CodeArts Check

User Guide

Issue 01

Date 2024-11-29

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


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 Working with CodeArts Check

CodeArts Check is a cloud-based service that check code. With years of experience in automatic static check and enterprise application, CodeArts Check provides rich check services on code style, common quality, cyber security risk, and other elements. It also includes comprehensive check reports, convenient bug handling, and many other efficient, easy-to-use functions for enterprises to effectively improve code quality.

Usage Process

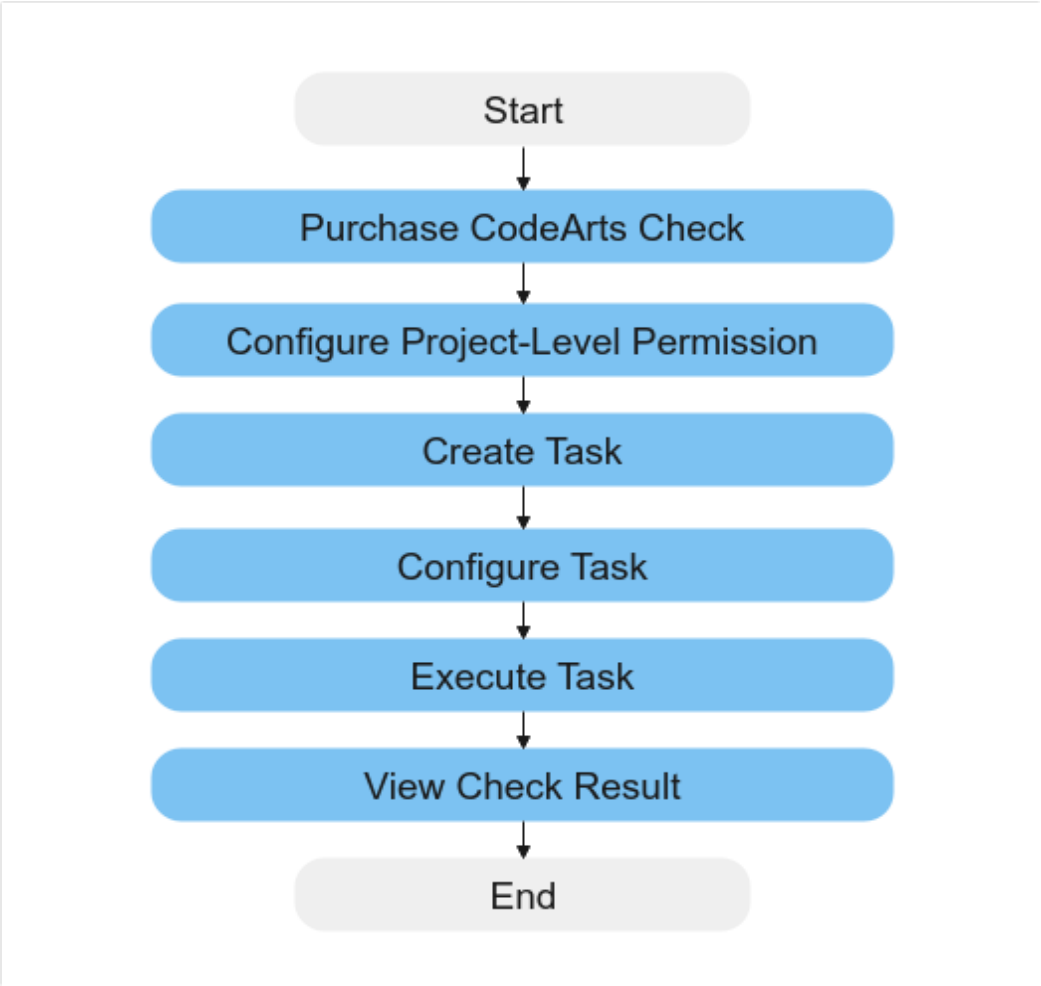


Table 1-1 Process description

Process	Description
Purchasing CodeArts Check	Purchase CodeArts Check.
Configuring Project-level Role Permissions	Access the CodeArts Check homepage and configure project-level permissions.
Creating a Task	Create tasks from different repositories.
Configuring a Task	Modify or configure existing check tasks.
Executing a Task	Execute a check task to identify issues in the source code.

Process	Description
Viewing Check Results	View the check results and fix code issues after a code check task is complete.

2 Purchasing CodeArts Check

2.1 Purchasing CodeArts Check

Prerequisites

You have registered with Huawei Cloud and completed real-name authentication. If you do not have a HUAWEI ID yet, follow these steps to create one:

1. Visit the **Huawei Cloud official website**.
2. Click **Sign Up** and create your account as instructed.

   Once your account is created, the system automatically redirects you to your personal information page.
3. Complete individual or enterprise real-name authentication. For details, see **Real-Name Authentication**.

Purchasing CodeArts Check

Refer to **Purchasing a CodeArts Package**.

3 Configuring Project-level Role Permissions

Assign a role to the new member. Each role comes with its own default permissions. For details, see **Table 3-1**.

Table 3-1 Default role permissions in CodeArts Check

Permission/Role		Project Administrator	Project Manager	Developer	Test Manager	Tester	Participant	Viewer	Product Manager	System Engineer	Committer
Tasks	Create	√	√	√	×	×	×	×	×	√	√
	Execute	√	√	√	×	×	×	×	√	√	√
	View	√	√	√	×	×	×	×	√	√	√
	Edit	√	√	√	×	×	×	×	√	√	√
	Delete	√	√	×	×	×	×	×	√	√	√
Rulesets	Set as default	√	√	×	×	×	×	×	√	√	√
Issues	View	√	√	√	×	×	×	×	√	√	√
	Edit	√	√	√	×	×	×	×	√	√	√

Prerequisites

- You have **purchased CodeArts Check and authorized users**.
- You have added members by referring to ***CodeArts** User Guide* > "Preparations" > "Adding Project Members", and assigned roles to the new members by referring to "Managing Permissions".

