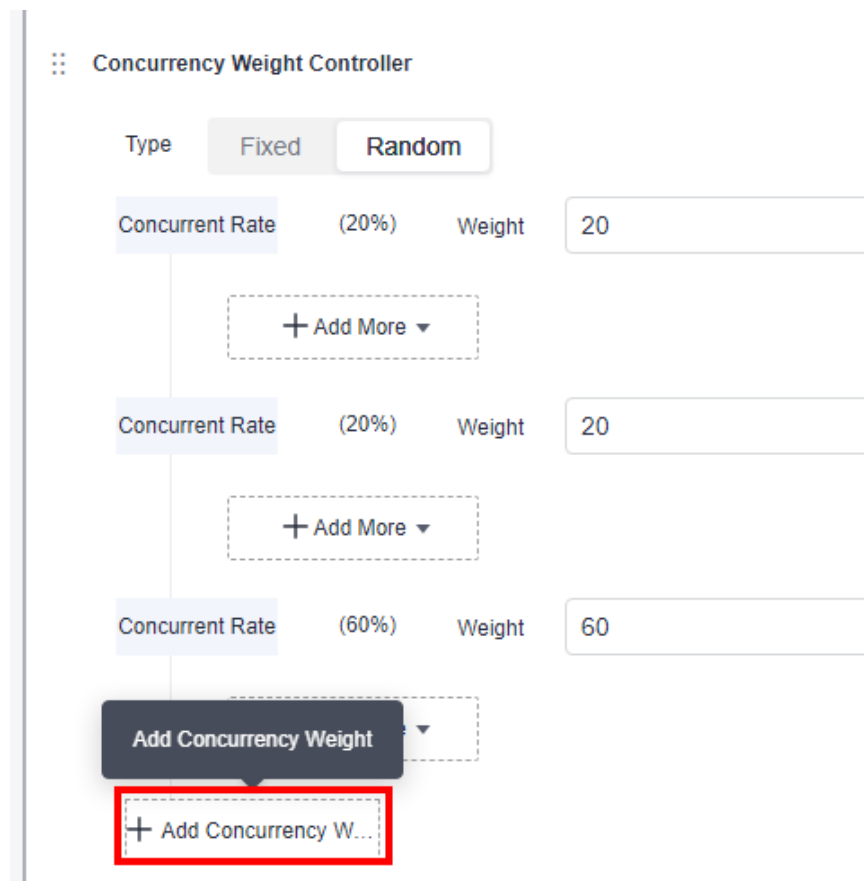
Step 5 Set parameters.

Type: The fixed branch and random branch modes are supported.

- **Fixed:** Randomly generates a fixed value for all concurrent executions. Subsequent executions use this fixed value to enter a fixed branch. This mode is used to group the concurrencies to simulate parallel tests in multiple scenarios in a case. Using multiple test cases for parallel testing in multiple low-concurrency scenarios requires many executors. You can group test scenarios in a case to save resources.
- **Random:** Generates a random value for each concurrent execution. Subsequent executions use this random value to enter a random branch. This mode is used to simulate the mixed ratio of the service interface RPS.

The concurrency weight is an integer greater than 0 and less than or equal to 10,000. If there is only one branch, the weight of the branch is 100%. You can click **Add Concurrency Weight** to add branches.

Figure 5-9 Concurrency weight controller



For example:

- **Fixed:** Assume that there are three low-concurrency test scenarios with 10, 20, and 30 concurrencies, respectively. The total number of concurrencies is 60.

Create a test case with 60 concurrencies, set the concurrency weight controller to **Fixed**, and set the weight ratio to A:B:C = 1:2:3. In this way, the three scenarios are combined into one test case.

- **Random:** If the peak RPS of a service system is 1,000 (200 on interface A and 800 on interface B), perform the test by selecting **Random** and setting the weight ratio to A:B = 2:8. In this way, the final ratio of the RPS on interface A to that on interface B is 2:8.

 **NOTE**

A maximum of 10 layers of steps can be nested.

Step 6 When the configuration is complete, click **Save**.

----End

Throughput Controller

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired case in the case list on the left.

Step 4 On the **Case Script** tab, choose **Add More > Throughput Controller**.

Step 5 Set parameters.

Set the RPS upper limit. The RPS executed by the controller cannot exceed the preset value. The RPS upper limit is **100** by default. The value can be up to 10 million and accurate to two decimal places.

 **NOTE**

A maximum of 10 layers of steps can be nested.

Step 6 When the configuration is complete, click **Save**.

----End

5.4.9 Pressure Configuration

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 Click **Pressure Configuration** and set phase parameters by referring to [Table 5-16](#).

 **NOTE**

- A maximum of 1000 phases can be added to a case.

Table 5-16 Adding a phase

Parameter	Description
Phase Name	Phases names are used to distinguish different phases. For example, you can set the name to Home page test .
Pressure Mode	<p>The following modes are supported:</p> <ul style="list-style-type: none">● Concurrency Mode: The performance test is performed based on the number of concurrent users. The number of concurrent users can be a fixed number or can be set to ramp up throughout the phase.● TPS Mode: The performance test is performed based on the number of transactions per second (TPS). In TPS mode, each execution of a test case by a virtual user is considered a transaction. The actual TPS may not reach your specified value because it will be affected by the specified maximum number of concurrent users and the average transaction latency.● Peakload Mode: The pressure test is performed based on the number of concurrent users who stays at the peak after increasing from the minimum concurrency to the maximum concurrency at a specified ramp-up time.● Surge Mode: The pressure test is performed based on the number of concurrent users who changes periodically. In each period, the minimum concurrency is tested first, then the maximum concurrency. The duration is based on the maximum concurrency and can be customized. The number of periods is the specified surge counts.● Oscillation Mode: The pressure test is performed based on the number of concurrent users who fluctuates periodically. The number of users in each period fluctuates between the minimum concurrency and the maximum concurrency. The number of periods is the specified oscillation counts.● Intelligent Peakload: The pressure test is performed based on the number of concurrent users who stays at the peak after increasing from the minimum concurrency to the maximum concurrency by a specified single-step execution duration. <p>NOTE</p> <p>The executor performs a pressure test by gradually increasing the number of concurrent users based on the specified gradient. During the pressure test, the background generates a phase data for each gradient. When the data in a phase has an obvious performance inflection point compared with that in the previous phase, the system determines that the peak performance is reached. The performance inflection point is defined in the following scenarios: The success rate is lower than the threshold (100% by default), the RPS curve decreases, and the rt/tp90/tp99 curve changes greatly in skew rate.</p>

Parameter	Description
Execution Policy	<p>The value can be Duration or Count.</p> <ul style="list-style-type: none"> • Duration: Tests will be performed based on the specified duration. • Count: Tests will be performed based on the total number of tests. <p>NOTE This parameter is involved only in the concurrency mode and TPS mode.</p>
Load Adjustment	<p>Auto and manual load adjustments are supported.</p> <ul style="list-style-type: none"> • Auto: The maximum number of concurrent users of all stages is used as the maximum concurrency. • Manual: The maximum number of concurrent users during a test is used as the maximum concurrency. After manual load adjustments, the original multi-stage pressure configuration becomes invalid. <p>NOTE This parameter is involved only in the concurrency mode and TPS mode, and Execution Policy is set to Duration.</p>
<p>When Pressure Mode is set to Concurrency Mode, Execution Policy is set to Duration, Load Adjustment is set to Auto, and Gradient Increment is disabled, set the following parameters.</p>	
Concurrent Users	<p>Concurrency refers to the number of users performing operations on a system at the same time.</p> <p>In CodeArts PerfTest, it is the number of virtual users you define when you configure a test phase.</p>
Duration (min)	Duration that the phase will last.
<p>When Pressure Mode is set to Concurrency Mode, Execution Policy is set to Duration, and Load Adjustment is set to Manual, set the following parameters.</p>	
Maximum Concurrency	Maximum number of virtual users for a test case.
Start Concurrency	Starting number of virtual users for a test case.
Duration (min)	Duration that the phase will last.
<p>When Pressure Mode is set to Concurrency Mode, Execution Policy is set to Duration, Load Adjustment is set to Auto, and Gradient Increment is enabled, set the following parameters:</p> <p>NOTE Each case can have only one phase whose Gradient Increment is enabled.</p>	
Start Concurrency	Starting number of concurrent users for a ramp-up.

Parameter	Description
Increasing Concurrency	Total number of concurrent users to be added. During a ramp-up, the starting number of concurrent users is simulated in the first subphase. The number of concurrent users in each subphase is the amount of (Concurrency in the previous phase + Increasing Concurrency x Increment). In the final subphase, the number of concurrent users will reach the amount of (Start Concurrency + Increasing Concurrency).
Increment	Number of concurrent users added in each subphase = Increasing Concurrency x Increment . The value can be 5%, 10%, 20%, or 50% .
Duration per Pressure Test (min)	Duration that each subphase will last.
When Pressure Mode is set to Concurrency Mode and Execution Policy is set to Count , set the following parameters: NOTE Only a single phase can be added.	
Concurrent Users	The number of concurrent users refers to the number of users performing operations on the system at the same time. In CodeArts PerfTest, it is the number of virtual users you define when you configure a test phase.
Total Sending Times	When the task is running, the performance test in this phase will be executed repeatedly until the number of execution times reaches the specified value. NOTE The total sending times must be greater than or equal to the number of concurrent users.
When Pressure Mode is set to TPS Mode , Execution Policy is set to Duration , and Load Adjustment is set to Auto , set the following parameters.	
Maximum Concurrency	Maximum number of virtual users for a test case.
TPS Value	Number of transactions executed per second.
Duration (min)	Duration that the phase will last.
When Pressure Mode is set to TPS Mode , Execution Policy is set to Duration , and Load Adjustment is set to Manual , set the following parameters.	
Maximum Concurrency	Maximum number of virtual users for a test case.
Start Concurrency	Starting number of virtual users for a test case.
TPS Value	Number of transactions executed per second.

Parameter	Description
Duration (min)	Duration that the phase will last.
When Pressure Mode is set to TPS Mode and Execution Policy is set to Count , set the following parameters:	
NOTE Only a single phase can be added.	
Maximum Concurrency	Maximum number of virtual users for a test case.
TPS Value	Number of transactions executed per second.
Total Sending Times	When the task is running, the performance test in this phase will be executed repeatedly until the number of execution times reaches the specified value. NOTE The total sending times must be greater than or equal to the number of concurrent users.
When Pressure Mode is set to Peakload Mode , set the following parameters.	
Start Concurrency	Starting number of virtual users for a test case.
Maximum Concurrency	Maximum number of virtual users for a test case.
Ramp Up(s)	Duration for which the number of virtual users increases from the start concurrency to the maximum concurrency. NOTE The recommended ramp-up time is shorter than the duration. If the ramp-up time is the same as the duration and the concurrency is large, the sampled maximum concurrency may be slightly less than the configured maximum concurrency.
Duration (min)	Duration that the phase will last.
Expected KPI	<ul style="list-style-type: none"> Maximum response time: 60,000 ms Minimum success rate: 0 <p>In peakload mode, the expected request response time of the tested system is less than or equal to the input value, and the request success rate is greater than or equal to the input value. If an indicator is unacceptable for six consecutive seconds for the first time (for example, the response time is greater than the expected value for six consecutive seconds), use the data in the first second as the analysis result.</p> NOTE When the execution of a test case reaches the expected level, the task continues.
When Pressure Mode is set to Surge Mode , set the following parameters.	
Duration (min)	Duration that the phase will last.

Parameter	Description
Maximum Concurrency	Maximum number of virtual users for executing a test case, which is also the peak value during the changing number of virtual users.
Minimum Concurrency	Minimum number of virtual users for executing a test case, which is also the minimum value during the fluctuation of the number of virtual users.
Surge Count	Number of periods in which the number of virtual users fluctuates within the duration.
Peak Duration(s)	Duration of a stage based on the maximum number of concurrent virtual users in a period.
When Pressure Mode is set to Oscillation Mode , set the following parameters.	
Duration (min)	Duration that the phase will last.
Maximum Concurrency	Maximum number of virtual users for executing a test case, which is also the peak value during the fluctuation of the number of virtual users.
Minimum Concurrency	Minimum number of virtual users for executing a test case, which is also the trough value during the fluctuation of the number of virtual users.
Oscillation Count	Number of periods in which the number of virtual users fluctuates within the duration.
When Pressure Mode is set to Intelligent Peakload , set the following parameters.	
Start Concurrency	Starting number of virtual users for a test case.
Increase Concurrency	Number of virtual users increased by step. NOTE <ul style="list-style-type: none"> • Increase Concurrency increases exponentially.
Single-step Execution Duration(s)	Duration of each incremental concurrency. You are advised to set this parameter to at least 20 seconds.
Duration (min)	Duration that the phase will last. You are advised to set this parameter to less than 30 minutes. NOTE <ul style="list-style-type: none"> • When the success rate is lower than the threshold (100% by default), the RPS curve decreases, or the rt/tp90/tp99 curve changes greatly in skew rate, the pressure test stops.
Success Rate Threshold	The tolerable success rate. The default value is 100 .

Step 5 When the configuration is complete, click **Save**.

----End

5.4.10 Advanced Configuration

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 Click **Advanced Configuration**.

1. **Executors:** The setting of the number of executors takes effect only when private resource groups are used. If this parameter is not set, the default policy is used. By default, concurrency supported by a single executor is 5,000 (HTTP/HTTPS), 5,000 (WebSocket), 5,000 (MQTT), 1,000 (JMeter), or 1,000 (HLS/RTMP/HTTP-FLV). Executors \geq Maximum number of concurrent users in all phases/5,000.
2. **Log Collection Policy:** If there are too many requests in your case or cyclic nesting exists, use the request mode.
 - In **Request Mode**, the requests in a case generate logs separately.
 - In **Case Mode**, all requests in a case are associated with each other before logs are generated.

Step 5 When the configuration is complete, click **Save**.

----End

5.4.11 SLA Configuration

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 Click **SLA Configuration**. For details, see [Configuring SLAs](#).

Step 5 When the configuration is complete, click **Save**.

----End

5.5 Configuring a Test Case (Scalable Test Projects)

5.5.1 Case Details

Case details describe the background, preconditions and test steps of a test case to record its basic information for reuse. Case details are optional.


NOTE

Extensible test projects are for whitelisted users. To use this feature, contact background support personnel to enable it and approve your request.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** Click **Case Details** on the right of the page. On the case details page that is displayed, enter the test case description, prerequisites, and procedure, and set basic case information based on section [Table 5-17](#).

Table 5-17 Basic case info

Parameter	Description
Status	Status of a test case, such as Creating , Designing , Testing , and Finished .
Case ID	ID of the test case.
Case Level	<ul style="list-style-type: none">• L0: verification of underlying functions. Each module should have 10 to 20 test cases. L0 test cases account for 5% of all test cases.• L1: basic function verification for inherited features or before iteration acceptance. L1 test cases account for 20% of all cases.• L2: important feature verification for manual tests in non-regression versions. L2 test cases account for 60% of all cases.• L3: verification of minor and non-important functions, and exception tests on basic and important functions. L3 test cases account for 10% to 15% of all cases.• L4: verification of special input, scenarios, and threshold conditions. L4 test cases account for less than 5% of all cases.
Label	<p>A custom parameter that is used to identify a case.</p> <ul style="list-style-type: none">• Enter a label: Click the label text box, enter a label, and press the space bar or Enter to generate a label. Each case can contain a maximum of 10 labels.• Delete a label: Click the label text box and click  next to a generated label to delete it.

Parameter	Description
Created at	Time when a test case was created. The value is automatically generated after the test case is saved.
Modified at	Time when a test case was last modified. The value is automatically generated after the test case is saved.

Step 5 After configuring the case details, click **Save**.

----End

5.5.2 Configuring a Case Script

After a test case is created, you can modify it as required.

Creating a Case Script

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 Click **Case Script** and select **Pre-Steps**, **Steps**, or **Post-steps**.

Pre-Steps are executed only once in each execution thread. This configuration is optional.

Steps are executor of a test case, which needs to be configured.

Post-Steps are executed only once in each execution thread. This configuration is optional.

Step 5 Create steps based on service requirements. For details, see [Add Step](#).

Step 6 When the configuration is complete, click **OK**.

----End

5.5.3 Add Step

Request Header

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **Header** under **Add Control**.

Step 6 Set Name and Value.

Add headers and contents based on the headers to be verified or used by a pressure test server. Headers are not mandatory. Only user-defined headers are transparently transmitted to the pressure test server. For details about **Header**, see [Header Description](#).

Step 7 When the configuration is complete, click **Save**.

----End

Comment

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **Comment** under **Add Control**.

Step 6 Enter the comment content. Comments are added to facilitate test case maintenance. Comments are not actually executed.

Step 7 When the configuration is complete, click **Save**.

----End

Sleeping Time

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **Sleep** under **Add Control**.

Step 6 Enter the sleeping time. You can add the sleeping time for subsequent interface invoking.

Step 7 When the configuration is complete, click **Save**.

----End

Group

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **Group** under **Add Control**.

Step 6 Enter a group name. If there are too many interfaces in a test case, you can use the grouping component to classify functions to facilitate subsequent maintenance. The grouping component is used only for classification maintenance.

Step 7 When the configuration is complete, click **Save**.

----End

Transactions

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **Transaction** under **Add Control**.

Step 6 Enter the transaction name.

Step 7 When the configuration is complete, click **Save**.

----End

if condition

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **if condition** under **Add Control**.

Step 6 Enter the judgment condition. Execute the corresponding branch based on the judgment result of the if condition statement.

Step 7 When the configuration is complete, click **Save**.

----End

Loop

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.


Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **Loop** under **Add Control**.

Step 6 Enter the number of loops.

Step 7 Click  on the right of **Loops** to set the loop interruption condition. When the test meets the interruption condition, the loop exits.

Step 8 When the configuration is complete, click **Save**.

----End

Fail

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **Fail** under **Add Control**.

Step 6 Set the conditions for failing the test in the if statement. When the conditions are met, the operation fails and subsequent interfaces will not be executed.

Step 7 When the configuration is complete, click **Save**.

----End

Percentage

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 On the **Case Script** tab, click **Steps**.

Step 5 Click **Fail** under **Add Control**.

Step 6 Set the concurrent weight percentage.

Step 7 When the configuration is complete, click **Save**.

----End

5.5.4 Pressure Configuration

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** Click **Pressure Configuration** and set phase parameters by referring to [Table 5-18](#).

 **NOTE**

- A maximum of 1000 phases can be added to a case.

Table 5-18 Adding a phase

Parameter	Description
Phase Name	Phases names are used to distinguish different phases. For example, you can set the name to Home page test .

Parameter	Description
Pressure Mode	<p>The following modes are supported:</p> <ul style="list-style-type: none">● Concurrency Mode: The performance test is performed based on the number of concurrent users. The number of concurrent users can be a fixed number or can be set to ramp up throughout the phase.● TPS Mode: The performance test is performed based on the number of transactions per second (TPS). In TPS mode, each execution of a test case by a virtual user is considered a transaction. The actual TPS may not reach your specified value because it will be affected by the specified maximum number of concurrent users and the average transaction latency.● Peakload Mode: The pressure test is performed based on the number of concurrent users who stays at the peak after increasing from the minimum concurrency to the maximum concurrency at a specified ramp-up time.● Surge Mode: The pressure test is performed based on the number of concurrent users who changes periodically. In each period, the minimum concurrency is tested first, then the maximum concurrency. The duration is based on the maximum concurrency and can be customized. The number of periods is the specified surge counts.● Oscillation Mode: The pressure test is performed based on the number of concurrent users who fluctuates periodically. The number of users in each period fluctuates between the minimum concurrency and the maximum concurrency. The number of periods is the specified oscillation counts.● Intelligent Peakload: The pressure test is performed based on the number of concurrent users who stays at the peak after increasing from the minimum concurrency to the maximum concurrency by a specified single-step execution duration. <p>NOTE The executor performs a pressure test by gradually increasing the number of concurrent users based on the specified gradient. During the pressure test, the background generates a phase data for each gradient. When the data in a phase has an obvious performance inflection point compared with that in the previous phase, the system determines that the peak performance is reached. The performance inflection point is defined in the following scenarios: The success rate is lower than the threshold (100% by default), the RPS curve decreases, and the rt/tp90/tp99 curve changes greatly in skew rate.</p>

Parameter	Description
Execution Policy	<p>The value can be Duration or Count.</p> <ul style="list-style-type: none"> • Duration: Tests will be performed based on the specified duration. The number of tests is not limited. • Count: Tests will be performed based on the total number of tests. <p>NOTE This parameter is involved only in the concurrency mode and TPS mode.</p>
Load Adjustment	<p>Auto and manual load adjustments are supported.</p> <ul style="list-style-type: none"> • Auto: The maximum number of concurrent users of all stages is used as the maximum concurrency. • Manual: The maximum number of concurrent users during a test is used as the maximum concurrency. After manual load adjustments, the original multi-stage pressure configuration becomes invalid. <p>NOTE This parameter is involved only in the concurrency mode and TPS mode, and Execution Policy is set to Duration.</p>
<p>When Pressure Mode is set to Concurrency Mode, Execution Policy is set to Duration, and Load Adjustment is set to Auto, set the following parameters.</p>	
Concurrent Users	<p>Concurrency refers to the number of users performing operations on a system at the same time.</p> <p>In CodeArts PerfTest, it is the number of virtual users you define when you configure a test phase.</p>
Start Concurrency	Starting number of virtual users for a test case.
Ramp Up(s)	Time required for increasing the start concurrency to the maximum concurrency.
Duration (min)	Duration that the phase will last.
<p>When Pressure Mode is set to Concurrency Mode, Execution Policy is set to Duration, and Load Adjustment is set to Manual, set the following parameters.</p>	
Maximum Concurrency	Maximum number of virtual users for a test case.
Start Concurrency	Starting number of virtual users for a test case.
Duration (min)	Duration that the phase will last.
<p>When Pressure Mode is set to Concurrency Mode and Execution Policy is set to Count, set the following parameters:</p> <p>NOTE Only a single phase can be added.</p>	

Parameter	Description
Concurrent Users	The number of concurrent users refers to the number of users performing operations on the system at the same time. In CodeArts PerfTest, it is the number of virtual users you define when you configure a test phase.
Total Sending Times	When the task is running, the performance test in this phase will be executed repeatedly until the number of execution times reaches the specified value. NOTE The total sending times must be greater than or equal to the number of concurrent users.
When Pressure Mode is set to TPS Mode , Execution Policy is set to Duration , and Load Adjustment is set to Auto , set the following parameters.	
Maximum Concurrency	Maximum number of virtual users for a test case.
TPS Value	Number of transactions executed per second.
Duration (min)	Duration that the phase will last.
When Pressure Mode is set to TPS Mode , Execution Policy is set to Duration , and Load Adjustment is set to Manual , set the following parameters.	
Maximum Concurrency	Maximum number of virtual users for a test case.
Start Concurrency	Starting number of virtual users for a test case.
TPS Value	Number of transactions executed per second.
Duration (min)	Duration that the phase will last.
When Pressure Mode is set to TPS Mode and Execution Policy is set to Count , set the following parameters: NOTE Only a single phase can be added.	
Maximum Concurrency	Maximum number of virtual users for a test case.
TPS Value	Number of transactions executed per second.
Total Sending Times	When the task is running, the performance test in this phase will be executed repeatedly until the number of execution times reaches the specified value. NOTE The total sending times must be greater than or equal to the number of concurrent users.
When Pressure Mode is set to Peakload Mode , set the following parameters.	
Start Concurrency	Starting number of virtual users for a test case.

Parameter	Description
Maximum Concurrency	Maximum number of virtual users for a test case.
Ramp Up(s)	Duration for which the number of virtual users increases from the start concurrency to the maximum concurrency. NOTE The recommended ramp-up time is shorter than the duration. If the ramp-up time is the same as the duration and the concurrency is large, the sampled maximum concurrency may be slightly less than the configured maximum concurrency.
Duration (min)	Duration that the phase will last.
Expected KPI	<ul style="list-style-type: none"> Maximum response time: 60,000 ms Minimum success rate: 0 <p>In peakload mode, the expected request response time of the tested system is less than or equal to the input value, and the request success rate is greater than or equal to the input value. If an indicator is unacceptable for six consecutive seconds for the first time (for example, the response time is greater than the expected value for six consecutive seconds), use the data in the first second as the analysis result.</p> NOTE When the execution of a test case reaches the expected level, the task continues.
When Pressure Mode is set to Surge Mode , set the following parameters.	
Duration (min)	Duration that the phase will last.
Maximum Concurrency	Maximum number of virtual users for executing a test case, which is also the peak value during the changing number of virtual users.
Minimum Concurrency	Minimum number of virtual users for executing a test case, which is also the minimum value during the fluctuation of the number of virtual users.
Surge Count	Number of periods in which the number of virtual users fluctuates within the duration.
Peak Duration(s)	Duration of a stage based on the maximum number of concurrent virtual users in a period.
When Pressure Mode is set to Oscillation Mode , set the following parameters.	
Duration (min)	Duration that the phase will last.
Maximum Concurrency	Maximum number of virtual users for executing a test case, which is also the peak value during the fluctuation of the number of virtual users.

Parameter	Description
Minimum Concurrency	Minimum number of virtual users for executing a test case, which is also the trough value during the fluctuation of the number of virtual users.
Oscillation Count	Number of periods in which the number of virtual users fluctuates within the duration.
When Pressure Mode is set to Intelligent Peakload , set the following parameters.	
Start Concurrency	Starting number of virtual users for a test case.
Increase Concurrency	Number of virtual users increased by step. NOTE <ul style="list-style-type: none">• Increase Concurrency increases exponentially.
Single-step Execution Duration(s)	Duration of each incremental concurrency. You are advised to set this parameter to at least 20 seconds.
Duration (min)	Duration that the phase will last. You are advised to set this parameter to less than 30 minutes. NOTE <ul style="list-style-type: none">• When the success rate is lower than the threshold (100% by default), the RPS curve decreases, or the rt/tp90/tp99 curve changes greatly in skew rate, the pressure test stops.

Step 5 When the configuration is complete, click **Save**.

----End

5.5.5 Advanced Configuration

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the desired test case in the case list on the left.

Step 4 Click **Advanced Configuration**.

1. **Executors:** The setting of the number of executors takes effect only when private resource groups are used. If this parameter is not set, the default policy is used. By default, concurrency supported by a single executor is 500. Executors \geq Maximum number of concurrent users in all phases/500.
2. **Whether to redirect:**
 - When this option is not selected, the target URL is not automatically redirected when an AW returns 302.

- When this option is selected, the 302 HTTP return code is returned, and the system continues to request the redirected URL.
 - 3. Select the log type. There are two log formats: ASON log and no log.
 - Intelligent log: displays the first 10 logs of each response code.
 - No log: No log is displayed.
 - 4. Select a request protocol.
- Step 5** When the configuration is complete, click **Save**.

----End

5.5.6 SLA Configuration

Procedure


- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** Click **SLA Configuration**. For details, see [Configuring SLAs](#).
- Step 5** When the configuration is complete, click **Save**.

----End

5.5.7 Action Word Library Management

Action Word Lib is used to manage APIs generated by importing files. After keywords are set for API files, you can quickly invoke APIs in test cases by searching for the keywords.


Importing an Interface File

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** Choose the **Cases** tab, and on the **Action Word Lib** page on the right, click **Import API** tab. You can also click  on the right of the **Keyword**.
- Step 4** In the displayed window, add the Swagger file and click **OK**.
- Step 5** After the API file is successfully imported, a directory named after the file is automatically generated in the **Keyword**.

----End

Adding a Keyword Directory Group

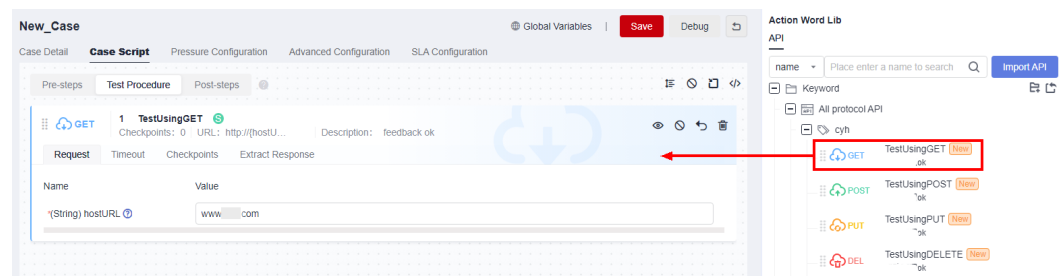
- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** Choose the **Cases** tab, and on the **Action Word Lib** page on the right, click on the right of the **Keyword**. 
- Step 4** Enter the directory name of the new group and click another position on the page. The directory name is automatically saved.
- End

Invoking the Keyword Directory API

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** Drag the API card to be invoked from **Action Word Lib** on the right to the **Test Procedure** of **Case Script**.

Figure 5-10 Invoking an API



- Step 5** Configure the test procedure in the invoked API template based on the service requirements.
- Step 6** When the configuration is complete, click **Save**.
- End

5.6 Setting Global Variables (High-Performance Test Projects)

5.6.1 Overview of Global Variables (High-Performance Test Projects)

Global variables are used to build data sets and enrich test data. If a global variable is used in a packet of a request, variable values in the packet will be replaced with specified values during a pressure test.


Global variables can be used in many scenarios. For example, you can store different usernames and passwords as global variables to simulate a scenario with multiple users.

5.6.2 Adding an Integer, Text, or Enumerated Global Variable

Procedure


- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, click **Global Variables**.
- Step 4** Add a global variable manually.
 1. Click **Add Variable**.
 2. Set parameters by referring to [Table 5-19](#).

Table 5-19 Setting global variables

Parameter	Description
Name	Name of a new global variable.
Type	Variable type, which can be integer, text, or enumerated.
Value	<p>A pressure test task reads the value range of the corresponding global variable when it is running. For each virtual user, the variable value is polled. That is, the first virtual user obtains the first value of the variable, and the second virtual user obtains the second value. If the values are exhausted, the next virtual user obtains the first value again.</p> <p>NOTE</p> <p>Variable content is transmitted in plain text during the pressure test. Exercise caution when entering the content.</p> <ul style="list-style-type: none">– When the variable type is Integer, enter the value range of the variable.– When the variable type is Enumerated, click  to enter a variable value.– When the variable type is Text, enter a text value.
Read Mode	<p>Sequential Mode: Variables are read in sequence.</p> <p>Random Mode: Variables are read randomly.</p>
Sharing Mode	<p>Case: All concurrent requests share a variable.</p> <p>Concurrency: Each request copies a variable with the same name for use. Different concurrent requests read the variable values in sequence and do not affect each other.</p>

Parameter	Description
Sensitive	If this option is selected, the variable value is encrypted for storage. Editing the variable after the page is refreshed will clear the variable value. Please exercise caution. NOTE Sensitive parameters apply only to text variables.

For details about the variable read rules for different variable read and sharing modes, see [Variable Read Rules](#).

3. When the settings are complete, click  to save them.
4. After creating a variable, you can perform the following operations:
 - Click **Edit** to modify it. If the global variable is referenced by a transaction, the value of the global variable in the transaction will be changed synchronously after the modification.
 - Click **Delete** to delete the added global variable file.

 **NOTE**

The global variable file cannot be deleted when it is referenced.

----End

5.6.3 Adding a .csv or .xlsx Global Variable File

If some parameters are dynamically obtained and the number of parameters is large, you can use .csv or .xlsx files to dynamically transfer the parameters during pressure tests.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, click **Global Variables**.
- Step 4** You can add global variables by importing file variables.
 1. Click **Download the .csv Template** or **Download the .xlsx Template** to obtain the required variable file template, and enter variables and values in the template.

 NOTE

In the templates, enter variable names in the first row and enter variable values from the second row.

The restrictions on variable files are as follows:

1. Only .csv (UTF-8 without BOM) and .xlsx files can be imported.
 2. The maximum length of a file name (including the extension) is 50 bytes.
 3. Maximum file size:
 - .xlsx: 10 MB for the Professional edition and 20 MB for the Enterprise edition.
 - .csv: 10 MB for the Professional edition and 80 MB for the Enterprise edition by default.
2. Click **Import File Variable** to import a .csv or .xlsx file.
 - **Name:** name of the file to be imported
 - **Type: File**
 - **Value:** actual variable name. Multiple variable names are separated by commas (,).
 - **Read Mode**
 - **Sequential Mode:** Variables are read in sequence.
 - **Random Mode:** Variables are read randomly.
 - **Sharing Mode**
 - **Case:** All concurrent requests share a variable.
 - **Concurrency:** Each request copies a variable with the same name for use. Different concurrent requests read the variable values in sequence and do not affect each other.
 3. After importing a global variable, you can perform the following operations:
 - Click **Edit** to modify the variable read mode and sharing mode.
For details about the variable read rules for different variable read and sharing modes, see [Variable Read Rules](#).
 - Click **Download** to download the global variable file.
 - Click **Delete** to delete the imported global variable file.

 NOTE

The global variable file cannot be deleted when it is referenced.

- If the imported global variable file is modified locally, click **Upload Again** to upload it again.

----End

5.6.4 Variable Read Rules

Table 5-20 lists the rules for reading manually specified or imported variables when different variable read modes and sharing modes are selected.

Table 5-20 Variable read rules

Variable Read Mode	Variable Sharing Mode	Description	Example
Sequential	Case	Each concurrency reads a variable value in sequence.	For example, for a task with a concurrency of 10, each concurrency reads a variable value in sequence.
Sequential	Concurrency	Each concurrency copies a variable with the same name for use. Different concurrent requests read the variable values in sequence and do not affect each other.	For example, for a task with a concurrency of 10, each concurrency reads the copied variable values in sequence.
Random	Case	Each concurrency randomly reads a variable value.	For example, for a task with a concurrency of 10, each concurrency randomly reads a variable value.
Random	Concurrency	Each concurrency copies a variable with the same name for use. Different concurrencies read the variable values randomly and do not affect each other.	For example, for a task with a concurrency of 10, each concurrency reads the copied variable values randomly.

Assume that the **number** variable has three values: **1**, **2**, and **3**, and there are two concurrencies: **A** and **B**.

- In sequential mode:
Case mode: **A** reads **1** of **number**, and **B** reads **2** of **number**. The process goes on cyclically.
Concurrency mode: **A** reads a copied variable **number** starting from **1**.
Concurrency mode: **B** also reads a copied variable **number** starting from **1**.
- In random mode:
The method of reading variable values is the same as that in sequential mode, regardless whether the **Case** or **Concurrency** sharing mode is used.

5.6.5 Inserting a Variable

When adding a request, enter **\$** in the text box to insert a variable.

When Type is set to Custom, set the following parameters:

- **Range:** variable range.

- **Name:** name of a custom variable. Specifies the name of a variable when the variable is added.

When Type is set to System, select a function name.

- **ID card number:** Randomly generates an ID number.
- **Mobile phone number:** Randomly generates a phone number.
- **Random number in a range:** Randomly generates an integer within the specified range.
- **Random character string:** Randomly generates a string consisting of lowercase letters and numbers based on your specified string length (1 to 32 characters).
- **Time stamp:** Generates a timestamp for the current time based on your specified unit. A 10-digit timestamp represents time in seconds whereas a 13-digit timestamp represents time in milliseconds.
- **Time stamp(Format):** The current time is output in the specified format. Time offsets in seconds are supported.

The time format is described as follows:

yyyy: indicates a four-digit year, for example, 2024.

yy: indicates a two-digit year. For example, 24 indicates 2024.

MM: indicates a two-digit month. For example, 01 indicates January.

M: indicates a month without leading zeros. For example, 1 indicates January.

dd: indicates a two-digit day, for example, 03.

d: indicates a day without leading zeros, for example, 3.

HH: indicates a 24-hour clock value. For example, 13 indicates 13:00.

H: same as **HH**.

hh: indicates a 12-hour clock value. For example, 01 indicates 01:00.

h: indicates a 24-hour clock value without leading zeros. For example, 1 indicates 01:00.

mm: indicates a minute, for example, 04.

m: indicates a minute without leading zeros, for example, 4.

ss: indicates a second, for example, 05.

s: indicates a second without leading zeros, for example, 5.

SSS: indicates a millisecond, for example, 098.

- **UUID:** randomly generates a string of 32 characters.

When the variable type is the encryption function, you need to select the encryption type and the position where the variable is inserted. The encryption types are as follows:

- **MD5:** Encrypts the val data using MD5 and returns the encrypted data string.
- **MD5-BASE64:** Performs MD5-BASE64 encoding on val data and returns the encoded data string.
- **SHA-1:** Encrypts the input data using SHA-1 and returns the encrypted data string.
- **SHA-224:** Encrypts the input data using SHA-224 and returns the encrypted data character string.

- SHA-256: Encrypts the input data using SHA-256 and returns the encrypted data character string.
- SHA-384: Encrypts the input data using SHA-384 and returns the encrypted data string.
- SHA-512: Encrypts the input data using SHA-512 and returns the encrypted data character string.
- BASE64 encoding: Encodes the input data using BASE64 and returns the encoded data string.
- BASE64 decoding: Performs BASE64 decoding on the input data and returns the decoded data character string.

5.7 Setting Global Variables (Extensible Test Projects)

5.7.1 Overview of Global Variables (Extensible Test Projects)

Global variables are used to build data sets and enrich test data. If a global variable is used in a packet of a request, variable values in the packet will be replaced with specified values during a pressure test.

Global variables can be used in many scenarios. For example, you can store different usernames and passwords as global variables to simulate a scenario with multiple users.

NOTE

Extensible test projects are for whitelisted users. To use this feature, contact background support personnel to enable it and approve your request.

Prerequisites


A test case has been created.

5.7.2 Creating and Deleting Global Variables

The global variables of extensible test projects are organized and managed in directories.


Creating a Variable

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.
- Step 4** You can create a variable in any of the following ways:
 - Click **New Variable**.
 - Right-click the blank area under the **No.** row and click **New Variable** from the displayed list.

- If a variable or variable folder exists, click  in the operation list where the variable or variable folder is located.

Step 5 Set parameters by referring to [Table 5-21](#).

Table 5-21 Setting global variables

Parameter	Description
Name	Name of a new global variable.
Type	The variable types such as String, Integer, Long, Double, Ciphertext, BigDecimal and Boolean are supported.
Value	When a pressure test task is running, the parameter values of the corresponding global variables are read. If the variable value is too long, click  on the right of the text box, and enter a value on the displayed page. Press Enter to increase the number of lines of the input value. After the input is complete, click Backfilling . If the input content is in JSON format, you can click JSON Conversion to view the input value.
Description	Description of the new variable.

Step 6 Click **Save**.

----End

Creating a Variable Folder

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

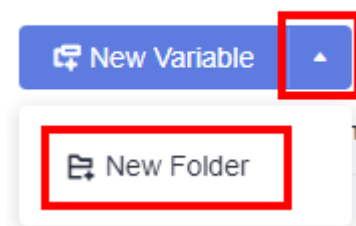
Step 3 On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.


Step 4 You can create a variable folder in any of the following ways:

- Click the drop-down button on the right of **New Variable** and click **New Folder**.

Figure 5-11 Creating a Folder

Global Variable



- Right-click the blank area under the **No.** row and click **New Folder** from the displayed list.
- If a variable or variable folder exists, click  in the operation list where the variable or variable folder is located.

Step 5 Enter the folder name and click **Save**.

----End

After a folder is created, you can create subfolders and subvariables in the folder. You can create subfolders and subvariables in either of the following ways:

- Click the icon for creating a subfolder or subvariable in the folder operation list.
- Right-click the row where the folder is located and choose **New Subfolder** or **New Subvariable**.

Deleting Variables and Variable Folders


Prerequisites: You have created variables and variable folders.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.

Step 4 You can delete a variable or variable folder in either of the following ways:

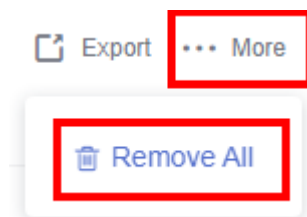
- Click  in the operation list of the variable or variable folder to be deleted.
- Right-click the row where the variable or variable folder to be deleted is located and click **Delete**.

Step 5 Click **Save**.

----End

To delete all variables and variable folders, choose **More > Remove All**. In the displayed dialog box, click **OK**.

Figure 5-12 Remove All



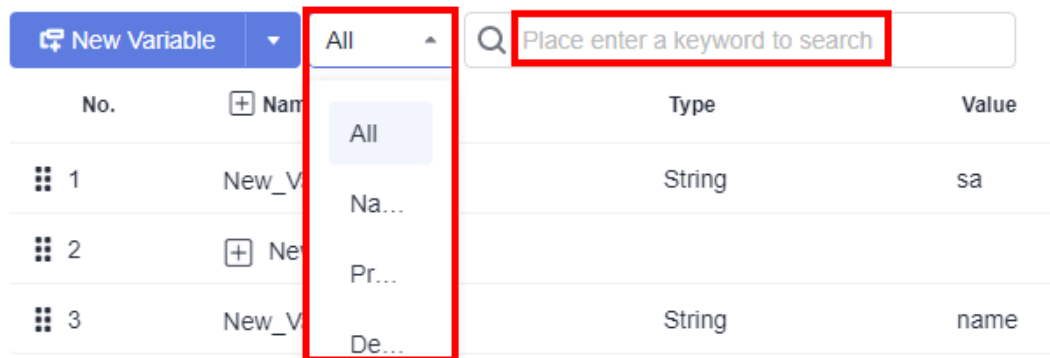
5.7.3 Global Variable Management

Directory Search

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.
- Step 4** Select a search type. You can search for global variables by name, variable value, description, or all.

Figure 5-13 Search

Global Variable








- Step 5** Enter a keyword in the search box and click the search icon on the left of the search box.

----End

Copying a Variable

Prerequisites: You have created a variable.

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.
- Step 4** You can copy variables in either of the following ways:
 - Click  in the operation list of the variable to be copied.
 - Right-click the row where the variable to be copied is located and click **Copy**.
- Step 5** After  is displayed in the operation list, click  to copy the variable to the root directory. If a variable folder has been created in the variable directory,  is

also displayed in the variable folder operation list. You can click  to copy the variable to the variable folder.

Step 6 Click **Save**.

----End

Copying a Variable Folder


Prerequisites: You have created a variable folder.





Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.

Step 4 You can copy a variable folder in either of the following ways:

- Click  in the operation list of the variable folder to be copied.
- Right-click the variable folder to be copied and click **Copy**.

Step 5 After   is displayed in the operation list, click  to copy the variable folder to the folder, and click  to copy the variable folder to the root directory.

Step 6 Click **Save**.

----End

Encrypting Variable Values

When the global variable type is Ciphertext (encryption type), the variable value must be encrypted. If the global variable type is changed to encryption, you need to manually update the test case that references the variable for the encryption variable to take effect.

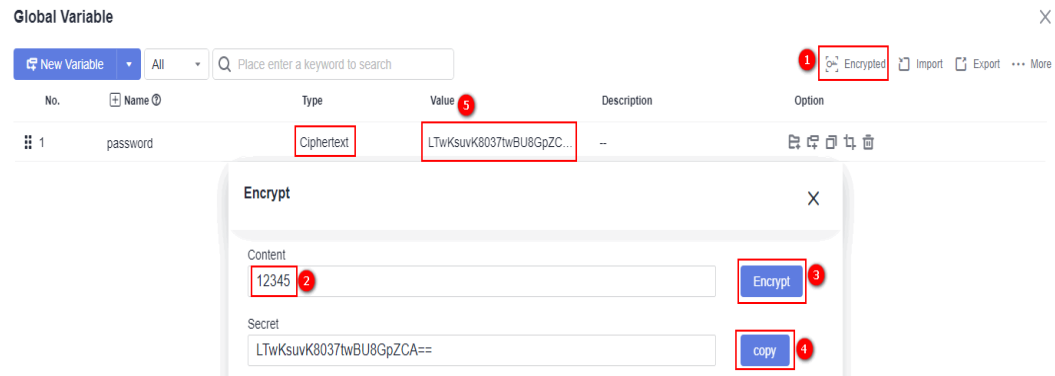
Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.

Step 4 Create a variable or select a created variable and set the variable type to Ciphertext.

Step 5 Click **Encrypted**. In the displayed dialog box, enter the content to be encrypted and click **Encrypt** to generate a key.