Accessing CodeArts Check

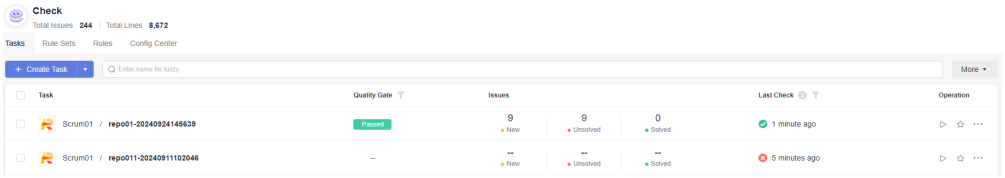
Step 1 **Log in to the Huawei Cloud console**.

Step 2 Click  in the upper left corner and choose **Developer Services** > **CodeArts Check** from the service list.

Step 3 You can access CodeArts Check in either of the following ways:

- **Homepage entry**

  Click **Access Service**. This page displays the check task list of the current user.

- **Project entry**

  a. Click **Access Service**.

  b. On the navigation bar, click **Homepage**.

  c. Click the name of the project to be viewed.

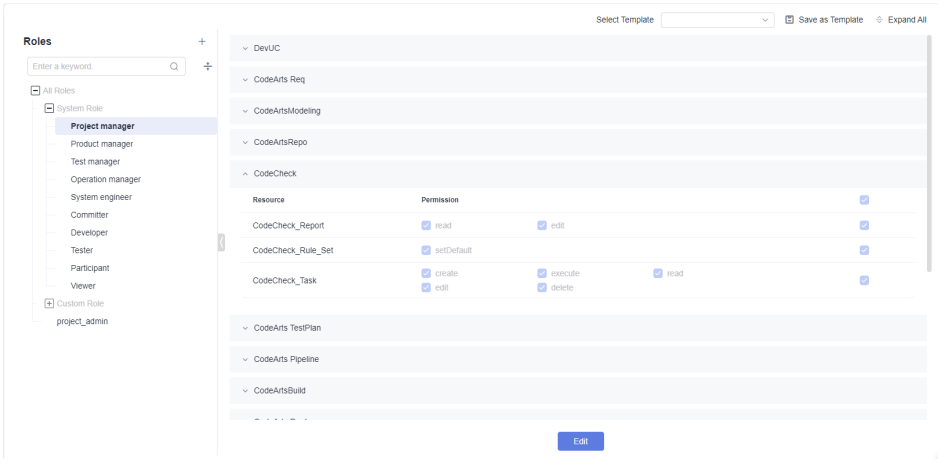
  d. Choose **Code** > **Check**.

Click  in the upper left corner of the page and select a region.

----End

Configuring Project-level Role Permissions

1. **Access CodeArts Check** through a project.
2. In the navigation pane, choose **Settings** > **Permissions**.
3. On the displayed page, configure permissions for different roles in CodeArts Check.



4 Creating a Task

CodeArts Check supports different code repositories. Before task creation, you should:

- Create a project.
- (Optional) **Create a code repository** by referring to *CodeArts Repo User Guide* > "Creating a Repo".
- (Optional) Create a third-party code repository.

Creating a Task to Check Code from CodeArts Repo

If you select **Automatically create Check task** when creating a code repository in CodeArts Repo, a code check task will be created synchronously. You can view the task in the task list of CodeArts Check.

If you do not select **Automatically create Check task** when creating a code repository, perform these steps to create a task:

Step 1 **Access CodeArts Check** through a project.

Step 2 Click **Create Task**, configure parameters by referring to **Table 4-1**, and click **Create Task**.

Table 4-1 Task parameters

Parameter	Description	Mandatory
Project	Project that the task belongs to. • This parameter is set by default when you access CodeArts Check from the project entry. • During access through the service entry, select the project created in **Creating a Project** based on the site requirements.	Yes
Code Source	Select **Repo**.	Yes

Parameter	Description	Mandatory
Name	Customize your task name. Enter 1 to 128 characters: letters, digits, underscores (_), hyphens (-), and periods (.).	Yes
Repository	Select the code repository to be checked.	Yes
Branch	Select the code branch to be checked.	Yes
Language	Select the code language to be checked.	No

NOTE

- To modify the task name and default branch, go to the task details page, choose **Settings** > **Basic Info**, and then modify the target task.
- To delete a task, go to the task details page, choose **Settings** > **Basic Info**, and then delete the target task. Alternatively, click *** in the row where the target task is located, and choose **Delete** from the drop-down list.

  When the task to be deleted has a code repository with multiple branches, the deletion will be successful if there are fewer than 200 branches. Otherwise, the deletion will fail.

----End

Creating a Task to Check Code from Git

Step 1 Access CodeArts Check through a project.

Step 2 Click **Create Task**, configure parameters by referring to **Table 4-2**, and click **Create Task**.

Table 4-2 Task parameters

Parameter	Description	Mandatory
Project	Project that the task belongs to. • This parameter is set by default when you access CodeArts Check from the project entry. • During access through the service entry, select the project created in **Creating a Project** based on the site requirements.	Yes

Parameter	Description	Mandatory
Code Source	Select **Git**. For code hosted on other services, use a Git connection to pull the code.	Yes
Name	Customize your task name. Enter 1 to 128 characters: letters, digits, underscores (_), hyphens (-), and periods (.).	Yes
Endpoint	Select the service endpoint connected to the Git repository. If no, **create one**.	Yes
Repository	Select the code repository to be checked.	Yes
Branch	Select the code branch to be checked.	Yes
Language	Select the code language to be checked.	No

NOTE

- To modify the task name and default branch, go to the task details page, choose **Settings** > **Basic Info**, and then modify the target task.
- To delete a task, go to the task details page, choose **Settings** > **Basic Info**, and then delete the target task. Alternatively, click *** in the row where the target task is located, and choose **Delete** from the drop-down list.