Figure 5-14 Variable encryption

Step 6 Click **Copy**. After the message **Copied to clipboard successfully** is displayed, click the variable value text box and paste the generated key to the variable value.

Step 7 Click **Save**.

----End

Importing Variables

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.

Step 4 Click **Import**. In the displayed dialog box, add the JSON file and click **OK**.

----End

Exporting Variables

Prerequisites: You have created a variable.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select the created test case in the case list on the left, and click **Global Variable**.

Step 4 Click **Export**. The created global variables are saved to the local host in JSON format.

----End

Inserting Global Variables

When adding request information, enter **\$\$** in the text box to display the global variable list and click the required global variable.

5.8 Binding a Domain Name

CodeArts PerfTest allows you to configure a Domain Name Service (DNS) address pool to map the domain name to the address. The domain name is defined in the URL and the IP address is automatically obtained through the DNS address pool.

NOTE


Domain names cannot be bound to extensible test projects.


Prerequisites

You can bind a domain name only when the **Request URL** on the **Packet** tab page contains the domain name.

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab page, click **Domain name binding**
- Step 4** Click **Adding a domain name**, and enter the domain name and IP address.



- Step 5** Click the enable button, enter an IP address, and click  .
----End

5.9 Plug-in Management

5.9.1 Summary

NOTE

Plug-ins are for whitelisted users. To use this feature, contact background support personnel to enable it and approve your request.

CodeArts PerfTest allows you to use the Go language to develop custom requests, functions, and checkpoints. Custom requests can be used to implement interconnection of protocols such as database protocols, MQTT, and proprietary protocols. Custom functions can be used to assist in generating test data, such as generating specific encryption and decryption data and generating data using

custom algorithms. Custom checkpoints are used for non-text, JSON, and other data serialization data or scenarios that are not supported by CodeArts PerfTest.

To implement custom requests, functions, and checkpoints, develop them based on [Signature Conventions](#), compile them into `so` files based on [Guidelines for Compiling Go Plug-ins](#), and use JSON files to describe them.

5.9.2 Signature Conventions

Request Signatures

```
func(map[string]string) interface{}
```

Input parameters: CodeArts PerfTest built-in parameters and custom parameters.

Output parameters: The `IResultV1` interface needs to be implemented. If necessary, use [IResultV1 Interface Implementation](#) provided by CodeArts PerfTest.

Table 5-22 CodeArts PerfTest built-in parameters

Name	Description	Remarks
<code>__name</code>	Name of a request	-
<code>__goroutine_id</code>	Coroutine ID	-
<code>__executor_index</code>	Executor index	The sequence number starts from 0.
<code>__executor_count</code>	Total number of executors	-

Function Signatures

```
func(map[string]string) string
```

Input parameters: customized parameters

Output parameters: character strings

Checkpoint Signatures

```
func(map[string]string) string
```

Input parameters: customized parameters.

Output parameters: character strings. An empty character string indicates that the check is successful, and a non-empty character string indicates the check fails.

IResultV1 Interface Implementation

```
package main

import (
    "fmt"
    "time"
```

```
)  
const (  
    InnerVarName      = "__name"  
    InnerVarGoroutineId = "__goroutine_id"  
    InnerVarExecutorIndex = "__executor_index"  
    InnerVarExecutorCount = "__executor_count"  
)  
  
type IResultV1 interface {  
    GetName() string  
    GetUrl() string  
    GetMethod() string  
    GetRequestHeader() map[string]string  
    GetRequestBody() string  
    GetSentBytes() int  
    GetResponseCode() int  
    GetResponseHeader() map[string]string  
    GetResponseBody() string  
    GetReceivedBytes() int  
    GetFailureMessage() string  
    IsSuccess() bool  
    GetBeginTime() int64  
    GetEndTime() int64  
    GetSubResults() []interface{}  
}  
  
//acquireResult generates root result.  
//just call one time on the main func and generate sub result using parent.addSub()  
func acquireResult(name string) *Result {  
    result := &Result{}  
    result.Name = name  
    result.RequestHeader = map[string]string{}  
    result.ResponseHeader = map[string]string{}  
    result.ResponseCode = 200  
    result.Success = true  
    result.BeginTime = time.Now().UnixMilli()  
    result.EndTime = time.Now().UnixMilli()  
    return result  
}  
  
type Result struct {  
    Name      string  
    Url       string  
    Method    string  
    RequestHeader map[string]string  
    RequestBody string  
    SentBytes  int  
    ResponseCode int  
    ResponseHeader map[string]string  
    ResponseBody string  
    ReceivedBytes int  
    FailureMessage string  
    Success      bool  
    BeginTime    int64  
    EndTime      int64  
    SubResults   []interface{}  
    SubIndex     int  
}  
  
func (r *Result) GetName() string {  
    return r.Name  
}  
  
func (r *Result) GetUrl() string {  
    return r.Url  
}  
  
func (r *Result) GetMethod() string {
```

```
    return r.Method
}

func (r *Result) GetRequestHeader() map[string]string {
    return r.RequestHeader
}

func (r *Result) GetRequestBody() string {
    return r.RequestBody
}

func (r *Result) GetSentBytes() int {
    return r.SentBytes
}

func (r *Result) GetResponseCode() int {
    return r.ResponseCode
}

func (r *Result) GetResponseHeader() map[string]string {
    return r.ResponseHeader
}

func (r *Result) GetResponseBody() string {
    return r.ResponseBody
}

func (r *Result) GetReceivedBytes() int {
    return r.ReceivedBytes
}

func (r *Result) GetFailureMessage() string {
    return r.FailureMessage
}

func (r *Result) IsSuccess() bool {
    return r.Success
}

func (r *Result) GetBeginTime() int64 {
    return r.BeginTime
}

func (r *Result) GetEndTime() int64 {
    return r.EndTime
}

func (r *Result) GetSubResults() []interface{} {
    return r.SubResults
}

//begin records begin time, do not forget call this function to update
func (r *Result) begin() {
    r.BeginTime = time.Now().UnixMilli()
}

//end records end time, do not forget call this function to update
func (r *Result) end() {
    r.EndTime = time.Now().UnixMilli()
}

//addSub adds sub result to parent, call this function adding sub result always.
//if name is not empty, renaming will be disabled
func (r *Result) addSub(name string) *Result {
    if name == "" {
        name = fmt.Sprintf("%s-%d", r.Name, r.SubIndex)
        r.SubIndex++
    } else {
        name = fmt.Sprintf("%s-%s", r.Name, name)
    }
}
```

```
}  
sub := acquireResult(name)  
r.SubResults = append(r.SubResults, sub)  
return sub  
}
```

Table 5-23 Structure of the IResultV1 interface

Name	Description	Remarks
Name	Name of a request	The root result must be set using the built-in parameter __name .
Url	Request URL	-
Method	Method	Used for HTTP POST and GET.
RequestHeader	Request header	Used for HTTP
RequestBody	Request data	Do not record data with a large request size, for example, uploading a file. You only need to record the abstract.
SentBytes	Number of bytes sent	-
ResponseCode	Response code	Response status, which can be used for HTTP status codes or custom status codes. It is used to report the number of response statuses for analysis. Response code range: 100–599
ResponseHeader	Response header	Used for HTTP.
ResponseBody	Response data	Do not record data with a large response, for example, downloading a file. You only need to record the abstract.
ReceivedBytes	Bytes received	-
FailureMessage	Failure message	-
Success	Whether the request is successful	-
BeginTime	Request start time	Unit: ms
EndTime	Request end time	Unit: ms

Name	Description	Remarks
SubResults	Sub-request	When a custom request has multiple sub-requests, this field is used to record the execution status of each sub-request.
SubIndex	Sub-request index	When the addSub method of the parent result is used to generate a sub-result, and the sub-result name is not customized, this field is used to generate an index for generating the sub-result name.

5.9.3 Guidelines for Compiling Go Plug-ins

Constraint

The Go language has many restrictions on plug-ins, such as the Go version, third-party dependency version, and compilation parameters. The plug-ins must be strictly consistent with the main program of the CodeArts PerfTest executor. Otherwise, exceptions including but not limited to those described in [Table 5-24](#) may occur. Compile the `so` file strictly according to the conventions.

Table 5-24 Guidelines for compiling Go plug-ins

Item	Constraint Value	Description
arch	linux amd64	-

Item	Constraint Value	Description
Go version	1.19.6	<p>Run the go version command to check the version.</p> <ul style="list-style-type: none">• The following exception occurs when medium versions are inconsistent: fatal error: runtime: no plugin module data• The following exception occurs when minor versions are inconsistent: plugin was built with a different version of package runtime
go.mod	github.com/go-resty/ resty/v2 v2.7.0 github.com/shopspring/ decimal v1.3.1 github.com/xuri/excelize/v2 v2.6.0 github.com/tidwall/gjson v1.14.4 github.com/valyala/fasthttp v1.43.0 go.uber.org/atomic v1.10.0 golang.org/x/net v0.8.0 golang.org/x/time v0.2.0	<p>If a third-party library is directly or indirectly used, use the version according to the constraint value in the go.mod file.</p> <p>If the third-party versions are inconsistent, the following exception occurs: plugin was built with a different version of package github.com/json-iterator/go</p>
TrimPath	Using TrimPath	<p>If this parameter is not used, the following exception occurs: plugin was built with a different version of package internal/unsafeheader</p> <p>Run the Linux command strip to remove symbols and debugging information.</p>

Compilation Example

```
go build -buildmode=plugin -trimpath -o random.so random.go
strip random.so
```

5.9.4 JSON Description Statement

Example:

```
{
  "name": "random",
  "plugin": "random.so",
  "desc": "",
  "funcs": [{
    "name": "RandomNum",
    "type": "func",
    "desc": "",
    "params": [{
      "name": "min",
      "default": "0",
      "range": [],
      "desc": "this is a min",
      "required": true
    }],
    {
      "name": "max",
      "default": "10",
      "range": [],
      "desc": "this is a max",
      "required": true
    }
  ]
}
```

Table 5-25 JSON example description

Name	Description	Remarks
name	Key alias	Aliases are used for grouping.
plugin	Add-on name	Same as the name of the uploaded plug-in.
desc	Plug-in description	-
funcs	Function list	See Table 5-26 .

Table 5-26 Function table

Name	Description	Remarks
name	Function name	Must be exportable.
desc	Function description	-
type	Function type	req: request func: function check: checkpoint

Name	Description	Remarks
params	Parameters	See Table 5-27 .

Table 5-27 Parameters

Name	Description	Remarks
name	Parameter name	Key to be used as an input parameter.
default	Default value	-
range	Value range	Example: ["on","off"]
desc	Description	-
required	Mandatory	-

5.9.5 Custom Examples

- [Custom Request Example](#)
- [Example of a Custom Function](#)

Custom Request Example

Example of a custom Go request:

```
package main

import (
    "crypto/tls"
    "encoding/json"
    "errors"
    "fmt"
    "io/ioutil"
    "net/http"
    "net/http/cookiejar"
    "strings"
    "sync"
)

var (
    trs = sync.Map{}
    jars = sync.Map{}
)

func getTr(addr string) *http.Transport {
    if v, ok := trs.Load(addr); ok {
        return v.(*http.Transport)
    } else {
        tr := &http.Transport{
            TLSClientConfig: &tls.Config{
                InsecureSkipVerify: true,
            },
            MaxConnsPerHost: 5000,
            MaxIdleConnsPerHost: 5000,
        }
    }
}
```

```
v, _ = trs.LoadOrStore(addr, tr)
return v.(*http.Transport)
}
}

func getJar(goroutineId string) *cookiejar.Jar {
    if v, ok := jars.Load(goroutineId); ok {
        return v.(*cookiejar.Jar)
    } else {
        jar, _ := cookiejar.New(nil)
        v, _ = jars.LoadOrStore(goroutineId, jar)
        return v.(*cookiejar.Jar)
    }
}

func outgoingLength(req *http.Request) int64 {
    if req.Body == nil || req.Body == http.NoBody {
        return 0
    }
    if req.ContentLength != 0 {
        return req.ContentLength
    }
    return -1
}

func redirectBehavior(reqMethod string, resp *http.Response, req *http.Request) (redirectMethod string,
shouldRedirect, includeBody bool) {
    switch resp.StatusCode {
    case 301, 302, 303:
        redirectMethod = reqMethod
        shouldRedirect = true
        includeBody = false

        if reqMethod != "GET" && reqMethod != "HEAD" {
            redirectMethod = "GET"
        }
    case 307, 308:
        redirectMethod = reqMethod
        shouldRedirect = true
        includeBody = true

        if resp.Header.Get("Location") == "" {
            shouldRedirect = false
            break
        }
        if req.GetBody == nil && outgoingLength(req) != 0 {
            shouldRedirect = false
        }
    }
    return redirectMethod, shouldRedirect, includeBody
}

func Do(params map[string]string) interface{} {
    parent := acquireResult(params[InnerVarName])
    sub := parent
    var (
        err      error
        redirectCount int
        req      *http.Request
        res      *http.Response
    )
    defer func() {
        if p := recover(); p != nil {
            sub.ResponseCode = 500
            sub.Success = false
            sub.FailureMessage = fmt.Sprintf("%v", sub)
        } else if err != nil {
            sub.ResponseCode = 500
            sub.Success = false
        }
    }
}
```

```
        sub.FailureMessage = err.Error()
    }
}()

redirect := params["redirect"]
for {
    // process params
    url, method, body := params["url"], params["method"], params["body"]
    sub.Url, sub.Method, sub.RequestBody = url, method, body
    sub.SentBytes = len(body)

    // create request
    req, err = http.NewRequest(method, url, strings.NewReader(body))
    if err != nil {
        return parent
    }

    // load cookies
    jar := getJar(params[InnerVarGoroutineId])
    cookies := jar.Cookies(req.URL)
    for _, c := range cookies {
        req.AddCookie(c)
    }

    // write headers
    if params["headers"] != "" {
        headers := map[string][]string{}
        if err = json.Unmarshal([]byte(params["headers"]), &headers); err != nil {
            return parent
        }
        for k, vs := range headers {
            for _, v := range vs {
                req.Header.Add(k, v)
            }
        }
    }

    for k, v := range req.Header {
        sub.RequestHeader[k] = strings.Join(v, ",")
    }

    // do request
    sub.begin()
    res, err = getTr(fmt.Sprintf("%s://%s", req.URL.Scheme, req.URL.Host)).RoundTrip(req)
    sub.end()
    if err != nil {
        return parent
    }
    var b []byte
    b, err = ioutil.ReadAll(res.Body)
    if err != nil {
        _ = res.Body.Close()
        return parent
    }

    // store cookies
    jar.SetCookies(req.URL, res.Cookies())

    for k, v := range res.Header {
        sub.ResponseHeader[k] = strings.Join(v, ",")
    }

    sub.ResponseBody = string(b)
    sub.ReceivedBytes = len(b)
    sub.ResponseCode = res.StatusCode
    _ = res.Body.Close()
    if redirect != "on" {
        break
    }
}
```

```
if redirectMethod, shouldRedirect, includeBody := redirectBehavior(method, res, req); shouldRedirect {
    redirectCount++
    if redirectCount > 10 {
        err = errors.New("stopped after 10 redirects")
        return parent
    }
    params["url"] = res.Header.Get("Location")
    params["method"] = redirectMethod
    if !includeBody {
        params["body"] = ""
    }
    parent = acquireResult(params[InnerVarName])
    sub = parent
    continue
}
break
}
return parent
}
```

Example of a custom request in JSON format:

```
{
  "name": "Http 1.0",
  "plugin": "http.so",
  "desc": "Http plugin supports http 1.0 test.",
  "funcs": [
    {
      "name": "Do",
      "type": "req",
      "desc": "This is the entry of http request.",
      "params": [
        {
          "name": "url",
          "default": "",
          "range": [],
          "desc": "The url of http request",
          "required": true
        },
        {
          "name": "headers",
          "default": "{\"Content-Type\": [\"text/plain;utf-8\"]}",
          "range": [],
          "desc": "The headers of http request",
          "required": true
        },
        {
          "name": "method",
          "default": "GET",
          "range": [
            "GET",
            "HEAD",
            "POST",
            "PUT",
            "PATCH",
            "DELETE",
            "CONNECT",
            "OPTIONS",
            "TRACE"
          ],
          "desc": "The method of http request",
          "required": true
        },
        {
          "name": "body",
          "default": "",
          "range": [],
          "desc": "The body of http request",
          "required": true
        }
      ]
    }
  ]
}
```

```
{
  "name": "redirect",
  "default": "on",
  "range": [
    "on",
    "off"
  ],
  "desc": "If enable auto redirect",
  "required": true
}
]
```

Example of a Custom Function

Example of a custom Go function:

```
package main

import (
    "fmt"
    "math/rand"
    "strconv"
)

func RandomNum(params map[string]string) (s string) {
    var err error
    defer func() {
        if r := recover(); r != nil {
            s = fmt.Sprintf("%v", r)
        } else if err != nil {
            s = err.Error()
        }
    }()

    min, err := strconv.ParseInt(params["min"], 10, 64)
    if err != nil {
        return
    }
    max, err := strconv.ParseInt(params["max"], 10, 64)
    if err != nil {
        return
    }
    if min > max {
        min, max = max, min
    }
    return strconv.FormatInt(min+rand.Int63n(max-min+1), 10)
}

var (
    charSet = []rune("abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789")
)

func RandomText(params map[string]string) (s string) {
    var err error
    defer func() {
        if r := recover(); r != nil {
            s = fmt.Sprintf("%v", r)
        } else if err != nil {
            s = err.Error()
        }
    }()

    length, err := strconv.ParseInt(params["length"], 10, 64)
    if err != nil {
        return
    }
}
```



```
if length == 0 {
    return ""
}
set := []rune(params["charset"])
if len(set) == 0 {
    set = charset
}
n := len(set)
rs := make([]rune, length)
for i := int64(0); i < length; i++ {
    rs[i] = set[rand.Intn(n)]
}
return string(rs)
}
```

Example of a custom JSON function:

```
{
  "name": "Random",
  "plugin": "random.so",
  "desc": "Random plugin supports generating random value.",
  "funcs": [
    {
      "name": "RandomNum",
      "type": "func",
      "desc": "RandomNum generates random number.",
      "params": [
        {
          "name": "min",
          "default": "0",
          "range": [],
          "desc": "The maximum of random number.",
          "required": true
        },
        {
          "name": "max",
          "default": "100",
          "range": [],
          "desc": "The maximum of random number.",
          "required": true
        }
      ]
    },
    {
      "name": "RandomText",
      "type": "func",
      "desc": "RandomText generates random text of a specified length.",
      "params": [
        {
          "name": "length",
          "default": "32",
          "range": [],
          "desc": "The length of random text.",
          "required": true
        },
        {
          "name": "charset",
          "default": "",
          "range": [],
          "desc": "Random text is generated randomly from the specified character set. if empty, default char set is 0-9 a-z A-Z",
          "required": false
        }
      ]
    }
  ]
}
```

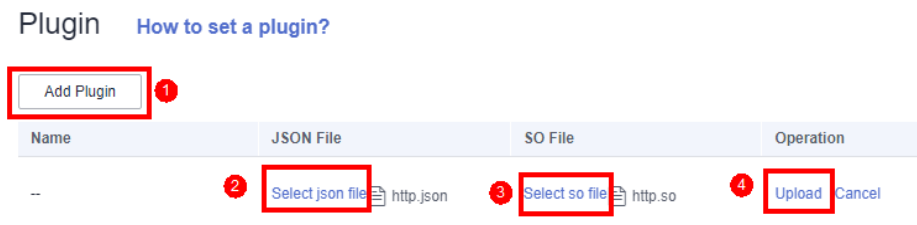
5.9.6 Plug-in Usage

After compiling the **so** and **JSON** files, upload the files to the PerfTest project and add custom requests and functions to the test cases.

Adding a Plug-in

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, click **Plugin**.
- Step 4** In the displayed window, click **Add Plugin**. Click **Select json file** and **Select so file**. Then click **Upload**.

Figure 5-15 Adding a plug-in



NOTE

1. The **JSON** file size cannot exceed 50 KB, and the file name length cannot exceed 64 characters.
2. Restrictions on **so** files: The file size cannot exceed 80 MB, and the number of files cannot exceed 20.

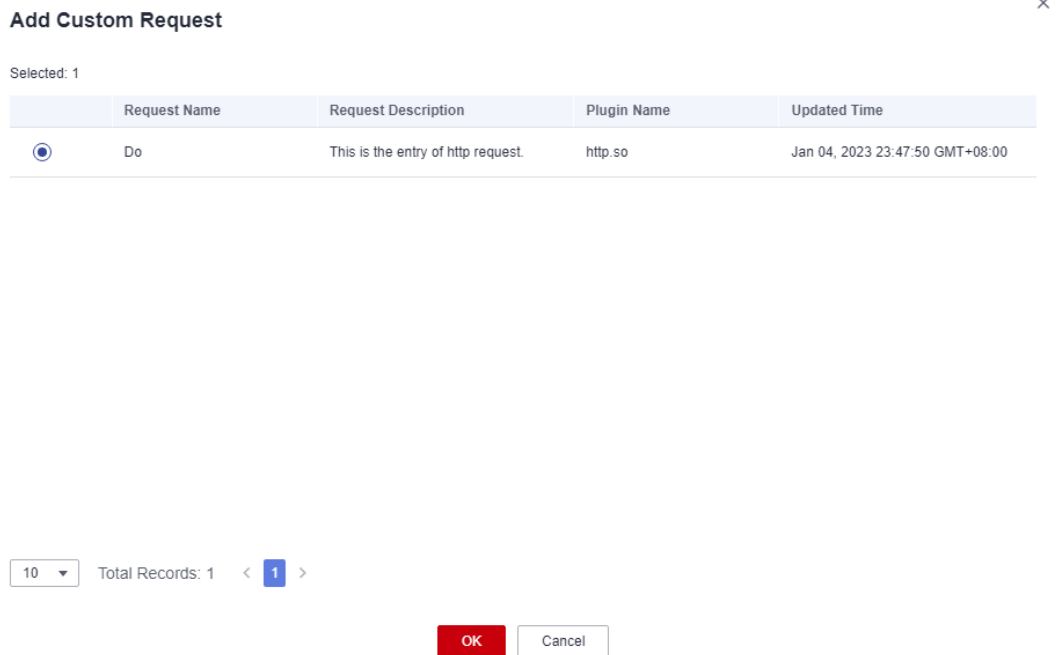
----End

Adding a Custom Request

Ensure that you have added the custom request plug-in.

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** On the **Case Script** tab, choose **Add Custom Request**.
- Step 5** In the window that is displayed, select a request.

Figure 5-16 Adding a custom request



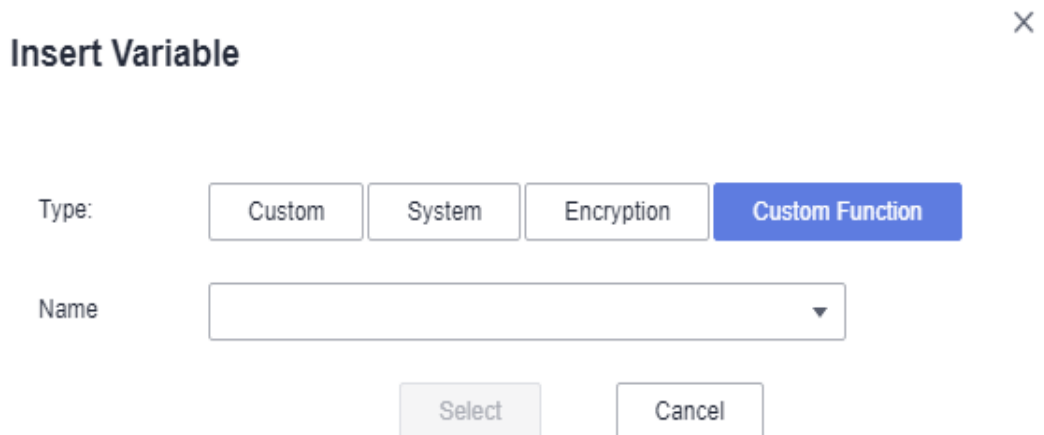
Step 6 Click **OK**.

----End

Adding a Custom Function

After adding a custom function plug-in, enter \$ in the text box and set variable type to **Custom Function**. Select the name of the imported function, and click **Select**.


Figure 5-17 Adding a custom function



Adding a Custom Checkpoint

Ensure that you have added the custom checkpoint plug-in.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

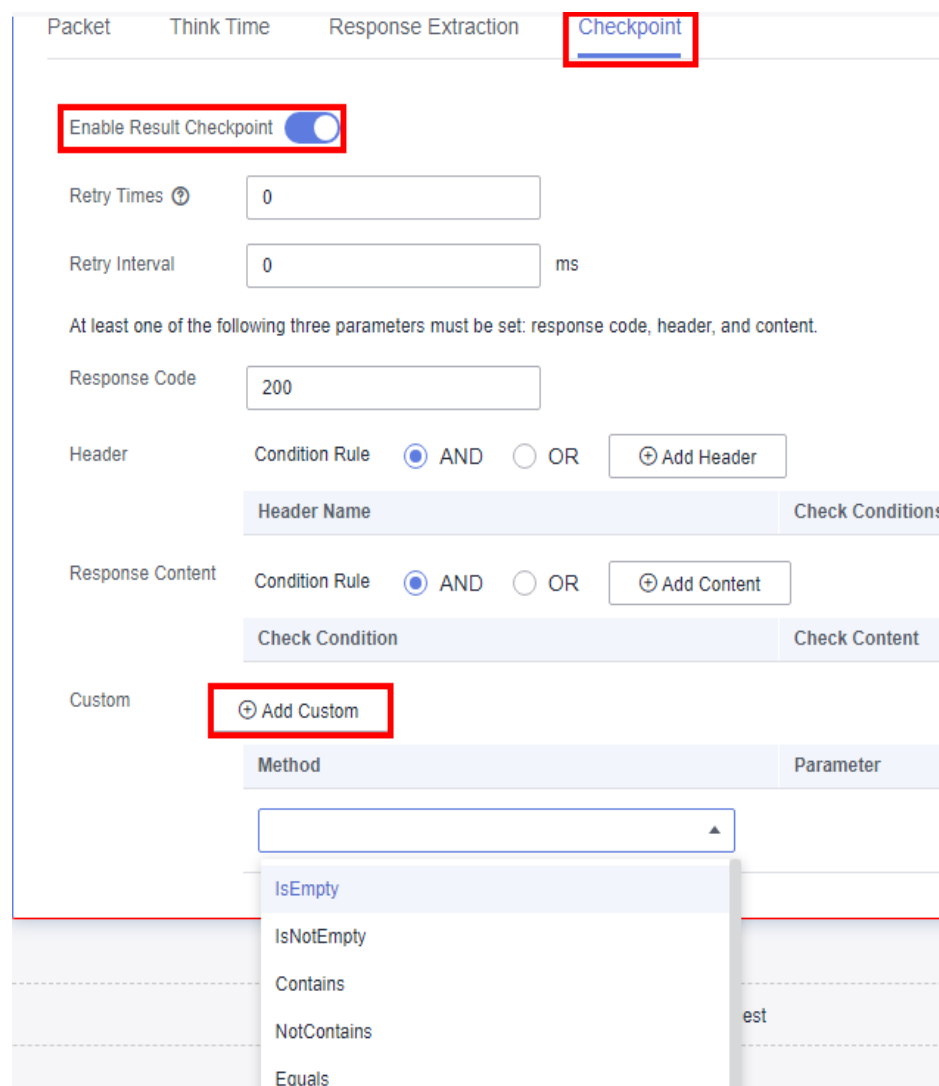
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.
- Step 4** On the **Case Script** tab, select the request for which a custom checkpoint is to be added and click  in the operation column.
- Step 5** On the **Checkpoint** tab, enable result check and set the response code, header, or response content.

 **NOTE**

When enabling the checkpoint, set at least one of the following three parameters: response code, header, and response content.

- Step 6** Click **Add Custom**. In the displayed dialog box, select the imported checkpoints.

Figure 5-18 Adding a custom checkpoint



Packet Think Time Response Extraction **Checkpoint**

Enable Result Checkpoint

Retry Times

Retry Interval ms

At least one of the following three parameters must be set: response code, header, and content.

Response Code

Header Condition Rule AND OR

Header Name Check Conditions

Response Content Condition Rule AND OR

Check Condition Check Content

Custom

Method Parameter

- IsEmpty
- IsNotEmpty
- Contains
- NotContains
- Equals


----End

5.10 Resetting Configurations

When you modify a test case, you can reset the configurations that are not saved.

Procedure


- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the desired test case in the case list on the left.

- Step 4** Click  in the upper right corner of the page. In the dialog box that is displayed, click **OK**.

----End

5.11 Deleting a Test Case

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, locate the case to be deleted in the case list on the left.
- Step 4** Click  next to the case to be deleted, and choose **Delete** from the drop-down list.
- Step 5** Click **OK**.

----End

5.12 Debugging a Case

Prerequisites

- Test cases have been added.
- The resource group is in the **Running** state.
- Ports 32001 and 32003 on the debugging node of the resource group are enabled in the security group.
- The execution node of the resource group and applications under the pressure test are connected through the network.

Debugging a Test Case

After you create or edit a test case, you can debug it to detect syntax or configuration errors before testing.

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select the test case to be debugged in the case list on the left.
- Step 4** Click **Debug** in the upper right corner of the page.
- Step 5** Select the target test resource group as the executor and click **Start**.
- Step 6** Click **View log** to view the test case debugging details.

If an error was reported in the debugging result, edit the case based on the log information and debug it again.

- Step 7** On the **Debug log** tab page, you can view the debugging history.

----End

5.13 Batch Operations

The case list supports batch operations. Batch operations apply only to cases (including cases in the directory but excluding the directory itself).

Batch Starting High-Performance Test Cases

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Cases** tab, select multiple cases or directories in the case list on the left.
- Step 4** In the floating box displayed in the lower part of the page, click **Start**.
- Step 5** In the displayed **Executing Temporary Test Task** dialog box, modify the parameters of the temporary task.

Table 5-28 Temporary test tasks

Parameter	Description
Temporary Task Name	The current case will not generate a test task, and the generated test report will be displayed in the performance report as a temporary task report and automatically deleted 30 days later.
Execution Mode	Serial or parallel

Parameter	Description
Enterprise	Provides a cloud resource management mode where cloud resources and members are centrally managed by project.
Resource Group Type	Shared resource group or private resource group

Step 6 Click **Start** to execute the temporary task.

Step 7 After the test case is started, you can click **View Report** to view the real-time performance report.

You can also switch the tab page in the upper left corner and choose **Performance Reports > Temporary Task List** to view the performance report by temporary task name.

----End

Batch Debugging Extensible Test Cases

NOTE

Scalable test projects are for whitelisted users. To use this feature, contact background support personnel to enable it and approve your request.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select multiple cases or directories in the case list on the left.

Step 4 In the floating box displayed in the lower part of the page or in the upper right corner of the page, click **Debug**.

Step 5 In the displayed dialog box, select a test resource group as the executor.

Step 6 Click **Start**.

----End

Batch Deleting Test Cases

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Cases** tab, select multiple cases or directories in the case list on the left.

Step 4 In the floating box displayed in the lower part of the page, click **Delete**.

Step 5 Click **OK**.

----End

6 PerfTest Task Management

6.1 Creating a Test Task

A test task initiates a performance test based on a defined test model. It is a series of tests performed at different pressure points to continuously initiate pressure tests on a system. Testing allows you to obtain and analyze the performance data of the system. You can add multiple test tasks to a test project.

Prerequisites

- A test case has been created and configured based on service requirements.
- (Optional) A transaction has been added.

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Test Tasks** tab, click **Create Task**.

Step 4 Enter a task name and select an execution policy.

There are two execution policies:

- **Serial:** Test cases in the test task are executed in sequence.
- **Parallel:** Test cases in the test task are executed concurrently.

Step 5 Click **Add Case**. In the dialog box that is displayed, select created cases.

NOTE

A test case can be added to multiple test tasks.

Step 6 When the configuration is complete, click **Save**.

----End

6.2 Starting a Test Task


Prerequisites

- Test cases have been added.
- The resource group is in the **Running** state.
- Ports 32001 and 32003 on the debugging node of the resource group are enabled in the security group.
- The execution node of the resource group and applications under the pressure test are connected through the network.

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 Click the **Test Tasks** tab, and then click  in the **Operation** column of the desired task.

Step 4 In the **Executing Test Task** dialog box, specify **Enterprise** (not required if an individual account is used.)


Enterprise projects let you manage cloud resources and users by project.

For example, if you select EPS1, CDRs will be generated for EPS1.

1. If enterprise project EPS1 has an available package, VUMs in the package are deducted.
2. If enterprise project EPS1 does not have any package or the remaining package capacity is insufficient, VUMs are deducted from the account's packages that do not belong to the enterprise project, if any. If no such packages exist or the remaining package capacity is insufficient, you will be billed on a pay-per-use basis.

Step 5 In the **Executing Test Task** dialog box, select a resource group type from the **Resource Group Type** drop-down list.

Step 6 Click **Start** to start the test task.

Step 7 (Optional) After the test task is executed, click **View Report** or  in the **Operation** column of the test task to view the real-time test report.

----End

6.3 Managing Test Tasks

After a test task is created, you can manage it and the task phases.

Starting Test Tasks in Batches


Start multiple test tasks under the same test project.

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Test Tasks** tab page, select multiple tasks and click **Start**.
- Step 4** Select a resource group and enterprise project, and click **Start**.

----End


Stopping Test Tasks

Stop multiple test tasks under the same test project.

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** On the **Test Tasks** tab, click  in the row containing the desired test task. You can also select multiple tasks and click **Stop** to stop them at a time.

----End

Editing a Test Task

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Click the name of the desired PerfTest project to go to the details page.
- Step 3** Click the **Test Tasks** tab, and then click  in the **Operation** column of the desired task.
- Step 4** Edit the test task name, execution policy, and cases as required.
 - Click the original task name to display a text box. Enter a new task name in the box.
 - Modify the execution policy.
 - Delete or add test cases.

- Step 5** Click **Save**.

----End

Deleting a Test Task

Deleted test tasks cannot be recovered. Exercise caution when deleting a task.

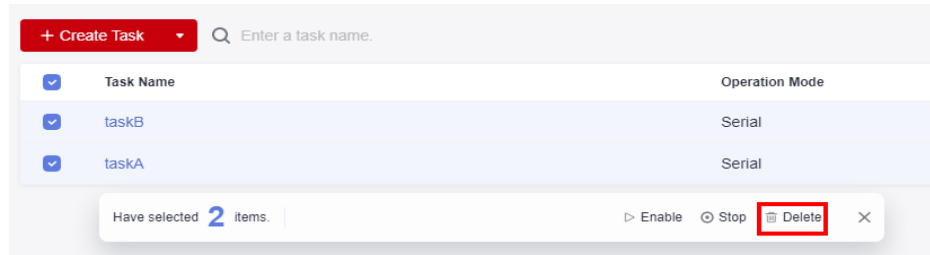
- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Test Tasks** tab, click **...** in the row containing the desired test task, and click **Delete**.

You can also select one or more test tasks and click **Delete**.

Figure 6-1 Deleting test tasks



Step 4 Delete the test tasks as prompted.

----End

6.4 Associating Analysis Objects

Select an associated analysis object for a test task. If the test task has been enabled, analysis objects cannot be associated.

Prerequisites

A test task is not started.

Procedure

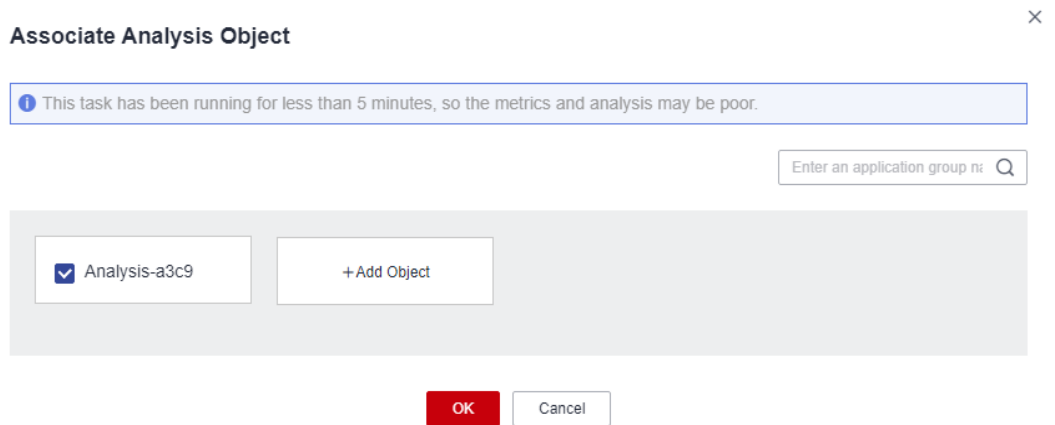
Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired PerfTest project to go to the details page.

Step 3 On the **Test Tasks** tab page, click **...** in the **Operation** column of the test task to be associated with the analysis object, and then choose **Associate Analysis Object**.

Step 4 Select the analysis object to be associated. If you need to associate another analysis object, click **Add Object** to add one by referring to [Setting Intelligent Analysis](#). Then select the added analysis object to associate it, and click **OK**.

Figure 6-2 Associating analysis objects



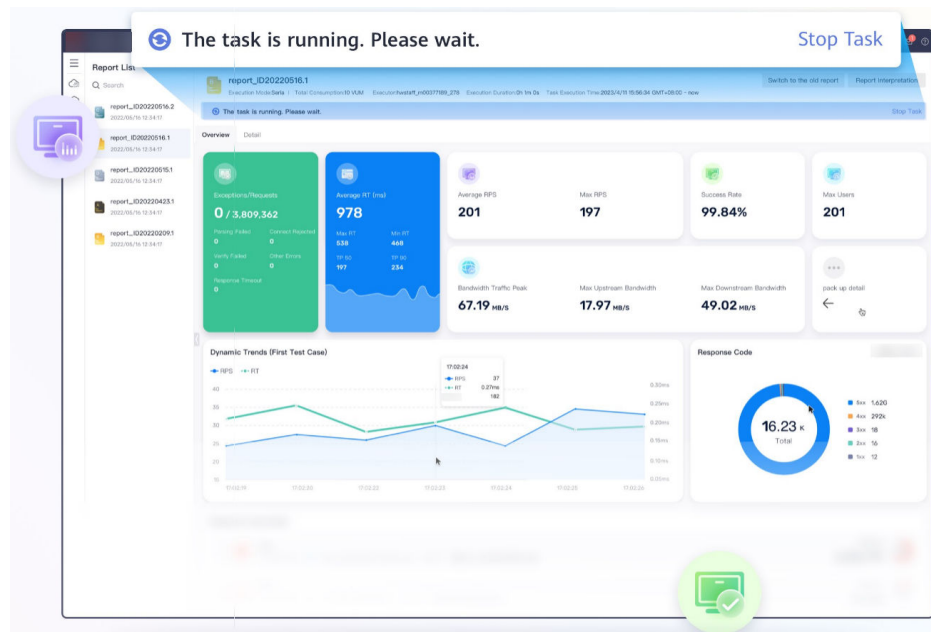
----End

7 PerfTest Report Management

7.1 Test Report Description

CodeArts PerfTest provides comprehensive pressure test reports that cover multiple dimensions, including over 20 performance metrics such as TPS, RT, SuccessRate, TPxx, StatusCode, and execution logs. By analyzing real-time access resources and call chain relationships through visualized data, CodeArts PerfTest provides a thorough evaluation of performance metrics and compares multiple rounds of reports. This enables quick and accurate identification of performance bottlenecks, ensuring smooth service rollout.

CodeArts PerfTest provides real-time and offline test reports for you to view and analyze test data anytime.



Report Overview

For details about the test report, see [Table 7-1](#).

This report shows the response performance of the tested system in a scenario with a large number of concurrent users. To understand the report, see the following information:

- **Statistical dimension:** In this report, RPS, response time, and concurrency are measured in a single case. If a request has multiple packets, the request is considered successful only when all the packets are responded to. The response time of the request is the sum of the response time of the packets.
- **Response timeout:** If the corresponding TCP connection does not return the response data within the set response timeout (5s by default), the case request is counted as a response timeout. Possible causes include: the tested server is busy or crashes, or the network bandwidth is fully occupied.
- **Verification failure:** The response packet content and response code returned from the server do not meet the expectation (the default expected response code of HTTP/HTTPS is 200), such as code 404 or 502. A possible cause is that the tested service cannot be processed normally in scenarios with a large number of concurrent users. For example, a database bottleneck exists in the distributed system or the backend application returns an error.
- **Parsing failure:** All response packets are received, but some packets are lost. As a result, the entire case response is incomplete. This may be caused by network packet loss.
- **Bandwidth:** This report collects statistics on the bandwidth of the execution end of CodeArts PerfTest. The uplink indicates the traffic sent from CodeArts PerfTest, and the downlink indicates the received traffic. In the external pressure test scenario, you need to check whether the EIP bandwidth of the executor meets the uplink bandwidth requirement and whether the bandwidth exceeds the downlink bandwidth of 1 GB.
- **Requests per second (RPS):** indicates the number of requests per second. Average RPS = Total number of requests in a statistical period/Statistical period.
- **How to determine the quality of tested applications:** According to the service quality definition of an application, the optimal status is that there is no response or verification failure. If there is any response or verification failure, it must be within the defined service quality range. Generally, the value does not exceed 1%. The shorter the response time is, the better. User experience is considered good if the response time is within 2s, acceptable if it is within 5s and needs optimization when it is over 5s. TP90 and TP99 can objectively reflect the response time experienced by 90% to 99% of users.
- **SLA Result:** If you have configured SLA rules for a test case, you can view the SLA report to check whether the SLA rules are triggered during testing. In the SLA report, **Request Name** is the name of the request in the test case. **Rules** indicates the configured SLA rules. **Average Value** (displayed as the current value for a running task) indicates the average value of service metrics and the total number of times that SLA rules have been triggered since the pressure test starts. **Trigger Events** displays the request name and when and how many times the SLA rules were triggered during testing.

Table 7-1 Test report description

Parameter	Description
Total Number of Metrics	<p>Total number of metrics of all cases.</p> <ul style="list-style-type: none">● Maximum Concurrency: indicates the maximum number of concurrent virtual users.● RPS: requests per second.● Response Time: duration from the time when a client sends a request to the time when the client receives a response from the server.● Response Code: distribution of response codes during the running of a pressure test task.● Bandwidth: records the real-time bandwidth usage during the running of a pressure test task.● SLA Events: occurrence of an event defined in the SLA.● Normal Response: indicates the number of case responses that pass the set checkpoints. If no checkpoints are set, the number of case responses with code 2XX will be calculated by default.● Abnormal Response: indicates the number of parsing failures, verification failures, response timeout, 3XX, 4XX, 5XX, and connection failures.● Success Rate: Number of requests that are normally returned/Total number of requests.
Average RPS	Total number of requests in a statistical period/Statistical period
Average RT	Average response time of all requests sent in a second
Concurrent Users	Changes on the number of concurrent virtual users during testing
Bandwidth (kbit/s)	<p>Records the real-time bandwidth usage during the running of a pressure test task.</p> <ul style="list-style-type: none">● Uplink bandwidth: speed at which the CodeArts PerfTest execution node sends out data.● Downlink bandwidth: speed at which the CodeArts PerfTest execution node receives data

Parameter	Description
Response distribution statistics	<p>Number of cases processed per second for normal return, parsing failure, verification failure, response timeout, connection rejection, and other errors. This metric is related to the think time, concurrent users, and server response capability. For example, if the think time is 500 ms and the response time of the last request for the current user is less than 500 ms, the user requests twice per second.</p> <ul style="list-style-type: none">● Normal Response: indicates the number of case responses that pass the set checkpoints. If no checkpoints are set, the number of case responses with code 2XX will be calculated by default.● Abnormal Response: indicates the number of parsing failures, verification failures, response timeout, 3XX, 4XX, 5XX, and connection failures.● Parsing Failure: indicates the number of HTTP responses that fail to be parsed.● Verification Failure: indicates the number of case responses that do not pass the set checkpoints. If no checkpoints are set, the number of case responses returning contents other than 2XX will be calculated by default.● Response Timeout: indicates the number of case requests that do not receive responses from the server within 5 seconds after a request packet is sent.● Rejected: indicates the number of rejected connection requests.● Others: indicates the number of other errors.
Response Code statistics	1XX/2XX/3XX/4XX/5XX
Response Time Ratio	Ratio of response time of cases.