  When the task to be deleted has a code repository with multiple branches, the deletion will be successful if there are fewer than 200 branches. Otherwise, the deletion will fail.

----End

Creating a Service Endpoint

When you select any third-party repository, the **Endpoint** is a mandatory setting.

A service endpoint is an extension to CodeArts and supports connection to third-party services.

By default, CodeArts Check pulls code from CodeArts Repo for your build. CodeArts Build also uses service endpoints to connect to third-party repositories to obtain project source code.

NOTE

- The network may be unstable or other problems may occur when a third-party repository is used.
- Use the code import function of CodeArts Repo for secure, stable, and efficient download and build.

Creating a Git service endpoint

1. Click ▼ and click **Create Endpoint** in the drop-down list.
2. On the displayed page, click **Create Endpoint** and select **Git repository**.
3. In the displayed dialog box, set parameters by referring to **Table 4-3**.

Table 4-3 Git service endpoint parameters

Parameter	Description	Mandatory
Service Endpoint Name	Enter a maximum of 256 characters, including letters, digits, hyphens (-), underscores (_), periods (.), and spaces.	Yes
Git Repository URL	Enter the URL of the Git repository (HTTPS address).	Yes
Username	Enter the username of the Git repository.	No
Password or Access Token	Enter the password used for logging in to the Git repository.	No

4. Click **Confirm**. The page for creating a task is displayed.

5 Configuring a Task

5.1 Configuring a Rule Set

5.1.1 Configuring a Preset Rule Set

A rule set is a set of rules used for code checks, with variations within different sets. These sets are classified based on different objectives, such as security, Android apps, and coding styles.

You can modify the rule sets as needed, and the updated rules will be applied during the next check.

CodeArts Check supports rule sets in multiple languages at different levels. For details about system rule sets, see **Table 5-1**.

NOTICE

Rule sets of multiple languages cannot be used to check a task at the same time. Example: C# rule sets cannot be applied with other language rule sets to check a task.

Table 5-1 Preset rule sets

Language	System Rule Set	Default Rule Set	Extension of Supported File
C++	• Generic Inspection Rule Set	Generic Inspection Rule Set	.c/.cc/.cpp/.cxx/.cp/.c++/.inc/.inl/.mm/.h/.hh/.hpp/.hxx/.h++
Java	• Generic Inspection Rule Set	Generic Inspection Rule Set	.java

Language	System Rule Set	Default Rule Set	Extension of Supported File
Go	• Generic Inspection Rule Set	Generic Inspection Rule Set	.go
Python	• Generic Inspection Rule Set	Generic Inspection Rule Set	.py
JavaScript	• Generic Inspection Rule Set	Generic Inspection Rule Set	.js/.jsx
C#	• Generic Inspection Rule Set	Generic Inspection Rule Set	.cs
TypeScript	• Generic Inspection Rule Set	Generic Inspection Rule Set	.ts/.tsx
CSS	• Generic Inspection Rule Set	Generic Inspection Rule Set	.css
HTML	• Generic Inspection Rule Set	Generic Inspection Rule Set	.html
PHP	• Generic Inspection Rule Set	Generic Inspection Rule Set	.php
LUA	General Criterion Set	General Criterion Set	.lua
RUST	General Criterion Set	General Criterion Set	.rs
Shell	General Criterion Set	General Criterion Set	.sh
KOTLIN	General Criterion Set	General Criterion Set	.kt

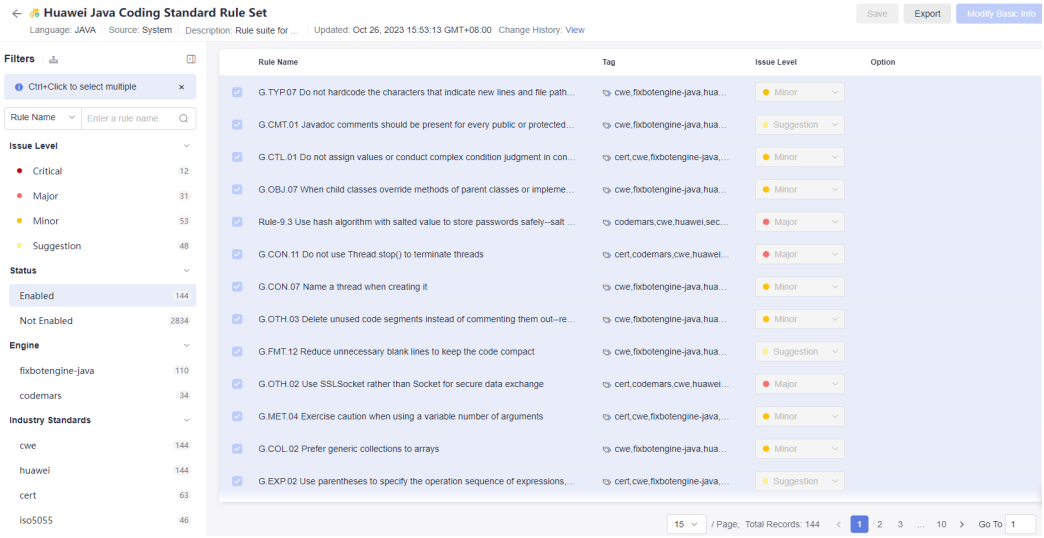
Viewing Rule Details in a Rule Set

View the rule details in rules sets, such as issue levels, compliant and non-compliant examples, and fix suggestions, to select a desired rule set.