Parameter	Description
Maximum Response Time of TP (ms)	<p>If you want to calculate the top percentile XX (TPXX) for a request, collect all the response time values for the request over a time period (such as 10s) and sort them in an ascending order. Remove the top $(100-XX)\%$ from the list, and the highest value left is the value of TPXX.</p> <ul style="list-style-type: none">• TP50: Remove the top 50% from the list, and the highest value left is the value of TP50.• TP75: Remove the top 25% from the list, and the highest value left is the value of TP75.• TP90: Remove the top 10% from the list, and the highest value left is the value of TP90.• TP95: Remove the top 5% from the list, and the highest value left is the value of TP95.• TP99: Remove the top 1% from the list, and the highest value left is the value of TP99.• TP99.9: Remove the top 0.1% from the list, and the highest value left is the value of TP99.9.• TP99.99: Remove the top 0.01% from the list, and the highest value left is the value of TP99.99.



7.2 Viewing a Real-Time Test Report

You can view the monitored metrics during a performance test in a real-time test report.

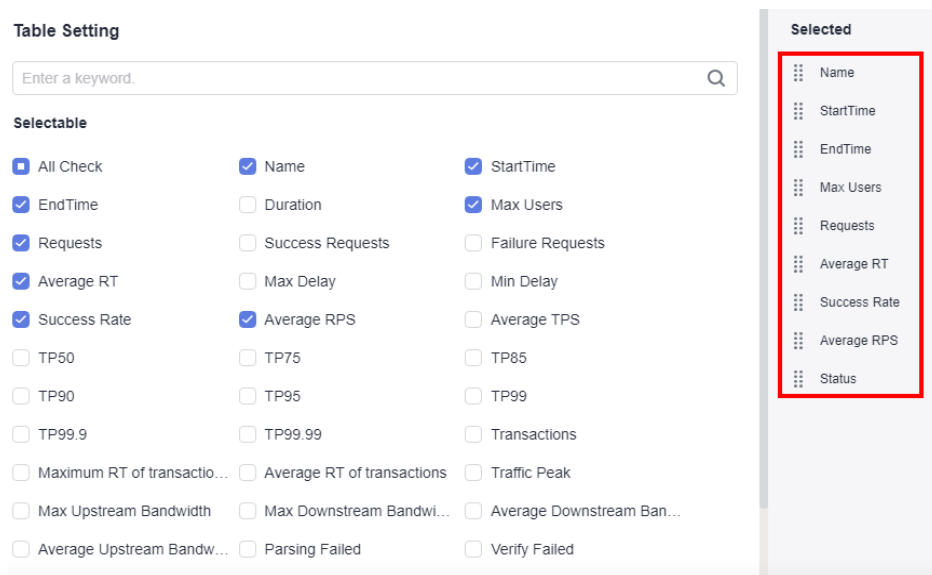
Prerequisites

A test task has been started.



Procedure

- Step 1** Log in to the CodeArts PerfTest console, choose **PerfTest Projects** in the left navigation pane, and choose  in the row containing the desired test project.
- Step 2** On the **Performance Reports** tab page, locate the desired task. The task must be running.
- Step 3** Click the task name or click  in the **Operation** column to view the real-time test report. Click **Stop Task** in the upper right corner of the page to stop the task. For details about the parameters, see [Test Report Description](#).
- Step 4** On the **Overview** tab page, you can view the number of exceptions/total requests, average RT, RPS, maximum concurrency, success rate, SLA alarms, bandwidth, dynamic trend, and response codes.
- Step 5** On the **Detail** tab page, you can view the logs, general test metrics, request details, and the SLA result of the test case.

1. Click **View Log**. In the displayed dialog box, view request logs, event logs, and pod information. Ten request logs are displayed based on the name, return code, and result.
Request logs whose case log collection policy is set to the request mode can be filtered by return code, result, or request name. The log collection policy is set in the advanced configuration of the test case. For details, see [Advanced Configuration](#).
2. During the execution of a test task, you can click **Process Voltage Adjustment** to dynamically adjust the pressure based on the real-time data. For the pressure mode, only the concurrency mode, TPS mode, and duration policy are supported.
3. The sampling modes displayed in the report details are equidistant sampling value and equidistant average value.
 - Equidistant sampling value: Based on the case execution duration, the trend chart of cases whose execution duration is longer than 45 minutes is displayed with sampling points at a fixed interval.
 - Equidistant average value: Based on the case execution duration, the trend chart of cases whose execution duration is longer than 45 minutes is displayed with average values at a fixed interval.
4. Click the **Data View** drop-down list box and enter a keyword to search for the required case or request data. You can also click the case or request data to be displayed in the case directory in the **Data View** drop-down list box.
5. On the **Detail** tab page, you can also click **List** to go to the report metric summary page.
 - Click **Customized Column**. In the displayed dialog box, select the items to be displayed, and drag the selected items in the **Selected** list on the right to change the item sequence.

Figure 7-1 Table settings

- Click  in the **Operation** column to view logs.

- Click  in the **Operation** column to adjust the process voltage. For the pressure mode, only the concurrency mode, TPS mode, and duration policy are supported.
- Click  in the **Operation** column to edit a test case.

Step 6 If the test task has been associated with **Application Monitor (APM)** in the intelligent analysis object, **Invoke Analysis** tab is displayed in the test report. On the **Invoke Analysis** tab, you can set search criteria to view the APM invoking details.

If the test task has been associated with **Node Monitor (AOM)** in the intelligent analysis object, **Monitored Indicator** tab is displayed in the test report. On the **Monitored Indicator** tab, you can view information such as CPU usage, memory usage, disk read speed, and disk write speed.

----End


7.3 Viewing an Offline Test Report

After a pressure test is complete, the system generates an offline test result report.


Prerequisites

The test task is complete.

Procedure

Step 1 Log in to the CodeArts PerfTest console, choose **PerfTest Projects** in the left navigation pane, and choose  in the row containing the desired test project.

Step 2 On the **Performance Reports** tab page, select a task to view its test report.

Step 3 Click the task name, or click  in the **Operation** column. In the **Report List** on the left, select the offline report to be viewed. For details, see [Table 7-1](#).

NOTE

- You can download an offline PDF report to a local PC and export the raw data (in CSV) for further processing.
- CodeArts PerfTest retains offline report data for one year.

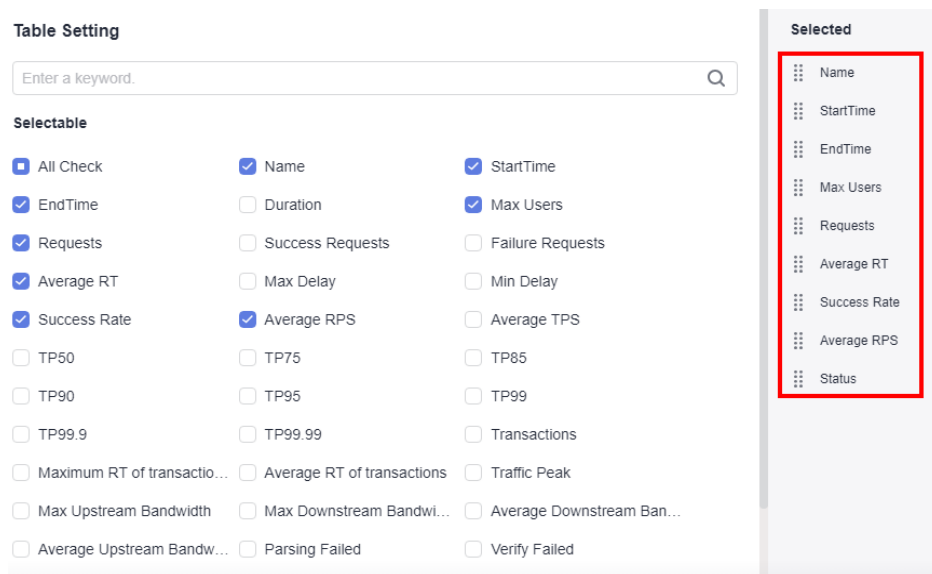
Step 4 On the **Overview** tab page, you can view the number of exceptions/total requests, average RT, RPS, maximum concurrency, success rate, SLA alarms, bandwidth, dynamic trend, and response codes.



Step 5 On the **Detail** tab page, you can view the logs, general test metrics, request details, and the SLA result of the test case.

1. Click **View Log**. In the displayed dialog box, view request logs, event logs, and pod information. Ten request logs are displayed based on the name, return code, and result.

Request logs whose case log collection policy is set to the request mode can be filtered by return code, result, or request name. The log collection policy is

- set in the advanced configuration of the test case. For details, see [Advanced Configuration](#).
- The sampling modes displayed in the report details are equidistant sampling value and equidistant average value.
 - Equidistant sampling value: Based on the case execution duration, the trend chart of cases whose execution duration is longer than 45 minutes is displayed with sampling points at a fixed interval.
 - Equidistant average value: Based on the case execution duration, the trend chart of cases whose execution duration is longer than 45 minutes is displayed with average values at a fixed interval.
 - Click the **Data View** drop-down list box and enter a keyword to search for the required case or request data. You can also click the case or request data to be displayed in the case directory in the **Data View** drop-down list box.
 - On the **Detail** tab page, you can also click **List** to go to the report metric summary page.
 - Click **Customized Column**. In the displayed dialog box, select the items to be displayed, and drag the selected items in the **Selected** list on the right to change the item sequence.

Figure 7-2 Table settings

- Click  in the **Operation** column to view logs.
- Click  in the **Operation** column to edit a test case.

Step 6 If the test task has been associated with **Application Monitor (APM)** in the intelligent analysis object, **Invoke Analysis** tab is displayed in the test report. On the **Invoke Analysis** tab, you can set search criteria to view the APM invoking details.

If the test task has been associated with **Node Monitor (AOM)** in the intelligent analysis object, **Monitored Indicator** tab is displayed in the test report. On the


Monitored Indicator tab, you can view information such as CPU usage, memory usage, disk read speed, and disk write speed.

----End

7.4 Report Comparison

For the test reports generated at different time or under different conditions, you can view the test result by comparing the reports.

Procedure

- Step 1** Log in to the CodeArts PerfTest console, choose **PerfTest Projects** in the left navigation pane, and choose  in the row containing the desired test project.
- Step 2** On the **Performance Reports** tab page, locate the desired task, and click the task name.
- Step 3** In the **Report List** area, click **Report Comparison**.
- Step 4** Select desired test reports and click **OK**.

NOTE

You can select a maximum of three offline reports for comparison. The first selected report is used as the baseline report.

- Step 5** Select a case from the **Test Case Comparison** box to compare the metrics of this case in different reports.
- Step 6** Test case comparison supports metric filtering.
 1. Click **Custom Comparison**. In the displayed dialog box, select the metrics to be displayed.
 2. (Optional) Drag the selected items in the **Selected** list on the right of the dialog box to change the item sequence.
 3. Click **OK**.

----End

8 Transaction Management

8.1 Creating a Transaction

A transaction refers to a complete operation process from end to end, such as a login, criteria-based query, and payment. A transaction can be used by multiple test cases. CodeArts PerfTest supports flexible combination of transactions. You can create multiple transactions to a test project.


 **NOTE**

The extensible test project does not support the transaction library function.

Prerequisites

A test project has been created. For details on how to create a test project, see [Creating a Test Project](#).

Procedure

- Step 1** Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.
- Step 2** Locate a PerfTest project to be edited and click  to edit the transaction library.
- Step 3** On the **Transaction Library** tab page, click create a transaction and configure the following parameters:
 - **Transaction Name:** name of a customized transaction.
 - **Type:** Normal transaction refers to transactions of performance tests in normal scenarios, including API tests of applications built based on HTTP/HTTPS/TCP/UDP/HLS/RTMP/WebSocket/HTTP-FLV/MQTT protocols. Multiple transaction requests can be connected with each other.

 **NOTE**

A maximum of 40 requests can be added to a normal transaction.

Click **OK**.

Step 4 Add requests to the transaction and click **OK**.

When the transaction type is **Normal**, you can add four types of request compositions: packet, think time, response extraction, and checkpoint. Packet is mandatory.

- **Packet:** data blocks transmitted between HTTP-based applications. For details, see [Adding Request Information \(Packet\)](#).
- **Think Time:** time interval between two actions. For details, see [Adding Request Information \(Think Time\)](#).
- **Response Extraction:** If multiple packets exist in the same transaction, use regular expressions or JSON to extract the output of the previous packet for the input of the next packet. For details, see [Adding Request Information \(Response Extraction\)](#).
- **Checkpoint:** used to verify if contents returned by servers are correct through customized verification information. For details, see [Adding Request Information \(Checkpoint\)](#).

Step 5 (Optional) A transaction contains one or more requests. If your transaction contains multiple requests, click **Add Request** to add more requests.

----End


8.2 Importing Recorded Scripts

In performance tests, virtual users simulate real users to use the tested system. The simulation process, such as **login, search, and purchase** in e-commerce applications, is implemented through recorded scripts. CodeArts PerfTest supports the record tool plug-in on the Chrome browser (version 26 or later versions). In the pressure test system, you can manually perform operations to capture request content and generate .json files. The recorded scripts simulate the actual operations of users, reducing the workload of manual packet writing and facilitating the use of the CodeArts PerfTest system.

Restrictions

The CodeArts PerfTest record tool requires version 26 or later versions of Chrome.

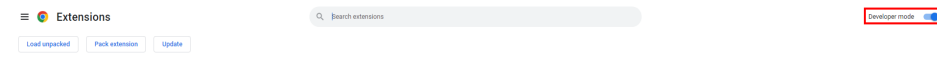
 **NOTE**

Click  in the upper right corner of the browser and choose **Help > About Chrome** to view the browser version.

Downloading and Installing the Chrome Recording Plug-in

- Step 1** [Download](#) the recording plug-in for Chrome.
- Step 2** Decompress the installation package using a decompression tool.
- Step 3** Enter **chrome://extensions/** in the address bar. The extension page is displayed.
- Step 4** Enable the developer mode on the extension page.

Figure 8-1 Enabling the developer mode



Step 5 Click **Load unpacked** and load the folder decompressed in [Step 2](#).


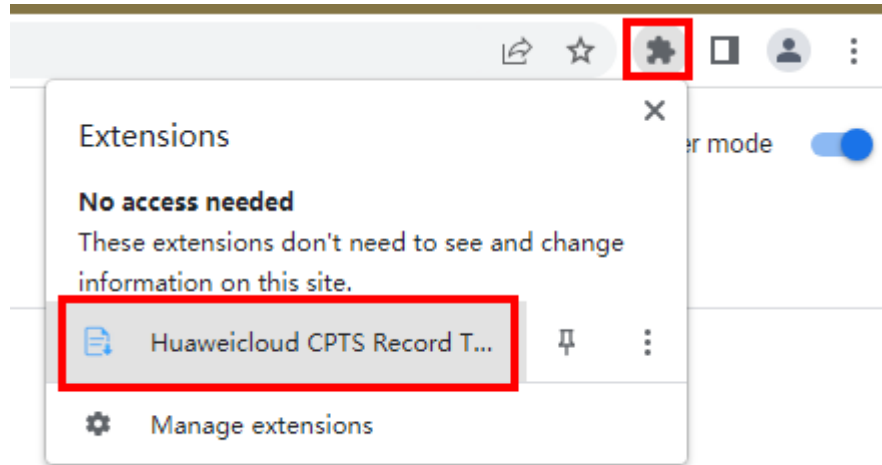
Step 6 When the installation is complete, the icon of the recording tool  is displayed in **Extensions** in the upper right corner of the browser.


Figure 8-2 Recording tool



----End

Recording Scripts

Use the CodeArts PerfTest recording tool to record scripts.

Step 1 Open Chrome and click the icon of the CodeArts PerfTest recording tool  in the upper right corner of the browser. The recording box is displayed.

Step 2 Click **Start Recording**.

Step 3 On the new tab page of the browser, enter the URL to be tested. The recording tool automatically records the HTTP requests during the access operation.

Step 4 When the recording is complete, switch to the recording tool page and click **Pause Recording**.

Step 5 By default, the record tool displays all types of requests. You can select the request composition based on the service requirements by clicking the **Filter Content** check box.

- other: others
- script: script
- xmlhttprequest: XML HTTP request
- main_frame: main frame
- stylesheet: style sheet

- image: image
- font: font file
- Ping: ping test

Step 6 If you select a request, the content of the request is displayed on the right of the page.

 **NOTE**

If the recorded content is not required or a new request needs to be recorded, perform the following steps to record scripts again:

1. Click **Clear Content** to clear the recorded request content.
2. Click **Start Recording**.
3. Perform operations on the new tab page and record again.

Step 7 Click **Export**, select the request to be exported, and click **Confirm Export**. The **Export Content** dialog box is displayed. Click **Generate .json File** to save the file to the local host.

 **NOTE**


Plug-in recording script. The body type does not support the payload format.

Step 8 Close the tab where the record tool is located to disable the record tool.

----End

Importing Recorded Scripts

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.

Step 3 Import a recorded script using either of the following methods:

Method 1: On the **Transactions** tab page, click **Import Script**. Set **Script Type** to **Manual recording script**. Select the transaction and recorded script file to be imported.

 **NOTE**

The recorded script contains one or more packet requests and can be imported to the corresponding transaction mode.


Method 2: On the right of the desired transaction name, click **Import Recorded Script** and select the script file to be uploaded.

Step 4 Click **Import**. If the scripts are imported successfully, the system displays a message indicating this.

Step 5 Under the transaction, you can view the packets successfully imported.

 **NOTE**


The number of packets successfully imported is the same as the number of requests in the uploaded script file.

Step 6 (Optional) Click  to modify the imported scripts by referring to [Adding Request Information \(Think Time\)](#) based on service requirements.

----End

8.3 Importing a Swagger File

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.

Step 3 On the **Transactions** tab page, click **Import Script**.

Step 4 Set **Script Type** to **Swagger file**, click **Upload**, and select the Swagger file to be uploaded.

NOTE

Only Swagger files in .yaml, .yml, and .json formats are supported.

Step 5 Click **Import**.

----End

8.4 Debugging Transactions


After adding or modifying a transaction, you can quickly find syntax or configuration errors through debugging.


Prerequisites

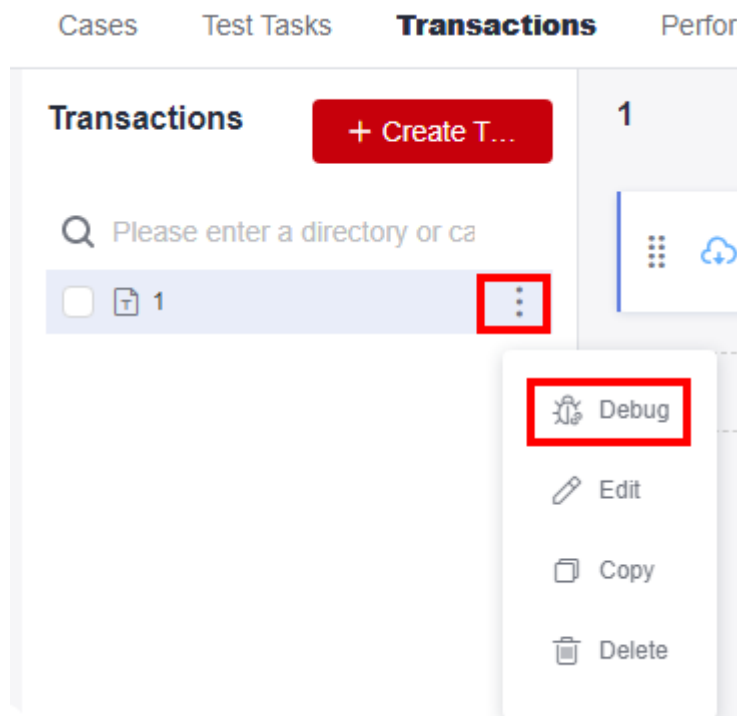
- The resource group is in the **Running** state.
- Ports 32001 and 32002 on the debugging node of the resource group are enabled in the security group.
- The network between the debugging node of the resource group and the tested application is normal.

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.

Step 3 On the **Transactions** tab page, select the transaction to be debugged and click **Debug** in the upper right corner of the page. You can also click  next to the transaction name and click **Debug**.



Step 4 Select the target test resource group as the executor, and click **Start** to start debugging.

Step 5 Click **View log** to view the transaction debugging details.

If an error is reported during the debugging, modify the transaction requests according to the log information, and then debug the transaction again.

Step 6 On the **Debug log** tab page, you can view the debugging history.

----End

8.5 Managing Transactions


After a transaction is created, you can modify, delete, and copy the transaction.

NOTE

A transaction used by a test cannot be deleted.

Modifying a Transaction

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.

Step 3 On the **Transactions** tab page, click  next to the transaction name and choose **Edit**.


Step 4 Modify the transaction name and click **OK**.

----End

Deleting a Transaction

Deleted transactions cannot be restored. Exercise caution when deleting a transaction.

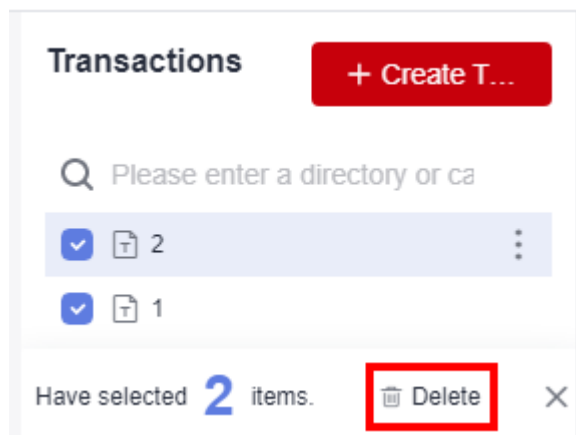
Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.

Step 3 On the **Transactions** tab page, click  next to the transaction name and click **Delete**. Delete the transactions as prompted.

You can also select a transaction name in the transaction list. In the floating box displayed in the lower part of the page, click **Delete** to delete the transaction as prompted.

Figure 8-3 Deleting transactions




NOTE

If some transactions to be deleted in batches are referenced by a test task, only transactions that are not referenced can be deleted.

----End

Copying a Transaction

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.

Step 3 On the **Transactions** tab page, click  next to the transaction name and click **Copy**.

Step 4 Enter the name of the new transaction and click **OK**.


----End


8.6 Managing Transaction Request Information

After a transaction request is created, you can insert, modify or delete it.

Inserting Request Information

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.


Step 3 On the **Transactions** tab page, select the transaction request to be inserted, click  in the upper right corner, and click **Insert a request**.


Step 4 Configure the request to be inserted and then click **OK**.

----End

Modifying a Transaction Request

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.


Step 3 On the **Transactions** tab page, select the transaction whose request information is to be modified, click  next to the desired request, and modify the request information as required. Then click **OK**.

----End


Disabling/Enabling a Transaction Request

If you want to retain a request of a transaction but do not want to use it, you can disable it. If you want to resume the use of the request, you can enable it again.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.

Step 3 Disable a transaction request.

On the **Transactions** tab page, select the transaction whose request is to be disabled, click  next to the desired request to disable it. Other requests are not affected.

Step 4 Enable a transaction request.

On the **Transactions** tab page, select the transaction whose request is to be


enabled, click  next to the desired request to enable it. Other requests are not affected.


----End

Deleting a Transaction Request

Deleted transaction requests cannot be restored. Exercise caution when deleting a transaction request.

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Locate a PerfTest project to be edited and click  to edit the transaction library.

Step 3 On the **Transactions** tab page, select the transaction whose request is to be deleted, click  next to the desired request to delete it. Other requests are not affected.

----End

9 JMeter Test Project Management

9.1 Introduction to the JMeter Project

If you use JMeter to develop pressure test scripts, it may be a challenge to execute existing scripts efficiently. CodeArts PerfTest is compatible with JMeter scripts and their execution. By importing existing JMeter scripts into a JMeter project, you can easily conduct pressure tests and significantly reduce the time required to build a large-scale JMeter testing environment. This allows you to quickly expand your test capacity as needed, resulting in improved overall efficiency and the ability to complete pressure testing in a matter of hours instead of weeks.



9.2 Managing JMeter Projects

A JMeter test project provides the native JMeter engine. You can import JMeter scripts into JMeter test projects, use the native JMeter engine to quickly initiate high-concurrency performance tests, and view complete performance test reports.

Creating a JMeter Project

- Step 1** Log in to the CodeArts PerfTest console, choose **JMeter Projects** in the left navigation pane, and click **Create JMeter Project**.
- Step 2** Set the basic information by referring to [Table 9-1](#).

Table 9-1 Creating a JMeter test project

Parameter	Description
Project Name	Name of a new test project. NOTE The project name can contain a maximum of 128 bytes, including letters, digits, hyphens (-), underscores (_), slashes (/), and periods (.).
Description	Description of a new test project.



- Step 3** When the configuration is complete, click **OK**.

When the JMeter project is created, you can add a test plan for it. For details, see [Managing JMeter Test Plans](#).

----End

Renaming and Deleting a Test Project

CodeArts PerfTest allows you to modify and delete a created JMeter project.


- Step 1** Log in to the CodeArts PerfTest console and choose **JMeter Projects** in the left navigation pane.
- Step 2** In the JMeter test project list, click  in the row containing the desired project. Enter a new project name and click **OK**.
- Step 3** In the JMeter test project list, click  > **Delete Project** in the row containing the desired project. Delete the project as prompted.

----End

9.3 Managing JMeter Test Plans

Creating a Test Plan

A test plan is used to initiate a performance test based on the defined JMeter files, JAR packages, and variable files.

Step 1 Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.

Step 2 On the **Test Plans** tab, click **Create Task Plan**.

Step 3 In the dialog box that is displayed, click **Select File**, select a .jmx file, and click **Upload**. When the file is imported, the dialog box closes automatically and the **Test Plans** tab is displayed. You can view the added test plan.

 **NOTE**

- The name (including the extension) of a JMX file can contain up to 64 bytes. The file size can be up to 10 MB.
- Basic JMeter scripts are supported, including thread group, HTTP request, HTTP Header Manager, request parameters, BeanShell, and timeout interval. You are advised to use a script of version 5.2.1 or later. Otherwise, a script parsing error may occur.

Step 4 On the **Test Plans** tab page, click **Third Party Jar**. In the displayed dialog box, click **Select File**, select and import the JAR package on which the test plan depends. After the package is imported, close the dialog box.

 **NOTE**


The name (including the extension) of a JAR package can contain up to 64 bytes. The package size can be up to 80 MB.

Step 5 View the newly added test plan on the **Test Plans** tab.

----End

Pressure Configuration

After creating a test plan, you can edit phase parameters.

Step 1 Log in to the CodeArts PerfTest console and choose **JMeter Projects** in the left navigation pane. Click  in the row containing the test project to which the desired test plan belongs.

Step 2 On the **Test Plans** tab page, click the name of the desired test plan. On the **Plan Detail** page, you can view the stages of different thread groups. Set the basic information by referring to [Table 9-2](#).

Table 9-2 Pressure configuration



Parameter	Description
Execution Policy	The value can be Duration or Count . <ul style="list-style-type: none">• Duration: Tests will be performed based on the specified duration.• Count: Tests will be performed based on the total number of tests.
Threads	Number of concurrent virtual users.
Ramp-up Period	It corresponds to the ramp-up period in JMeter scripts. The parameter indicates the time required for increasing the number of concurrent users from 0 to the specified value. NOTE This parameter is involved only in Phase1 .
Loop Count	Number of execution times of each virtual user. NOTE This parameter needs to be configured only when Execution Policy is set to Count .
Specify Thread Lifetime	If you enable this option, you can configure the Duration and Startup Delay parameters. NOTE This parameter needs to be configured only when Execution Policy is set to Count .
Duration	If the actual test duration reaches the maximum duration and the number of execution times does not reach the number of cycles, the thread group execution ends. NOTE This parameter can be configured only when Execution Policy is set to Count and Specify Thread Lifetime is enabled.
Duration	How long the test will run. It is recommended that the duration be set to at least 300 seconds. NOTE This parameter needs to be configured only when Execution Policy is set to Duration .
Startup Delay	When a test task is started, CodeArts PerfTest will wait the duration of startup delay before generating virtual users. NOTE This parameter can be configured when Execution Policy is set to Count and Specify Thread Lifetime is enabled. When Execution Policy is set to Duration , you can modify this parameter for Phase1 .

Step 3 Click **OK**.

----End

Debugging a Test Plan

When a test plan is added or modified, you can debug it to quickly detect syntax or configuration errors and ensure that the model is available in a performance test.

- Step 1** Log in to the CodeArts PerfTest console and choose **JMeter Projects** in the left navigation pane. Click  in the row containing the test project to which the desired test plan belongs.
- Step 2** On the **Test Plans** tab page, click the name of the task to be debugged. On the plan details page that is displayed, click  **Debug** in the upper part of the page.
- Step 3** On the debugging page, select a created private resource group as the executor and click **Start** to start debugging. You can view the debugging progress on the debugging page.
- Step 4** After the debugging is complete, if an error is reported, modify the case based on the exception information and debug the case again. For details on how to import a .jmx file, see the section on modifying a test plan in [Managing Test Plans](#).
- Step 5** Click **View log** to view the test plan debugging details.
- Step 6** On the Debug log tab page, you can view the debugging history.



----End

Associating an Analysis Object (Optional)

Prerequisites

The test plan is not enabled.

Procedure

- Step 1** Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.
- Step 2** On the **Test Plans** tab page, select the test task to be associated with the analysis object, and click .
- Step 3** Select the analysis object to be associated. If you need to associate another analysis object, click **Add Object** to add one by referring to [Configuring Intelligent Analysis for JMeter Projects](#). Then select the added analysis object to associate it, and click **OK**.

----End

Starting a Test Plan



Test plans are a series of tests performed at different pressure points to continuously initiate pressure tests on a system. Testing allows you to obtain and analyze the performance data of the system.

You can add multiple test plans to a test project.

Prerequisites

- The resource group is in the **Running** state.
- Ports 32001 and 32003 on the debugging node of the resource group are enabled in the security group.
- The execution node of the resource group and applications under the pressure test are connected through the network.

Procedure

- Step 1** Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.
- Step 2** Create a test plan. For details, see [Creating a Test Plan](#).
- Step 3** Click  in the row containing the created task.
- Step 4** In the **Start Test Task** dialog box, select a resource group. Only private resource groups are supported for executing JMeter tasks.
- Step 5** Click **Start** to start the test task. Click **View Report** to view the real-time performance report.

NOTE

It is recommended that the duration of a pressure test be at least 300s and the number of concurrent users be set based on the actual situation. Obtain the maximum pressure value for the application through repeated tests by adjusting test data, and perform continuous optimization and verification for the application.


----End

Managing Test Plans

You can manage created test plans.



Enabling test plans in batches

Enable multiple test plans in a test project.

- Step 1** Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.
- Step 2** On the **Test Plans** tab page, select multiple test plans and click **Start**.
- Step 3** Select resource groups of these tasks and click **Enable**.

----End

Advanced configuration of test plans

- Step 1** Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.
- Step 2** On the **Test Plans** tab page, click  in the row containing the desired plan.


Step 3 Configure advanced parameters.


1. **Executor:** The number of executors takes effect only when private resource groups are used. If this parameter is not set, the default policy is used, that is, a single executor supports concurrent JMeter (1000). Executors \geq Maximum number of concurrent users in all phases/1000.
2. **Failure Log Collection Ratio:** By default, the sampling ratio is 10%, and can be set to 1000% at most.

Step 4 When the configuration is complete, click **OK**.


----End

Editing a test plan

Step 1 Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.

Step 2 On the **Test Plans** tab page, click  in the **Operation** column of the desired plan, and choose **Update jmx**.

Step 3 In the displayed **Edit Test Plan** dialog box, import a new .jmx file, and then click **Close**.

Step 4 On the **Test Plans** tab page, click  in the **Operation** column of the desired plan, and choose **Variable File** to import the files to be referenced by the test plan.


 **NOTE**

1. Only .csv (UTF-8 without BOM) and .xlsx files can be imported.
2. The maximum length of a file name (including the extension) is 50 bytes.
3. Maximum file size:
 - .xlsx: 10 MB for the Professional edition and 20 MB for the Enterprise edition.
 - .csv: 10 MB for the Professional edition and 80 MB for the Enterprise edition by default.

----End

Deleting a test plan

Deleted test plans cannot be recovered. Exercise caution when deleting a test plan.

Step 1 Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.


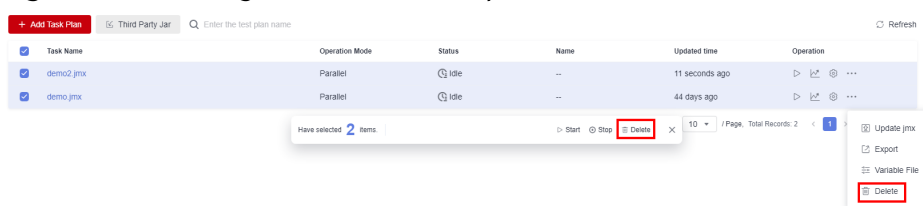

Step 2 On the **Test Plans** tab page, click  in the **Operation** column of the desired plan, and choose **Delete**. You can also select multiple test plans and click **Delete** to delete them all.

Figure 9-1 Deleting one or more test plans


Step 3 Delete the test plans as prompted.

----End

Deleting phases

Step 1 Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.

Step 2 On the **Test Plans** tab page, click the task name to go to the **Plan Detail** page.

Step 3 Select the desired thread group, click  next to the phase to be deleted, and click **OK**.


NOTE

Phase1 cannot be deleted.


----End

Exporting JMX Files, Variable Files, and JAR Packages

Prerequisites: The JMX files, variable files, and JAR packages have been imported.

Step 1 Log in to the CodeArts PerfTest console. In the navigation pane on the left, choose **JMeter Projects**. Then click  in the row containing the desired project.

Step 2 On the **Test Plans** tab page, click  in the **Operation** column of the desired plan, and choose **Export** to export its .jmx file.

Step 3 Click  in the row containing the test plan whose variable file is to be exported and choose **Variable File**. The **Variable File** dialog box is displayed.

Step 4 Click **Export** in the row containing the variable file to be exported.

Step 5 Click **Third Party Jar**. The **Third Party JAR Package** dialog box is displayed.

Step 6 Click **Export** in the row containing the desired JAR package to export it.

----End

9.4 Managing JMeter Test Reports

Test Report Description

Real-time and offline JMeter test reports allow you to view and analyze test data at any time.

For details about the JMeter test report, see [Table 9-3](#).

This report shows the response performance of the tested system in a scenario with a large number of concurrent users. To understand the report, see the following information:

- **Statistical dimension:** In this report, RPS, response time, and concurrency are measured in a single thread group. If a request has multiple packets, the request is considered successful only when all the packets are responded to.

The response time of the request is the sum of the response time of the packets.

- **Response timeout:** If the corresponding TCP connection does not return the response data within the set response timeout (customized in *.jmx files), the thread group request is counted as a response timeout. Possible causes include: the tested server is busy or crashes, or the network bandwidth is fully occupied.
- **Verification failure:** The response packet content and response code returned from the server do not meet the expectation (the default expected response code of HTTP/HTTPS is 200), such as code 404 or 502. A possible cause is that the tested service cannot be processed normally in scenarios with a large number of concurrent users. For example, a database bottleneck exists in the distributed system or the backend application returns an error.
- **Parsing failure:** All response packets are received, but some packets are lost. As a result, the entire case response is incomplete. This may be caused by network packet loss.
- **Bandwidth:** This report collects statistics on the bandwidth of the execution end of CodeArts PerfTest. The uplink indicates the traffic sent from CodeArts PerfTest, and the downlink indicates the received traffic. In the external pressure test scenario, you need to check whether the EIP bandwidth of the executor meets the uplink bandwidth requirement and whether the bandwidth exceeds the downlink bandwidth of 1 GB.
- **Requests per second (RPS):** indicates the number of requests per second. Average RPS = Total number of requests in a statistical period/Statistical period.
- **How to determine the quality of tested applications:** According to the service quality definition of an application, the optimal status is that there is no response or verification failure. If there is any response or verification failure, it must be within the defined service quality range. Generally, the value does not exceed 1%. The shorter the response time is, the better. User experience is considered good if the response time is within 2s, acceptable if it is within 5s and needs optimization when it is over 5s. TP90 and TP99 can objectively reflect the response time experienced by 90% to 99% of users.

Table 9-3 Description of the JMeter test report

Parameter	Description
Total Number of Metrics	<p>Total number of metrics in all thread groups.</p> <ul style="list-style-type: none">● Maximum Concurrency: indicates the maximum number of concurrent virtual users.● RPS: requests per second.● Normal Response: indicates the number of transaction responses that pass the set checkpoints. If no checkpoints are set, the number of transactions that return 2XX response code will be calculated by default.● Response Time: duration from the time when a client sends a request to the time when the client receives a response from the server.● Response Code: distribution of response codes during the running of a pressure test task.● Bandwidth: records the real-time bandwidth usage during the running of a pressure test task.● Abnormal Response: indicates the number of parsing failures, verification failures, response timeout, 3XX, 4XX, 5XX, and connection failures.● Success Rate: Number of requests that are normally returned/Total number of requests.
Average RPS	Total number of requests in a statistical period/Statistical period
Average RT	Changes of the average response time of a pressure test task
Concurrent Users	Changes on the number of concurrent virtual users during testing
Bandwidth (kbit/s)	<p>Records the real-time bandwidth usage during the running of a pressure test task.</p> <ul style="list-style-type: none">● Uplink bandwidth: speed at which the JMeter execution node sends out data.● Downlink bandwidth: speed at which the JMeter execution node receives data.

Parameter	Description
Response distribution statistics	<p>Number of transactions processed per second for normal return, parsing failure, verification failure, response timeout, connection rejection, and other errors. This metric is related to the think time, concurrent users, and server response capability. For example, if the think time is 500 ms and the response time of the last request for the current user is less than 500 ms, the user requests twice per second.</p> <ul style="list-style-type: none">● Normal Response: indicates the number of transaction responses that pass the set checkpoints. If no checkpoints are set, the number of transactions that return 2XX response code will be calculated by default.● Parsing Failure: indicates the number of HTTP responses that fail to be parsed.● Verification Failure: indicates the number of transaction responses that do not pass the set checkpoints. If no checkpoints are set, the number of transactions that return 2XX response code will not be calculated.● Response Timeout: number of case requests with no responses in the TCP connection within the configured response timeout interval.● Rejected: indicates the number of rejected connection requests.● Others: indicates the number of other errors.
Response Code statistics	1XX/2XX/3XX/4XX/5XX
Response Time Ratio	Ratio of response time of cases.

Parameter	Description
Maximum Response Time of TP (ms)	<p>If you want to calculate the top percentile XX (TPXX) for a request, collect all the response time values for the request over a time period (such as 10s) and sort them in an ascending order. Remove the top $(100-XX)\%$ from the list, and the highest value left is the value of TPXX.</p> <ul style="list-style-type: none">• TP50: Remove the top 50% from the list, and the highest value left is the value of TP50.• TP75: Remove the top 25% from the list, and the highest value left is the value of TP75.• TP90: Remove the top 10% from the list, and the highest value left is the value of TP90.• TP95: Remove the top 5% from the list, and the highest value left is the value of TP95.• TP99: Remove the top 1% from the list, and the highest value left is the value of TP99.• TP99.9: Remove the top 0.1% from the list, and the highest value left is the value of TP99.9.• TP99.99: Remove the top 0.01% from the list, and the highest value left is the value of TP99.99.



Viewing a Real-Time Test Report

View the monitoring data of each metric during a pressure test through the real-time test report.

Prerequisites

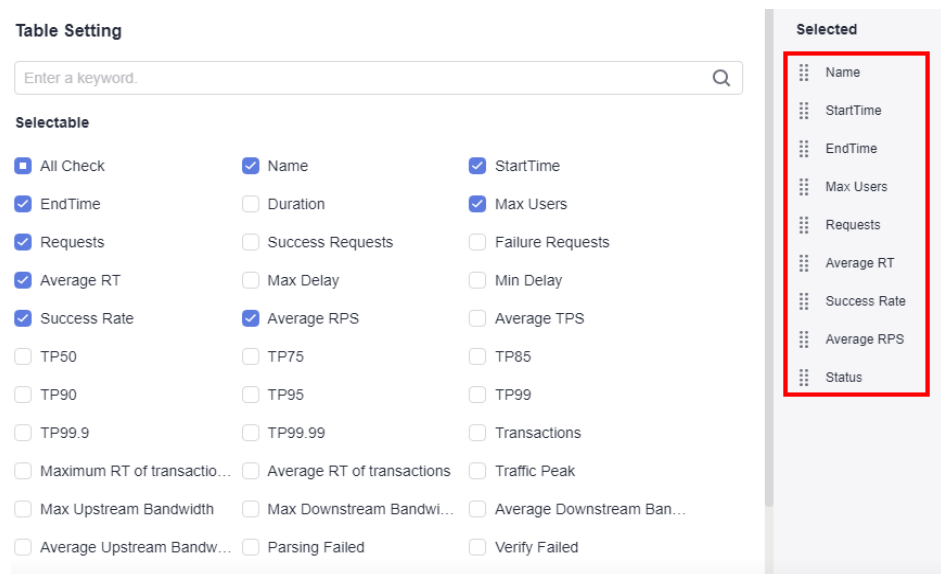
The test task is being executed.



Procedure

- Step 1** Log in to the CodeArts PerfTest console, choose **JMeter Projects** in the left navigation pane, and choose  in the row containing the desired test project.
- Step 2** On the **Performance Reports** tab page, click  in the row containing the test plan whose test report you want to view. For details about the parameters, see [Table 9-3](#). Click **Stop Task** to stop the current task.
- Step 3** On the **Overview** tab page, you can view the number of exception/total requests, average RT, maximum concurrency, success rate, bandwidth, dynamic trend, and response codes.
- Step 4** On the **Detail** tab, you can view the logs, common test metrics, and request details of the test plan.
 1. Click **View Log**. In the **View Logs** dialog box, view request logs, event logs, and pod information. Ten request logs are displayed based on the name, return code, and result.

Request logs can be filtered by return code, result, or request name.

2. The sampling modes displayed in the report details are equidistant sampling value and equidistant average value.
 - Equidistant sampling value: Based on the case execution duration, the trend chart of cases whose execution duration is longer than 45 minutes is displayed with sampling points at a fixed interval.
 - Equidistant average value: Based on the case execution duration, the trend chart of cases whose execution duration is longer than 45 minutes is displayed with average values at a fixed interval.
3. Click the **Data View** drop-down list box and enter a keyword to search for the required thread group or request data. You can also click the thread group or request data to be displayed in the directory in the **Data View** drop-down list box.
4. On the **Detail** tab page, you can also click **List** to go to the report metric summary page.
 - Click **Customized Column**. In the displayed dialog box, select the items to be displayed, and drag the selected items in the **Selected** list on the right to change the item sequence.

Figure 9-2 Table settings

- Click  in the **Operation** column to view logs.
- Click  in the **Operation** column to edit a thread group.

Step 5 If the test plan has been associated with **Application Monitor (APM)** in the intelligent analysis object, the **Invoke Analysis** tab is displayed in the test report. On the **Invoke Analysis** tab, you can set search criteria to view the APM invoking details.

If the test plan has been associated with **Node Monitor (AOM)** in the intelligent analysis object, the **Monitored Indicator** tab is displayed in the test report. On the

Monitored Indicator tab, you can view information such as CPU usage, memory usage, disk read speed, and disk write speed.

----End


Viewing an Offline Test Report


After a pressure test is complete, the system generates an offline test result report.

Prerequisites

The test task is complete.

Procedure

Step 1 Log in to the CodeArts PerfTest console, choose **JMeter Projects** in the left navigation pane, and choose  in the row containing the desired test project.

Step 2 On the **Performance Reports** tab page, click  in the row containing the test plan whose test report you want to view. For details about the parameters, see [Table 9-3](#).

NOTE

- You can download an offline PDF report to a local PC and export the raw data (in CSV) for further processing.
- CodeArts PerfTest retains offline report data for one year.