Step 1 Access CodeArts Check through a project.

Step 2 Click the **Rule Sets** tab.

Step 3 Click the rule set name to go to the details page.



Alternatively, click **Export** to export the enabled rules in the rule set to the local PC and view the rule details.

----End

Modifying the Default Rule Set

Each language has its default set, as shown in **Table 5-1**. Perform the following operations to modify the default rule set if needed.

NOTE

To view the default rule sets, access CodeArts Check through a project.

Step 1 On the **Rule Sets** tab page, filter the target language in the **Language** column.

Step 2 Click ••• in the row where the rule set locates and select **Default Set** to set the current rule set as default.

----End

Configuring the Rule Set to Be Used

Perform the following operations to use other preset rule sets. Alternatively, customize rule sets if needed. For details, see **Configuring a Custom Rule Set**.

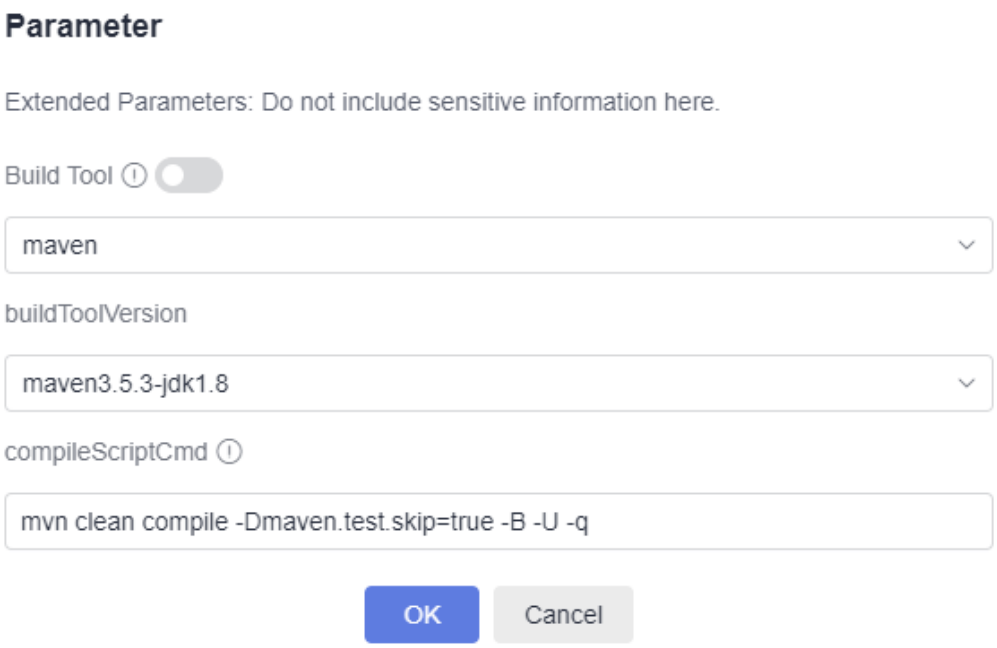
Step 1 Go to the task page. In the task list, click a task name.

Step 2 On the task details page, choose **Settings** > **Rule Sets**.

Step 3 In the **Languages Included** area, enable the switch of target language.

Step 4 In the **Enable Rule Set** area, click ○ to select the rule set to be used.

(Optional) To configure extended parameters for the selected rule set, click **Parameter**, configure extended parameters, enable the configuration function, and click **OK**.



NOTE

To use a code security enhanced package, you need to configure check parameters.

----End

5.1.2 Configuring a Custom Rule Set

CodeArts Check supports custom rule sets. Each rule set must contain at least one rule.

A rule set can be configured with only one language.

NOTE

Purchase the code security check enhancement package as required before using CodeArts Check. This package identifies code security risks and vulnerabilities more comprehensively for Java, C++, Go, and Python.

Customizing a Rule Set

Step 1 Access CodeArts Check through a project.

Step 2 Click the **Rule Sets** tab.

Step 3 Click **Create Rule Set**. In the displayed dialog box, enter a rule set name and description, and select a language.

You can also copy an existing rule set by selecting it from the **Replicated in** drop-down list.

Step 4 Click **OK**.

Step 5 Select rule names, set **Issue Level** and click **Save** in the upper right corner.

Customize rules if needed.

- Click the rule set name to go to the details page.
- Alternatively, on the details page, click **Export** to export the enabled rules in the rule set to the local PC and view the rule details.
- On the details page, click **Modify Basic Info** to modify the rule set name or description.

  NOTE

  - The **Language** cannot be modified.
  - Preset rule sets cannot be modified.
  - Custom rule sets can be deleted only by the one who created them.
- Click ··· in the row where the rule set is located and choose **Delete Set**.

  NOTE

  - Custom rule sets can be deleted only by the one who created them.
  - System rule sets and custom rule sets in use cannot be deleted.
  - To delete a rule set being used by a code check task, you can either delete the task or assign another rule set to the task.

----End

Using a Custom Rule Set

Step 1 Go to the task details page, and choose **Settings** > **Rule Sets**.

Step 2 If any changes are made to the code repository after you create a code check task, click in the **Languages Included** row to re-obtain the target language, and enable the switch of language.

Step 3 Click to select the created custom rule set.

----End

Customizing a Rule

- Custom rules: max. 10.
- Lines of code scanned by a rule set with only custom rules: max. 100,000. This applies when the rule set contains only custom rules.
- Duration per code check task with 100,000 LOC scanned by a rule set with only custom rules: max. 1.5 hours. This applies when the rule set contains only custom rules.

To scan more code or customize more rules, contact technical support.

Step 1 Click the **Rules** tab.

Step 2 Click **Create Rule**. Set parameters by referring to **Table 5-2**.

Table 5-2 Rule parameters

Parameter	Description	Mandatory
Rule Name	Custom rule name. It can be customized. Enter 1 to 128 characters: letters, digits, underscores (_), hyphens (-), and periods (.).	Yes
Tool Rule Name	Rule source code file (by default).	Yes
Tool	Check tool used by a custom rule. Currently, only SecBrella is supported.	Yes
Language	Language checked by a custom rule. Currently, only Java is supported.	Yes
Source Code	Rule source code file. Create a **.kirin** file, create a domain-specific language (DSL) for rules, run the local plug-in to generate a **SecH_*Rule name*.json** file in the **OutputReport** directory, and upload the **.json** file.	Yes
Severity	Severity of a code issue detected by a rule. The value can be **Critical**, **Major**, **Minor**, or **Suggestion**.	Yes
Tag	Rule tag for different scenarios. Use commas (,) to separate multiple tags.	No
Description	Rule description. The content can contain code in Markdown. Max. 10,000 characters.	Yes
Compliant Example	Compliant code example. The content contains code in Markdown. Max. 10,000 characters.	No
Noncompliant Example	Noncompliant code example. The content contains code in Markdown. Max. 10,000 characters.	No
Fix Suggestions	Issue fixing suggestions. The content can contain code in Markdown. Max. 10,000 characters.	No

Step 3 Click **Create Rule**.

----End

5.2 Configuring Quality Gates

Set the threshold based on the quality requirements to check whether the project is ready for production. If the actual check result surpasses the gate threshold, the project is deemed unfit for production.

Quality gate can be set at the tenant, project, and task levels in a descending order of precedence. The navigation paths are as follows:

- Tenant level: Go to the CodeArts Check homepage, and choose **Config Center**. The **Quality Gate** details are displayed by default.
- Project level: Go to the project details page, and choose **Code > Check > Config Center > Quality Gate**.
- Task level: Go to the task details page, and choose **Settings** > **Quality Gate**.

NOTE

Notification settings are supported at the project level and task level.

This section uses a task-level quality gate as an example to describe how to configure a quality gate.

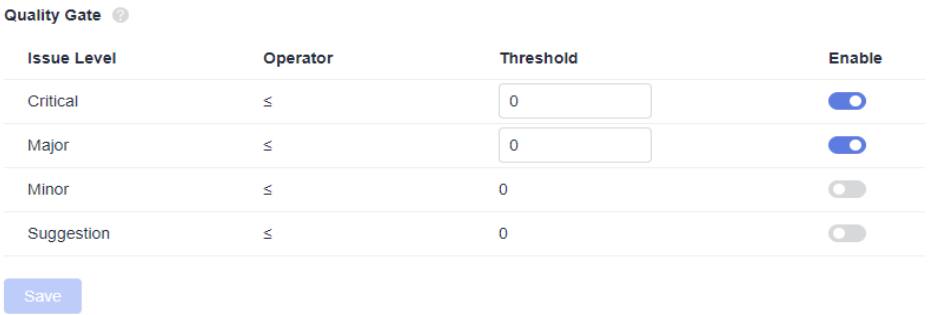
Step 1 Access CodeArts Check through a project.

Step 2 Go to the task page. In the task list, click a task name.

Step 3 On the task details page, choose **Settings** > **Quality Gate**.

Step 4 In the **Quality Gate** area, enable gate items and set thresholds based on quality requirements.

Figure 5-1 Configuring a gate



Step 5 Click **Save**.

You can click **Start Check** and view the check results.

If the issues of a gate item exceed the corresponding thresholds for a gate, **Failed** is displayed in the **Task-level Gate Result** area of the **Overview** tab page. Fix code as prompted.

----End

5.3 Configuring a Scheduled Task

You can set the execution plans of a code check task as required to periodically execute the check task for the default branch.

NOTE

- It is advised to configure this for tasks in non-pipeline mode.
- The interval on the current day must be longer than 5 minutes.

Configuring an Execution Plan

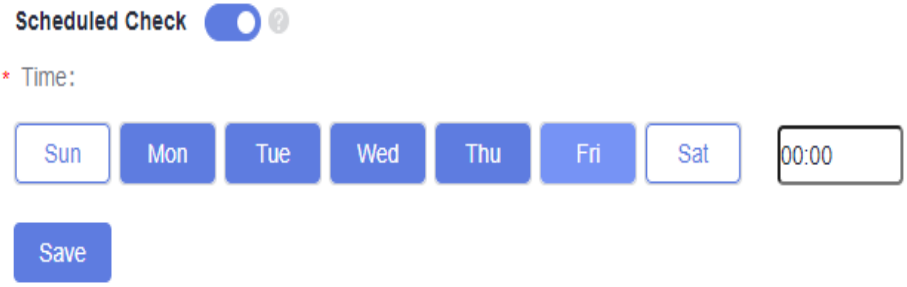
Step 1 Access CodeArts Check through a project.

Step 2 Go to the task page. In the task list, click a task name.

Step 3 On the task details page, choose **Settings** > **Execution Plans**.

Step 4 Enable **Scheduled Check** and set the automatic execution time.

Figure 5-2 Configuring an execution plan



Step 5 Click **Save**.

----End

5.4 Configuring Notifications

You can set notification modes for the code check task event types as required.

For example, when the code gate failed, you can configure the notification modes, recipient roles, and custom members. In this way, a notification is sent to the task executor, configured recipient roles, and custom members. Otherwise, a notification is only sent to the task executor.