Step 3 On the **Overview** tab page, you can view the number of exception/total requests, average RT, maximum concurrency, success rate, bandwidth, dynamic trend, and response codes.

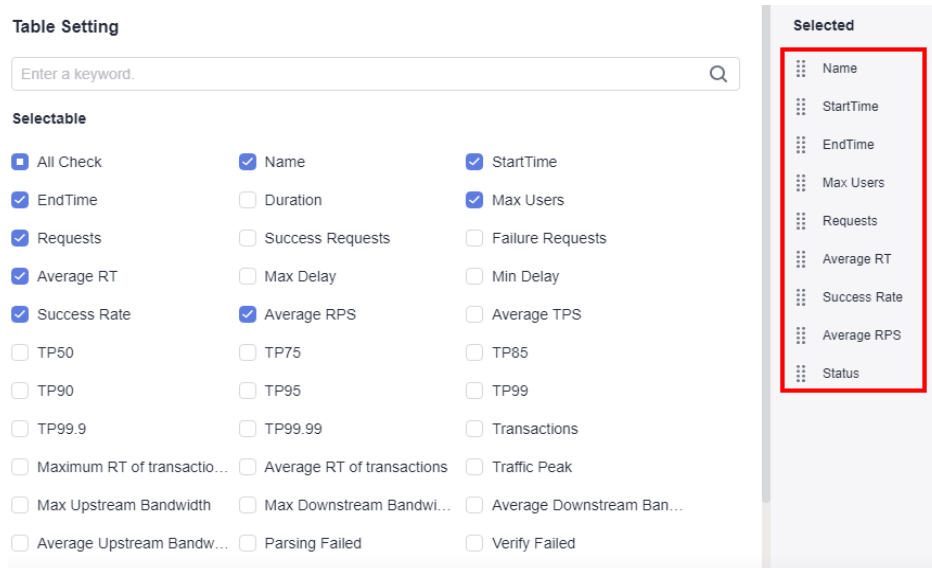
Step 4 On the **Detail** tab, you can view the logs, common test metrics, and request details of the test plan.



1. Click **View Log**. In the **View Logs** dialog box, view request logs, event logs, and pod information. Ten request logs are displayed based on the name, return code, and result.

Request logs can be filtered by return code, result, or request name.

2. The sampling modes displayed in the report details are equidistant sampling value and equidistant average value.
 - Equidistant sampling value: Based on the case execution duration, the trend chart of cases whose execution duration is longer than 45 minutes is displayed with sampling points at a fixed interval.
 - Equidistant average value: Based on the case execution duration, the trend chart of cases whose execution duration is longer than 45 minutes is displayed with average values at a fixed interval.
3. Click the **Data View** drop-down list box and enter a keyword to search for the required thread group or request data. You can also click the thread group or request data to be displayed in the directory in the **Data View** drop-down list box.
4. On the **Detail** tab page, you can also click **List** to go to the report metric summary page.

- Click **Customized Column**. In the displayed dialog box, select the items to be displayed, and drag the selected items in the **Selected** list on the right to change the item sequence.

Figure 9-3 Table settings

- Click  in the **Operation** column to view logs.
- Click  in the **Operation** column to edit a thread group.

Step 5 If the test plan has been associated with **Application Monitor (APM)** in the intelligent analysis object, the **Invoke Analysis** tab is displayed in the test report. On the **Invoke Analysis** tab, you can set search criteria to view the APM invoking details.



If the test plan has been associated with **Node Monitor (AOM)** in the intelligent analysis object, the **Monitored Indicator** tab is displayed in the test report. On the **Monitored Indicator** tab, you can view information such as CPU usage, memory usage, disk read speed, and disk write speed.

----End

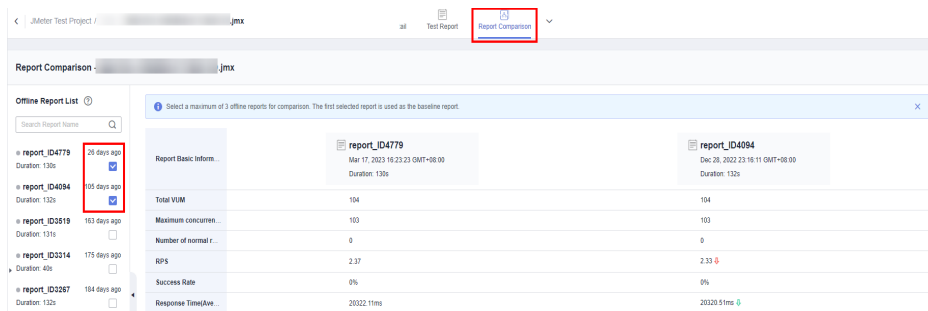
Report Comparison

For the test reports generated at different time or under different conditions, you can view the test result by comparing the reports.

Procedure

- Step 1** Log in to the CodeArts PerfTest console, choose **JMeter Projects** in the left navigation pane, and choose  in the row containing the desired test project.
- Step 2** On the **Performance Reports** tab page, click the name of the desired plan or click  in the **Operation** column.

Step 3 On the **Report Comparison** tab page, select the desired test reports.



Report Basic Inform.	report_ID4779 Mar 17, 2023 16:23:23 GMT+08:00 Duration: 136s	report_ID4094 Dec 28, 2022 23:16:11 GMT+08:00 Duration: 132s
Total VUM	104	104
Maximum concurren...	100	100
Number of normal r...	0	0
RPS	2.37	2.33
Success Rate	0%	0%
Response Time(Ave...	20322.11ms	20320.91ms

NOTE

You can select a maximum of three offline reports for comparison. The first selected report is used as the baseline report.

Step 4 Select a thread group from the **Thread Group Comparison** box to compare the metrics of this thread group in different reports.

Step 5 Thread group metric comparison supports metric filtering.

1. Click **Custom Comparison**. In the displayed dialog box, select the metrics to be displayed.
2. (Optional) Drag the selected items in the **Selected** list on the right of the dialog box to change the item sequence.
3. Click **OK**.

----End

9.5 Configuring Intelligent Analysis for JMeter Projects

You can add analysis objects for the JMeter test project you created.

NOTE


Currently, CodeArts PerfTest in the Singapore Region has been connected to APM 2.0, and the APM 1.0 configuration in CodeArts PerfTest is invalid. Use APM 2.0 when setting intelligent analysis.

Prerequisites

- APM has been enabled, a probe has been deployed, and an application has been generated.
- AOM has been enabled, and a probe has been deployed.

Adding an Analysis Object

Step 1 Log in to the CodeArts PerfTest console and choose **JMeter Projects** in the left navigation pane.

Step 2 In the test project list, click  in the row containing the desired test project.

Step 3 On the **Intelligent Analysis** tab page, click **Add Analysis Object**.

Step 4 In the displayed dialog box, the system automatically generates the analysis object name, but you can change it. Select the analysis object to be added. The analysis object can be either an application group, that is **Application Monitor (APM)**, or a node, that is **Node Monitor (AOM)**. The application group is the monitoring group that is set when APM probes are installed on the application server. The node is the VM where AOM probes are installed.


Step 5 Click **OK** after the configuration.

Step 6 Edit or delete analysis objects.

- Edit an analysis object: On the **Intelligent Analysis** tab page, click  next to the analysis object to modify and edit it as prompted.

 **NOTE**

If an analysis object is associated with a test plan, once the object parameters are edited, those in the test plan are also changed.

- Delete an analysis object: On the **Intelligent Analysis** tab page, click  next to the analysis object to be deleted and delete it as prompted.

 **NOTE**


Analysis objects cannot be deleted if they are being used in a test plan.


----End

Associating Analysis Objects

After the analysis objects are added, associate them with the test plan. If the test task has been enabled, analysis objects cannot be associated.

Step 1 Log in to the CodeArts PerfTest console and choose **JMeter Projects** in the left navigation pane.

Step 2 In the test project list, click  in the row containing the desired test project.

Step 3 On the **Test Plans** tab page, click  in the **Operation** column of the desired plan.

Step 4 Select the analysis object to be associated. To associate a new one, click **Add**. Select the added analysis object to associate it and click **OK**.

----End

9.6 Configuring Resource Tags

Add tags to the created JMeter test project. Each JMeter project can be regarded as a resource. You can tag resources under your account for classification. In scenarios such as cost management, you can quickly view details about a type of resources by tag.

Procedure

Step 1 Log in to the CodeArts PerfTest console and choose **JMeter Projects** in the left navigation pane.

Step 2 In the JMeter test project list, click  > **Modify Tag** in the row containing the desired project.



Step 3 On the **Resource Tags** tab, click **Add Tag**. Select a tag key, and a new key-value pair text box is displayed. You can add multiple tags at a time.

NOTE

- A test project can have a maximum of 20 tags. A tag key must be unique and can have only one value.
- Tag keys of different test projects can be invoked by each other. For example, add tag keys **a** and **b** to Test Project 1. When you add tags to Test Project 2, **a** and **b** will be automatically displayed in the tag key drop-down list.

Step 4 Click **OK** after the configuration.

Step 5 You can edit and delete tags.

- Editing a tag: On the **Resource Tags** tab, click  next to the desired tag, enter another value, and click **OK**.
- Deleting a tag: On the **Resource Tags** tab, click  next to the desired tag, and click **OK**.

----End

10 Crontask

10.1 Creating a Crontask

Prerequisites

A PerfTest or JMeter project has been created, and test tasks are included.

 **NOTE**

The extensible test project does not support the scheduled pressure test function.

Procedure

- Step 1** Log in to the CodeArts PerfTest console. In the navigation pane, choose **Scheduled Test**. Then click **Create**.
- Step 2** Set the basic information by referring to [Table 10-1](#).

Table 10-1 Creating a crontask

Parameter	Description
Related Project	Select an existing PerfTest or JMeter project.
Related Task	Select the task for which you want to perform a scheduled pressure test. NOTE A task can be associated with only one scheduled pressure test task to be executed.
Crontask Name	Name of the crontask
Description	Description of the crontask

Parameter	Description
Resource Group	Select a test resource group for the crontask. <ul style="list-style-type: none">You can select a shared resource group or an existing private resource group for a PerfTest project. Shared resource groups of a user can be bound to only one pressure test task.Only existing private resource groups can be selected for a JMeter test project.
Enterprise Project	Select the corresponding enterprise project. NOTE This option is unavailable for personal accounts.
Run Frequency	Frequency of running the crontask. Option: Run only once and Run regularly . <ul style="list-style-type: none">Run only once: The task is executed only once at the time set in Date of running.Run regularly: The period settings can be Monthly, Weekly, and Advanced.<ul style="list-style-type: none">Monthly: The task is executed every month on the specified running date (one day or multiple days) and time.Weekly: The task is executed every week on the specified running date (one day or multiple days) and time.Advanced: The task is repeatedly executed based on crontab expressions. Use this mode if you are familiar with crontab expressions. The rules for entering running dates are as follows: Minute: integers ranging from 0 to 59, and operators *, - / Hour: integers ranging from 0 to 23, and operators *, - / Day (days in a month): integers ranging from 1 to 31, and the operator ? *, - / Month: Only * (monthly) is supported. Week (days in a week): any days in a week and the question mark (?) operator NOTE The day and week rules are mutually exclusive. When a non-default value is set for one parameter, the default value (?) is automatically set for the other parameter.
Deadline	The task will not be executed after the deadline. The furthest deadline is one year later. NOTE Set this parameter only when Run Frequency is set to Run regularly .

Step 3 When the configuration is complete, click **Sure**.

----End

10.2 Managing a Crontask

After a crontask is created, you can view, stop, and delete it.

Viewing a Crontask


You can view created crontasks.

- Step 1** Log in to the CodeArts PerfTest console and choose **Scheduled Test** in the left navigation pane.
- Step 2** You can view the overview of crontasks in the calendar on the left of the page. By default, crontasks for the day are displayed on the right of the page. Switch to the other tab to view all crontasks. They can be sorted by the next running time and filtered by the name or status.

----End


Stopping a Crontask

After a crontask is started at the configured running time, you can stop the test if the pressure test plan changes.

- Step 1** Log in to the CodeArts PerfTest console and choose **Scheduled Test** in the left navigation pane.
- Step 2** On the **Crontask for the day** tab page in the right pane, click the crontask name to be stopped.
- Step 3** On the **Crontask Detail** page, click  in the upper right corner.
- Step 4** In the dialog box that is displayed, click **OK** to stop the crontask.

----End

Deleting a Crontask

- Step 1** Log in to the CodeArts PerfTest console and choose **Scheduled Test** in the left navigation pane.
- Step 2** On the **Crontask for the day** or **All crontask** tab page in the right pane, find the crontask to be deleted.
- Step 3** Click  in the **Operation** column of the crontask to be deleted. In the displayed dialog box, click **OK** to delete it.

----End

11 Configuring SLAs

A Service Level Agreement (SLA) defines criteria for performance metrics. You can gain a better understanding of the performance of your service by checking if the metrics meet the criteria specified in the SLA.

Only PerfTest projects support SLAs. JMeter projects do not support. When the SLA metric thresholds are met, alarms will be generated or the performance tests will be stopped.

You can create SLAs in either of the following ways:

- **Creating and Importing an SLA Template.** SLA templates can be imported for any PerfTest test tasks.
- **Create an SLA for a specific test case.** The SLA is applicable only to this test case.

Creating and Importing an SLA Template

Step 1 Log in to the CodeArts PerfTest console, choose **SLAs** in the navigation pane, and click **Create SLA Template**.

NOTE

You can create up to 20 SLA templates in your account in each region.

Step 2 Set the SLA basic information.

- **SLA Name:** This is used to identify an SLA. The SLA name must be unique in a region.
- **SLA Description:** This parameter is optional.

Step 3 Click **Add Rule**. On the **Create Rule** dialog box that is displayed, set the SLA rule by referring to **Table 11-1** and click **Confirm**.

Table 11-1 SLA rule parameters

Parameter	Description
Rule Name	This is used to identify a rule.

Parameter	Description
Metric	Three types of metrics are supported. CodeArts PerfTest will generate an alarm or stop a performance test when the metric reaches the specified threshold. <ul style="list-style-type: none">• Average response time (RT)• Requests per second (RPS)• Success rate
Thresholds	Thresholds for triggering an alarm or stopping a performance test. For example, if you set the metric threshold for Stop Test to < 100 and 3 consecutive times , the test task will be stopped when the metric exceeds 100 for three consecutive times.

Figure 11-1 Creating SLA rules in a template**Create Rule**

Rule name

Metric

Thresholds

Step 4 (Optional) To create multiple rules, repeat **Step 3**.

NOTE

Up to five rules can be created for each SLA.

Step 5 Click **Save** to complete the SLA template creation.

You can view the created SLA template on the **SLAs** page.

NOTE

To edit or delete an SLA template, locate the row that contains the SLA template, and click **Edit** or **Delete**.

Step 6 In the navigation pane, choose **PerfTest Projects**.

Step 7 Click the name of the desired project to go to the details page.

Step 8 In the **Case List** on the left, select the case for which the SLA rule is to be configured.

Step 9 Click the **SLA Configuration** tab, click the rectangle on the right of **Import SLA Template**, select the name of the created SLA template from the drop-down list, and import the SLA template.

Step 10 In the **SLA Rules** area, select one or more requests to which the SLA rules will be applied.

You can also edit, delete, or add rules as required.

Step 11 Click **Save** to finish the SLA configuration.

----End

Creating SLA Rules for a Test Case

Step 1 Log in to the CodeArts PerfTest console and choose **PerfTest Projects** in the left navigation pane.

Step 2 Click the name of the desired project to go to the details page.

Step 3 In the **Case List** on the left, select the case for which the SLA rule is to be configured.

Step 4 Click the **SLA Configuration** tab.

Step 5 Click **Add Rule**. On the **Create Rule** dialog box that is displayed, set the SLA rule by referring to [Table 11-2](#) and click **Confirm**.

Table 11-2 SLA rule parameters

Parameter	Description
Rule Name	This is used to identify a rule.
Metric	Three types of metrics are supported. CodeArts PerfTest will generate an alarm or stop a performance test when the metric reaches the specified threshold. <ul style="list-style-type: none">• Average response time (RT)• Requests per second (RPS)• Success rate
Thresholds	Thresholds for triggering an alarm or stopping a performance test. For example, if you set the metric threshold for Stop Test to < 100 and 3 consecutive times , the test task will be stopped when the metric exceeds 100 for three consecutive times.

Step 6 In the **Target Requests** column, select one or more requests for each rule.

Step 7 (Optional) To create multiple rules, repeat **6** to **7**.

NOTE

Up to five rules can be created for each SLA.

Step 8 Click **Save** to finish the SLA configuration.

----End

12 Auditing

12.1 CodeArts PerfTest Operations That Can Be Recorded by CTS

Key Operations Recorded by CTS

Cloud Trace Service (CTS) is a log audit service of the Huawei Cloud security solution. It records operations on cloud resources in your account, including collection, storage, and searching. You can use logs to perform security analysis, track resource changes, audit compliance, and locate faults.

Through CTS, you can record operations associated with Simple Message Notification (SMN) for later query, auditing, and backtrack operations.

Table 12-1 Key operations recorded by CTS

service_type	resource_type	resource_id	trace_name	Description
CPTS	cptsProject	Database primary key	createCptsProject	Creating a project
CPTS	cptsProject	Database primary key	deleteCptsProject	Deleting a project
CPTS	cptsProject	Database primary key	modifyCptsProject	Modifying a project
CPTS	cptsTask	Database primary key	createCptsTask	Creating a task
CPTS	cptsTask	Database primary key	deleteCptsTask	Deleting a task
CPTS	cptsTask	Database primary key	modifyCptsTask	Modifying a task

service_type	resource_type	resource_id	trace_name	Description
CPTS	cptsTemp	Database primary key	createCptsTemp	Creating a transaction
CPTS	cptsTemp	Database primary key	deleteCptsTemp	Deleting a transaction
CPTS	cptsTemp	Database primary key	modifyCptsTemp	Modifying a transaction
CPTS	cptsCase	Database primary key	createCptsCase	Creating a test case
CPTS	cptsCase	Database primary key	deleteCptsCase	Deleting a test case
CPTS	cptsCase	Database primary key	modifyCptsCase	Editing a test case
CPTS	cptsVariable	Database primary key	setVaribale	Creating a variable
CPTS	cptsVariable	Database primary key	updateVaribale	Modifying a variable
CPTS	cptsVariable	Database primary key	deleteVaribale	Deleting a variable
CPTS	cptsTask	Database primary key	tempDebug	Debugging a task case
CPTS	cptsTaskStatus	Database primary key	updateTaskStatus	Updating task status

12.2 Querying Real-Time Traces

Scenarios

After you enable CTS and the management tracker is created, CTS starts recording operations on cloud resources. After a data tracker is created, the system starts recording operations on data in OBS buckets. CTS stores operation records generated in the last seven days.



This section describes how to query and export operation records of the last seven days on the CTS console.




- [Viewing Real-Time Traces in the Trace List of the New Edition](#)
- [Viewing Real-Time Traces in the Trace List of the Old Edition](#)

Constraints




- Traces of a single account can be viewed on the CTS console. Multi-account traces can be viewed only on the **Trace List** page of each account, or in the OBS bucket or the **CTS/system** log stream configured for the management tracker with the organization function enabled.
- You can only query operation records of the last seven days on the CTS console. To store operation records for more than seven days, you must configure an OBS bucket to transfer records to it. Otherwise, you cannot query the operation records generated seven days ago.
- After performing operations on the cloud, you can query management traces on the CTS console 1 minute later and query data traces on the CTS console 5 minutes later.

Viewing Real-Time Traces in the Trace List of the New Edition

1. Log in to the management console.
2. Click  in the upper left corner and choose **Management & Governance** **Management & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. On the **Trace List** page, use advanced search to query traces. You can combine one or more filters.
 - **Trace Name**: Enter a trace name.
 - **Trace ID**: Enter a trace ID.
 - **Resource Name**: Enter a resource name. If the cloud resource involved in the trace does not have a resource name or the corresponding API operation does not involve the resource name parameter, leave this field empty.
 - **Resource ID**: Enter a resource ID. Leave this field empty if the resource has no resource ID or if resource creation failed.
 - **Trace Source**: Select a cloud service name from the drop-down list.
 - **Resource Type**: Select a resource type from the drop-down list.
 - **Operator**: Select one or more operators from the drop-down list.
 - **Trace Status**: Select **normal**, **warning**, or **incident**.
 - **normal**: The operation succeeded.
 - **warning**: The operation failed.
 - **incident**: The operation caused a fault that is more serious than the operation failure, for example, causing other faults.
 - Time range: Select **Last 1 hour**, **Last 1 day**, or **Last 1 week**, or specify a custom time range.
5. On the **Trace List** page, you can also export and refresh the trace list, and customize the list display settings.
 - Enter any keyword in the search box and click  to filter desired traces.

- Click **Export** to export all traces in the query result as an .xlsx file. The file can contain up to 5000 records.
 - Click  to view the latest information about traces.
 - Click  to customize the information to be displayed in the trace list. If **Auto wrapping** is enabled () , excess text will move down to the next line; otherwise, the text will be truncated. By default, this function is disabled.
6. For details about key fields in the trace structure, see [Trace Structure](#) section "Trace References" > "Trace Structure" and [Example Traces](#) section "Trace References" > "Example Traces".
 7. (Optional) On the **Trace List** page of the new edition, click **Go to Old Edition** in the upper right corner to switch to the **Trace List** page of the old edition.

Viewing Real-Time Traces in the Trace List of the Old Edition

1. Log in to the management console.
2. Click  in the upper left corner and choose **Management & Governance** **Management & Deployment** > **Cloud Trace Service**. The CTS console is displayed.
3. Choose **Trace List** in the navigation pane on the left.
4. Each time you log in to the CTS console, the new edition is displayed by default. Click **Go to Old Edition** in the upper right corner to switch to the trace list of the old edition.
5. Set filters to search for your desired traces. The following filters are available:
 - **Trace Type, Trace Source, Resource Type, and Search By:** Select a filter from the drop-down list.
 - If you select **Resource ID** for **Search By**, specify a resource ID.
 - If you select **Trace name** for **Search By**, specify a trace name.
 - If you select **Resource name** for **Search By**, specify a resource name.
 - **Operator:** Select a user.
 - **Trace Status:** Select **All trace statuses, Normal, Warning, or Incident**.
 - Time range: You can query traces generated during any time range in the last seven days.
 - Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.
6. Click **Query**.
7. On the **Trace List** page, you can also export and refresh the trace list.
 - Click **Export** to export all traces in the query result as a CSV file. The file can contain up to 5000 records.
 - Click  to view the latest information about traces.
8. Click  on the left of a trace to expand its details.