The notification can be configured at the project level or task level. The navigation paths are as follows:

- Project level: Go to the project details page, and choose **Code** > **Check** > **Config Center** > **Notifications**.
- Task level: Go to the task details page and choose **Settings** > **Notifications**. This section uses the task-level notification settings as an example to describe how to enable and disable notifications.

Configuring Message Notices

Step 1 Access CodeArts Check through a project.

Step 2 Go to the task page. In the task list, click a task name.

Step 3 On the task details page, choose **Settings** > **Notifications**.

Step 4 Enable or disable the notification of an event type as required.

- If **Notify** is enabled, event notices will be sent to specific members of current project.
- If **Email** is enabled, emails will be sent to specific members of current project.

----End

5.5 Configuring Check Modes

You can set check modes and checked catalogs for efficiency.

NOTICE

Neither C# nor the secbrella engine supports merge requests.

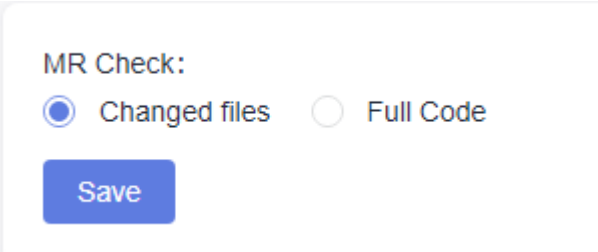
Configuring Check Modes

Step 1 Access CodeArts Check through a project.

Step 2 Go to the task page. In the task list, click a task name.

Step 3 On the task details page, choose **Settings** > **Check Modes**.

Step 4 In the **MR Check** area, select **Changed Files** or **Full Code**. By default, **Changed Files** is selected.



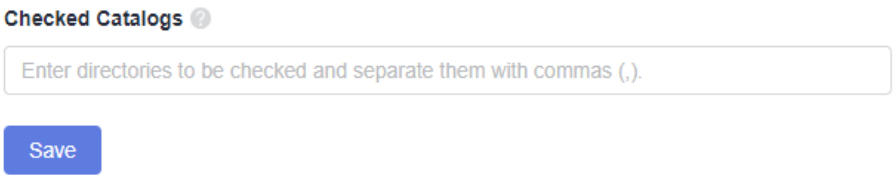
Step 5 Click **Save**.

----End

Configuring Checked Catalogs

You can specify certain directories to be checked in the code repository.

- By default, all code in the code repository is checked.
- If you specify one or more directories, the files in the specified directories are checked.



NOTE

You can specify multiple directories by separating them with commas (,). For example:

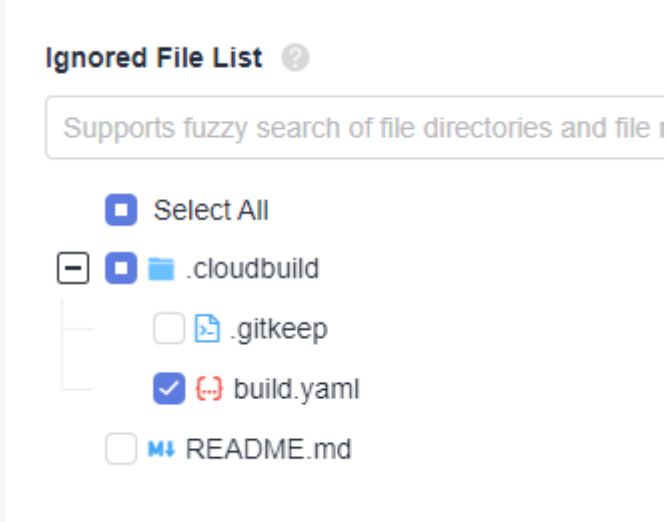
- To check the **dir1** and **dir2** directories, enter **dir1,dir2**.
- To check **dir3** in the **dir1** directory, enter **dir1/dir3**.
- You need to specify directories instead of file names. By default, all code in the code repository is checked.
- The specified directories cannot start with a period (.).
- By default, all directories are checked if you leave the text box empty. If you specify directories, all files within them are checked except for those that have been **set to be ignored**.
- This function takes effect only for the files that do not need to be compiled. If file B is generated after the compilation of file A, issues in file B are not reported.

Setting an Ignored File List

You can configure the file scope by task.

- If you do not specify a file path, all code of the selected repository branch is checked.
- If you specify a file path, the files in it are ignored during a check.

In the **Ignored File List** area, select the file paths that do not need to be checked. Run the code check task. The files in the selected paths will not be checked.



NOTE

This function takes effect for the files that (do not) need to be compiled.

- Files that do not need to be compiled: If these files are set to be ignored, they are not checked.
- Files that need to be compiled: For example, file A and file B depend on each other. If file A is set to be ignored, file A is not checked, but file A is still involved in the compilation process before the check.

5.6 Viewing Health Status

After the check task is complete, you can view the code health status and mark the code in the code repository.

The impact degree of each issue level is as follows:

Critical: 5

Major: 3

Minor: 1

Suggestion: 0.1

The health status levels are as follows (Formula: Health status score = SUM ∑(The number of issues × Issue level)/Code volume):

A: Health score ≤ 0.001

B: 0.001 < Health score ≤ 0.005

C: 0.005 < Health score ≤ 0.01

D: Health score > 0.01

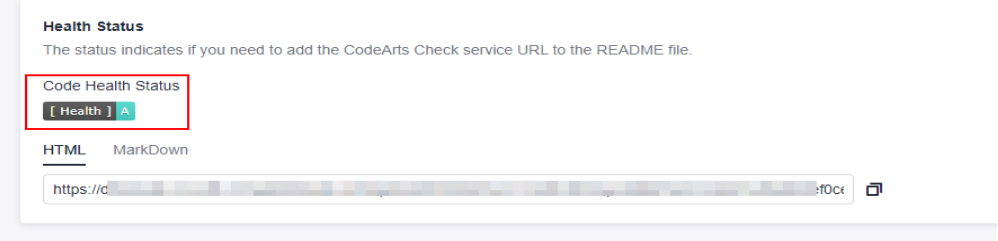
Viewing Health Status

Step 1 Access CodeArts Check through a project.