Trace Name	Resource Type	Trace Source	Resource ID	Resource Name	Trace Status	Operator	Operation Time	Operation
createDockerConfig	dockerlogincmd	SWR	--	dockerlogincmd	normal		Nov 16, 2023 10:54:04 GMT+08:00	View Trace

request	
trace_id	
code	200
trace_name	createDockerConfig
resource_type	dockerlogincmd
trace_rating	normal
api_version	
message	createDockerConfig, Method: POST Url=/v2/manage/utils/secret, Reason:
source_ip	
domain_id	
trace_type	ApiCall

9. Click **View Trace** in the **Operation** column. The trace details are displayed.

View Trace ×

```
{
  "request": "",
  "trace_id": "",
  "code": "200",
  "trace_name": "createDockerConfig",
  "resource_type": "dockerlogincmd",
  "trace_rating": "normal",
  "api_version": "",
  "message": "createDockerConfig, Method: POST Url=/v2/manage/utils/secret, Reason:",
  "source_ip": "",
  "domain_id": "",
  "trace_type": "ApiCall",
  "service_type": "SWR",
  "event_type": "system",
  "project_id": "",
  "response": "",
  "resource_id": "",
  "tracker_name": "system",
  "time": "Nov 16, 2023 10:54:04 GMT+08:00",
  "resource_name": "dockerlogincmd",
  "user": {
    "domain": {
      "name": "",
      "id": ""
    }
  }
}
```

10. For details about key fields in the trace structure, see [Trace Structure](#) section "Trace References" > "Trace Structure" and [Example Traces](#) section "Trace References" > "Example Traces".
11. (Optional) On the **Trace List** page of the old edition, click **New Edition** in the upper right corner to switch to the **Trace List** page of the new edition.

13 Reference

13.1 Header Description

Table 13-1 Header Description

Protocol Header Name	Description
Accept	Acceptable types of response contents.
Accept-Charset	Acceptable character sets.
Accept-Datetime	Acceptable versions displayed by time.
Accept-Encoding	Acceptable encoding methods.
Accept-Language	Acceptable natural languages of response contents.
Authorization	Information used for HTTP authentication.
Cache-Control	Instructions that must be followed by all cache mechanisms for a request or response chain.
Connection	Priority connection types for a browser.
Content-Length	The length of the request body represented by 8-byte arrays.
Content-MD5	The binary MD5 hash value of the content of the request body, which is encoded using Base64.
Content-Type	Multimedia types of the request body (used in POST and PUT requests).
Cookie	An HTTP cookie sent by servers through the Set-Cookie.
Date	The date and time when a message is sent.

Protocol Header Name	Description
DNT	A request for a web application to stop tracking a user. In the Firefox browser, it is equivalent to the X-Do-Not-Track protocol header field (supported for Firefox/4.0 Beta 11 or later). Safari and Internet Explorer 9 also support this field.
Expect	Specific actions required by a client for a server.
Front-End-Https	Refers to non-standard header fields used by Microsoft servers and load balancers.
From	The email address of the user who initiates the request.
Host	Domain name of the server (used for virtual host), and port number of the transmission control protocol listened to by the server. If the requested port is the standard port of the corresponding service, the port number can be omitted. This field is mandatory since HTTP/1.1. If the domain name in the URL is an IP address, this field is automatically added. Otherwise, enter the IP address and port number of the tested application in this field.
If-Match	The corresponding operation is performed only when the entity provided by the client matches the entity on the server. It is used in a method such as PUT to update a resource which has not been modified since the last update.
If-Modified-Since	Returning of 304 Not Modified is allowed when the corresponding content is not modified.
If-None-Match	Returning of 304 Not Modified is allowed when the corresponding content is not modified. Refer to the HTTP entity tag. In a typical use, when a URL is requested, the web server returns the resource and its corresponding Etag value, which is placed in the Etag field of the HTTP. The client can then decide whether to cache the resource and its Etag . If the client wants to request the same URL again, it sends a request that contains the saved Etag and the If-None-Match field.
If-Range	If an entity is not modified, send one or more parts that are missing to the sender. Otherwise, send the entire new entity.
If-Unmodified-Since	A response is sent only when the entity has not been modified since a specific time.
Max-Forwards	Refers to the number of times a message can be forwarded by the proxy and gateway.
Origin	Refers to a sharing request initiated for resources of different origins. The server is required to add an Access-Control-Allow-Origin field to the response.

Protocol Header Name	Description
Pragma	Related to specific implementations and may produce multiple effects at any time in the request or response chain.
Proxy-Authorization	Information used to authenticate a proxy.
Proxy-Connection	Derives from the errors in the implementation of an early HTTP version. The function of this field is the same as that of the standard Connection field.
Proxy-Password	Proxy password.
Proxy-Server	Proxy service.
Proxy-Username	Proxy username.
Range	Requests only a part of an entity with the byte offset starting from 0.
Referer	Refers to the previous page accessed by a browser. A link on this page brings the browser to the currently requested page.
TE	Refers to the transmission coding mode expected by a browser. You can use a value of Transfer-Encoding in the response protocol header. In addition, the value trailers (related to the block transmission mode) indicates that the browser expects to receive additional fields if the size of the last block is 0.
Upgrade	The server needs to be upgraded to another protocol.
User-Agent	Refers to the character string of the browser identity.
Via	Refers to request-sending agents informed to a server.
Warning	Refers to a general warning indicating that errors may exist in the body of an entity.
X-Wap-Profile	Refers to an XML file linked to the Internet. The file describes devices being connected.
X-Requested-With	Used to identify Ajax and XML requests. Most JavaScript frameworks send this field and set its value to XMLHttpRequest .
X-Http-Method-Override	A web application is requested to use the method (usually PUT or DELETE) specified in the protocol header field to override the method specified in the request (usually POST). Use this method when a browser or firewall prevents direct sending of the PUT or DELETE method (may be caused by a vulnerability in the software, which needs to be fixed, or because a configuration option is required, in which case the situation should not be bypassed).

Protocol Header Name	Description
X-Forwarded-Proto	Refers to a fact standard used to identify the protocol used at the beginning of an HTTP request.
X-Forwarded-Host	Refers to a fact standard used to identify the host request header originally sent by the client.
X-Forwarded-For	Refers to a fact standard used to identify the original Internet address of a client that is connected to a web server through an HTTP agent or load balancer.
X-Csrft-Token	The X-CSRFToken or X-XSRF-TOKEN header is used to prevent cross-site request forgery.
X-ATT-DeviceId	Enables the server to easily interpret the common device models and firmware information in the User-Agent field of the AT&T device.

13.2 Regular Expression Metacharacters

Table 13-2 Metacharacter description

Metacharacter	Description
.	Matches any characters except <code>\n</code> . If <code>\n</code> needs to be included, use other modes such as <code>[\s\S]</code> .
^	Matches the start position of an input character string and does not match any characters. Use <code>\^</code> to match the character itself.
\$	Matches the end position of an input character string and does not match any characters. Use <code>\\$</code> to match the character itself.
*	The preceding characters or sub-expressions are matched zero or more times. <code>*</code> is equivalent to <code>{0,}</code> . For example, <code>\^*b</code> can match <code>b</code> , <code>^b</code> , <code>^^b</code> , and so on.
+	Matches preceding characters or sub-expressions one or more times, equivalent to <code>{1,}</code> . For example, <code>a+b</code> can match <code>ab</code> , <code>aab</code> , <code>abb</code> , <code>aaab</code> , and so on.
?	Matches preceding characters or sub-expression zero times or once, equivalent to <code>{0,1}</code> . For example, <code>a[cd]?</code> can match <code>a</code> , <code>ac</code> , and <code>ad</code> . When this character follows any other qualifier such as <code>*</code> , <code>+</code> , <code>?</code> , <code>{n}</code> , <code>{n,}</code> , or <code>{n,m}</code> , the matching mode is non-greedy . Non-greedy mode matches the shortest possible searched character strings, and the default greedy mode matches the longest possible searched character strings. For example, the character string <code>oooo</code> , <code>o+?</code> matches only a single <code>o</code> , while <code>o+</code> matches all <code>o</code> .

Metacharacter	Description
	The logic or is performed on two matching conditions. For example, the regular expression (him her) matches it belongs to him and it belongs to her , but cannot match it belongs to them .
\	Marks the next character as a special character, text, reverse reference, or octal escape character. For example, n matches the character n , \n matches the newline character, \\ matches \ , and \(matches (.
\w	Matches a letter, digit, or underscore (_).
\W	Matches any character that is not a letter, digit, or underscore (_).
\s	Matches any blank character, such as a space, tab character, or form feed. It is equivalent to [\f\n\r\t\v] .
\S	Matches any character except blank characters and is equivalent to [^\f\n\r\t\v] .
\d	Matches any digit and is equivalent to [0-9] .
\D	Matches any non-digital character and is equivalent to [^0-9] .
\b	Matches a word boundary (the position between a word and a space) and does not match any characters. For example, er\b matches er in never but does not match er in verb .
\B	A non-word boundary match. For example, er\B matches er in verb but does not match er in never .
\f	Matches a form feed and is equivalent to \x0c and \cL .
\n	Matches a linefeed and is equivalent to \x0a and \cJ .
\r	Matches a carriage return character and is equivalent to \x0d and \cM .
\t	Matches a tab character and is equivalent to \x09 and \cI .
\v	Matches a vertical tab character and is equivalent to \x0b and \cK .
\cx	Matches control characters indicated by x . For example, if \cM matches Control-M or a carriage return character, the value of x must be between A-Z or a-z . Otherwise, the c character indicates c itself.
{n}	The value n is a non-negative integer and refers to the number of matching times. For example, o{2} does not match o in Bob , but matches o in food .

Metacharacter	Description
{n,}	The value n is a non-negative integer and refers to the minimum number of matching times. For example, o{2,} does not match o in Bob but matches all o in foooooo . o{1,} is equivalent to o+ , and o{0,} is equivalent to o* .
{n,m}	The values n and m ($n \leq m$) are non-negative integers, where n refers to the minimum number of matching times and m refers to the maximum matching times. For example, o{1,3} matches the first three os in foooooo , and o{0,1} is equivalent to o? . Note that a space cannot be inserted between commas and digits. For example, ba{1,3} can match ba , baa , or baaa .
x y	Matches x or y . For example, z food matches z or food , and (z f)ood matches zood or food .
[xyz]	Refers to a character set that matches any characters included. For example, [abc] matches a in plain .
[^xyz]	Refers to a reverse character set that matches any characters except xyz . For example, [^abc] matches p in plain .
[a-z]	Refers to a character range and matches any characters in the specified range. For example, [a-z] matches any lowercase letters from a to z .
[^a-z]	Refers to a reverse character range and matches any characters not in the specified range. For example, [^a-z] does not match any characters from a to z .
()	Defines expressions between (and) as group and save characters that match the expression to a temporary area. A maximum of nine characters can be saved in a regular expression, and these characters can be referenced by symbols \1 to \9 .
(pattern)	Matches pattern and captures sub-expressions of the match. You can use the \$0-\$9 attribute to retrieve captured matches from the result match set.
(?:pattern)	Matches pattern but does not capture sub-expressions of the match. That is, it is a non-capture match and does not store matches for future use. This is useful for the or character combined with (). For example, industr(?:y ies) is a simpler expression than industry industries .

Metacharacter	Description
(?=pattern)	Refers to a non-capture match and indicates a forward positive pre-check, searching character strings at the start position of any character strings that match pattern. There is no need to capture the match for future use. For example, "Windows(?=95 98 NT 2000)" matches "Windows" in "Windows2000", but does not match "Windows" in "Windows3.1". A pre-check does not consume characters. That is, after a match occurs, the next search starts immediately, instead of starting from pre-checked characters.
(?!pattern)	Refers to a non-capture match and indicates a forward negative pre-check, searching character strings at the start position of any character strings that do not match pattern. There is no need to capture the match for future use. For example, "Windows(?!95 98 NT 2000)" matches "Windows" in "Windows3.1", but does not match "Windows" in "Windows2000".

 NOTE

To match special characters, add \ before the special characters. For example, to match the following special characters: ^, \$, (), [], {}, ., ?, +, *, and |, use ^\, \\$, \ (, \), \ [, \], \{, \}, \., \?, \+, *, and \|.

13.3 Modifying an Exported Project File

Table 13-3 Request composition of think time

Parameter	Description
name	Can be modified.
t	<p>*Duration (ms)</p> <p>Think Time refers to the waiting time between two consecutive operations performed by a user, such as the interval between login and search.</p> <p>Suppose that the response time for each running of a transaction is 0.5s.</p> <ul style="list-style-type: none">To execute two transaction requests per second, do not set the Think Time request composition.To execute only one transaction request per second, set Think Time to 1s. If the think time is set to 1s and the response time is longer than 1s, the think time will not take effect, and pressure test requests are sent according to the response time.

Table 13-4 Request composition of packet

Parameter	Description
name	Can be modified.
http_version	Indicates the protocol type. HTTP, HTTPS, TCP, and UDP protocols are supported.
When Protocol Type is HTTP or HTTPS , set the following parameters.	
method	GET, POST, PATCH, PUT, and DELETE are supported.
return_timeout	Indicates the timeout interval for waiting for a response from the server when a request is sent. If this parameter is not set, the default response timeout interval (5,000 ms) is used.
URL	URL for sending a request, for example, http://domain name/path , or http://domain name/path?key1=value1&key2=value2 .
headers	Headers consist of keywords and values and are used to notify servers of clients' requests. For details about headers, see Header Description .
Packet content	The body of an entity contains a data block consisting of random data. Not all packets contain the body of an entity. If global variables have been set or local variables have been set for response extraction, variables can be used in the packet content. During a pressure test, variables in the packet content will be replaced with specified values.
When Protocol Type is TCP , set the following parameters.	
NOTE TCP packets do not support the response extraction and checkpoint functions.	
IP	IP address of the tested server to which requests are sent.
port	Port number of the tested server to which requests are sent.
connect_timeout	Timeout duration for the server's response after a connection is initiated.
return_timeout	Timeout duration for waiting for the server's response after a connection is established.
Connection Settings	<ul style="list-style-type: none">● Repeated use: When a request response is complete, the connection remains and is used to send and receive the next request response.● Close: When a request response is complete, the connection is closed and re-established next time.

Parameter	Description
check_end_type	<p>Used to judge whether a request response has been received.</p> <ul style="list-style-type: none">• Length of returned data: length of the returned data, in bytes. When a response of this length is received, data receiving is complete.• End-of-text character: ending mark of the returned data. When an end-of-text character is received, data receiving is complete. <p>NOTE It is recommended that a unique end-of-text character be set. If there are multiple such characters in a response, when the first character is received, the response is considered to have been received. As a result, the received response data is incomplete.</p>
body	<p>The body of an entity contains a data block consisting of random data. Not all packets contain the body of an entity.</p> <p>Content format: Select character strings or a hexadecimal code stream based on the service request content of the tested server.</p> <p>NOTE A hexadecimal code stream can contain only numbers 0–9 and letters a–f. The total number of characters must be an even number.</p> <p>If global variables have been set or local variables have been set for response extraction, variables can be used in the packet content. During a pressure test, variables in the packet content will be replaced with specified values.</p>

Table 13-5 Request composition of response extraction

Parameter	Description
name	Must be unique. The value of the response extraction is assigned to this variable.
range	<p>Range of the content to be extracted.</p> <ul style="list-style-type: none">• Packet content• Header• URL• Response Code <p>The packet content, header, and URL can be extracted using regular expressions.</p>

Parameter	Description
regexp	<p>A regular expression specifies a type of logical expressions performed on strings. In a regular expression, you use certain predefined strings and a combination of these strings to form a rule string that is used to specify a specific filtering logic.</p> <p>A complete regular expression consists of two types of characters: special characters (also called meta characters) and literal or common text characters (such as letters, numbers, and underscores). For details about meta characters, see Regular Expression Metacharacters.</p> <p>NOTE () indicates extraction. Enclose the content to be extracted in parentheses. The content between each () indicates a sub-expression.</p>
match_index	<p>This parameter indicates the sequence number of the matched content extracted through a regular expression.</p> <p>Value: a positive integer</p> <p>NOTE This parameter cannot be set to 0.</p>
exp_index	<p>This parameter indicates the sequence number of the parsed sub-expression.</p> <p>Value range: natural numbers</p> <p>NOTE Value 0 indicates that the entire regular expression is matched. Value 1 indicates that the first sub-expression of the regular expression is matched, that is, the content extracted by the first "()".</p> <p>After extracting content through Regular Expression and Sequence Number of Matching Item, use Expression Value to obtain the final content.</p>
JSON Key Name	<p>Enter the key name to be obtained.</p> <p>For example, {key:{"key1":"v1","key2":["v2","v3"]}}. If you want to obtain v2, enter key.key2[0].</p>
default	<p>Indicates the value returned when a regular expression match fails.</p>

Table 13-6 Request composition of checkpoint

Parameter	Description
name	Can be modified.
value	This parameter is used to respond to HTTP, HTTPS, TCP and UDP response status codes carried in a packet, including 1XX, 2XX, 3XX, 4XX, and 5XX.
header_checks	Headers of the HTTP, HTTPS, TCP, and UDP.

Parameter	Description
body_checks	This parameter refers to the body parts of HTTP, HTTPS, TCP, MQTT, and UDP, namely the load parts of the requests and responses.

Table 13-7 Adding a test project

Parameter	Description
name	Indicates the test project name.
description	Indicates description of the test project.

Table 13-8 Adding a transaction

Parameter	Description
name	Indicates the transaction name.
contents	Request content. You can add transaction requests under a transaction based on service requirements.

Table 13-9 Adding a transaction

Parameter	Description
issue_num	Number of concurrent users. The number of concurrent users refers to the number of users performing operations on the system at the same time. In CodeArts PerfTest, it is the number of virtual users set when you define test task phases.
name	Indicates the phase name. Set a name for the service scenario, for example, homepage test .
time	Duration (second). The maximum time for performing a pressure test in the current phase.
count	Total number of transmissions. During the running of a task, the transaction is calculated based on the number of running times. When the specified value is met, the performance test of the transaction under the task is terminated.

Table 13-10 Adding a global variable

Parameter	Description
name	Global variable name.
variable	Global variable value.
variable_type NOTE Variable content is transmitted in plain text during the pressure test. Exercise caution when entering the content.	<p>Global variable type.</p> <p>When the variable type is Integer, enter the value range of the variable.</p> <p>A pressure test task reads the value range of the corresponding global variable when it is running. For each virtual user, the variable value is polled. That is, the first virtual user obtains the first value of the variable, and the second virtual user obtains the second value. If the values are exhausted, the next virtual user obtains the first value again.</p> <p>Add multiple variables as required.</p> <p>A pressure test task reads the value range of the corresponding global variable when it is running. For each virtual user, the variable value is polled. That is, the first virtual user obtains the first value of the variable, and the second virtual user obtains the second value. If the values are exhausted, the next virtual user obtains the first value again.</p>

13.4 Mapping Between JMeter and PerfTest Fields

CodeArts PerfTest supports the import and automatic switchover of the JMeter script. [Table 13-11](#) describes the mapping between JMeter and PerfTest fields.

Table 13-11 Mapping between JMeter and PerfTest fields

JMeter Field		PerfTest Field	Parameter Description
Thread group	Name	Transaction Name/ Task Name	The HTTP request and the subsequent HTTP request form a transaction.
	Number of threads	Number of concurrent test tasks	
	Duration	Test task duration	If no valid value is obtained, number of concurrent requests is 1 and the duration is 1 minute by default.

JMeter Field		PerfTest Field	Parameter Description
User-defined variables	Name	Enumeration variable name	Global variable, enumerated type. The name must be unique.
	Value	Enumeration variable value	
User parameters	Name	Enumeration variable name	Global variable, enumerated type. The name must be unique.
	User_1	Enumeration variable value	
	User_2	Enumeration variable value	
	...	Enumeration variable value	
	User_N	Enumeration variable value	
HTTP header manager	Name	Packet header	When the POST, PATCH, PUT, or DELETE request mode does not contain the Content-Type header, the Content-Type header value is set to application/x-www-form-urlencoded by default.
	Value	Header value	
Default HTTP request	Protocol	-	If the HTTP request does not contain a value, the default value of the HTTP request is used. Priority: HTTP request > thread group > test plan.
	Server name or IP address		
	Port number		
	Path		
	Parameter		
	Message body data		
	Response timeout in the advanced options.		

JMeter Field		PerfTest Field	Parameter Description
HTTP request	Protocol	Protocol type	Packet. Ignore request modes except POST, GET, PATCH, PUT, and DELETE. The default value is 5,000 ms when no response timeout value is obtained.
	Protocol, server name or IP address, port number, and path	Request URL	
	Method	Request mode	
	Parameter	Add the content to the URL or packet content as required.	
	Message body data	Packet content	
	Response timeout in the advanced options.	Response timeout	
Regular expression extractor (the response field to be checked)	Body	Packet content	Response extraction. Only the regular expression extractor and fixed timer under the HTTP requests can be imported. If the default value is empty, the reference name is used by default.
	Body (unescaped)	Packet content	
	Body as a Document	Packet content	
	Message header	Header	
	URL	URL	
	Response code	Response code	
	Request Headers. Response information.	Import is not supported. The regular expression extractor is ignored.	
Regular expression extractor	Reference name	Variable name	
	Regular expression	Regular expression	
	Template	Sequence number of matching item	
	Match digits	Expression value	
	Default value	Default value	
Fixed timer	Thread latency	Duration	Think time. If no valid value is read, the default value 1,000 is used.

JMeter Field		PerfTest Field	Parameter Description
Random function	-	Random number in a range	An integer generated randomly within the range entered by a user. For example, <code>\$_Random(-2147483648,2147483647)</code>
JSON extractor	Name of created variables	Variable name	Response extraction. Default extraction range: parameter values in .json format. Only the JSON extractor under the HTTP requests can be imported.
	JSON Path expressions	Key name	
	Default Values	Default value	