Step 2 In the task list, click a task name.

Step 3 On the task details page, choose **Settings** > **Health Status**.

HTML and **MarkDown** are supported.



----End

5.7 Configuring Integration Services

If branch merging exists in CodeArts Repo, you can configure an automatic check task and generate a subtask. That is, if branch merging exists in a code repository, a new code check subtask is generated.

Configuring MR Check Status

Step 1 Access CodeArts Check through a project.

Step 2 Go to the task page. In the task list, click a task name.

Step 3 On the task details page, choose **Settings** > **Integration Services** > **CodeArts Repo**.

Step 4 Determine whether to select **MR Status** (MR indicates merge requests). If you select it, the automatic check of code review is enabled.

Step 5 If a branch is merged in the code repository of the task, a code check job is automatically generated.

1. Access the task details page.
2. Click  and create a merge request in CodeArts Repo.

   NOTE

   Based on the **Check Modes** you selected, the changed files or full code will be checked after the branch is merged.

3. Click  . You can view all historical records of MR-triggered code check in the drop-down list.

----End

Configuring an Automatic Check Task Executed upon Code Commit

Step 1 Select **Executed upon code submission**.

NOTE

This function applies only to branches with created tasks.

Step 2 Click **Save**.

----End

5.8 Configuring a Custom Environment

If a common build environment does not have the required dependency packages and tools when you use CodeArts Check, you can use the base image provided by a custom build environment to execute the code check task.

Prerequisite

You have pushed the custom environment to the image repository.

Configuring a Custom Image

Step 1 Access CodeArts Check through a project.

Step 2 Go to the task page. In the task list, click a task name.

Step 3 On the task details page, choose **Settings** > **Custom Environment**.

Step 4 In the **Custom Image** area, enter the content in the format of *Domain name/Organization name/Image repository name:Tag name* or an image link of SoftWare Repository for Container (SWR).

Step 5 Click **Save**.

----End

5.9 Configuring Advanced Configurations

Currently, this function is not available in the AP-Singapore, R-Istanbul, and ME-Riyadh regions.

Configuring the Start Time of New Issues

You can configure the start time of new issues for each task and perform the scanning again after configuration. If the finding time is later than the configured time, the issue is classified as new. If the start time is not configured, gaps between two consecutive check results are considered the number of new issues.

Step 1 Access CodeArts Check through a project.

Step 2 Go to the task page. In the task list, click a task name.

Step 3 On the task details page, choose **Settings** > **Advanced**.

Step 4 In the **New Defects Start** area, click  to configure the start time of the new issues.

NOTE

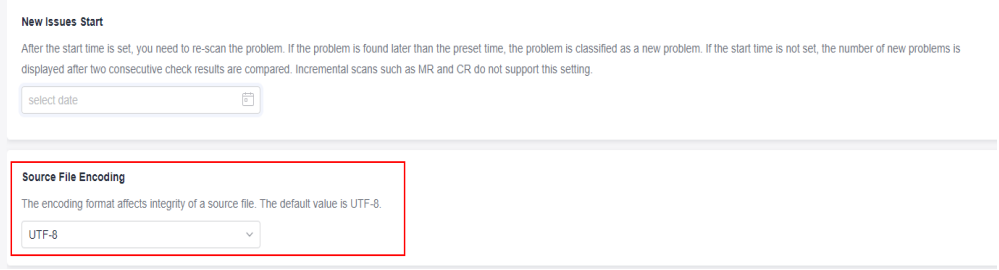
By default, findings since the first successful check are incremental.

----End

Configuring Source File Encoding

You can configure how to encode a file. The source file encoding affects integrity of the source code content identified by the system. The default encoding mode is UTF-8.

Step 1 In the **Source File Encoding** area, select an encode format from the drop-down list.



Step 2 Check whether a message is displayed, indicating that the modification is successful.

----End

6 Executing a Task

You can perform a check task to identify issues in the source code in time.

You can use either of the following methods to check code:

- Manual execution: After creating a task, manually execute it. This section uses it as an example.
- Scheduled execution: For details, see **Configuring a Scheduled Task**.

Prerequisites

- You have **created a task**.

Executing a Task

Step 1 **Access CodeArts Check** through a project.

Step 2 Click ▷ in the row where the task is located and wait until the task is complete.

----End

Checking Code in Other Branches

On the task details page, you can switch between checked and unchecked branches.

Step 1 Click a task name.

Step 2 On the task details page, the **Overview** tab page is displayed by default. Click master ▾ to switch among different branches.

Step 3 After switching to the target branch, click **Start Check** and wait until the task is complete.

Clicking ▷ again will automatically switch the branch to the default branch for checking.

----End

7 Viewing Check Results

Prerequisite

You have **created** and **executed** a task.

Viewing Check Details

Step 1 **Access CodeArts Check** through a project.

Step 2 Go to the **CodeArts Check** page and search for the target task. You can view the status of the check task in the **Issue** column of the code check task list.

- New: new issues that need to be solved after the task is executed.
- Unsolved: issues that have not been solved after the task is executed.
- Solved: issues that have been solved after the task is executed.

Step 3 Click the task name to view the check details.

- Number of lines of code.
- Gate result: The result can be **Passed** or **Failed**, and the numbers of critical issues and major issues are displayed.

  You can set the gate threshold by referring to **Configuring Quality Gates**.
- Number of issues detected.
- Number of delayed rollout issues detected by the upgraded check engines.
- Number of unresolved new issues.
- Number of resolved new issues.
- Average **cyclomatic complexity**.
- Code **duplication rate**.
- Number of non-blank non-comment lines of code.
- Top 10 rules with the most issues and the numbers of these issues.
- Pending issues by severity.
- Issue assignments. If the number of assigned issues does not match the total number of issues, there are some issues that have not been assigned to owners.

- Historical trends. By default, the figures show trends of issues, cyclomatic complexity, and duplicate rates in the last month.

----End

Viewing Code Issues

You can view code issue details in either of the following ways:

- **Exporting issues to the local PC**

  Method 1:

  a. On the task list page, select the tasks for which you want to export issues in batches (max. 20 tasks at a time).

  b. Choose **More** > **Export Task Issues**.

  c. In the displayed dialog box, select the items to be exported and click **Export**.

  d. Click **View Downloads** to view the export progress.

  e. After the export is complete, click **Download** to export the issues to the local PC.

  Method 2:

  a. Click the **Issues** tab.

  b. Choose **Apply to Filtered** > **Export** to export issues to the local PC.

- **Online viewing**

  a. Click the **Issues** tab. In the **Filters** area, set filter criteria to filter issues. The following table lists the filter criteria.

Filter Criteria	Description
Issue Level	The options are **Critical**, **Major**, **Minor**, and **Suggestion**.
Issue Status	The options are **Pending**, **Resolved**, **Ignored**, and **Fixed**.
Latency Issues	Indicates the number of issues in the official version and in the delayed version.
Check Time	Indicates **New** issues and **Inventory** issues distinguished based on the detection time. **NOTE** You can **set the start time of new issues**. Issues found after that time are new.
File	Filter issues by source file name.
Languag e	Filter issues by language.
Rule	Filter issues by rule so that users can solve the same type of issues at a time.

Filter Criteria	Description
User Tag	Filter issues by user tag of rules.
Owner	Filter issues by issue owner. **NOTE** During check, new issues are automatically assigned to the last committer of the faulty code line when you enable **Accurate Matching of Issue Owners**.
CWE	Filter issues by security rule name.

b. After setting the filter criteria, you can perform the following operations in the right pane.

- If an issue has been resolved, click **Pending** and select **Resolved**. If an issue does not need to be handled, click **Pending** and select **Ignore Issue**. To ignore the issue, you need to submit a comment.
- Click **Owner** to transfer the issue to another owner.
- Click **Help** to view the rule details corresponding to the issue.
- Click **View Context**. On the **Context** page, you can modify the issue status, transfer the owner, and click  to modify the code in the repository based on **Help**.
- Click **Create Ticket** to submit the service ticket to the specified owner. This function is available only in Scrum projects.
- Click **Generate Report** to generate a report in PDF format.
- Choose **Apply to Filtered** > **Change Status** to modify the status of all filtered issues.

  NOTE

  If the number of issues is 0, this button is not displayed.
- Choose **Apply to Filtered** > **Change Owner** to transfer all the filtered issues to a specified owner.

Viewing Code Metrics

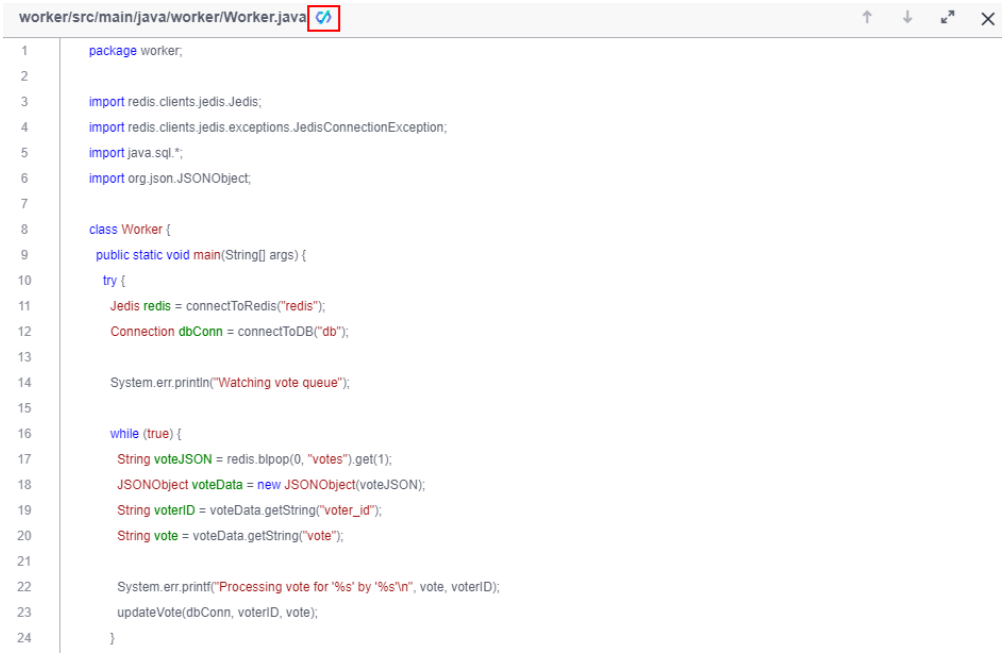
Code metrics reflect the quality of committed code, helping you detect and fix issues in a timely manner and improving R&D efficiency. Code metrics cover the following two types of issues:

- Cyclomatic complexity.

  NOTE

  Code cyclomatic complexity is not supported for code in Shell.

  Click a file name to go to the file details page. Click  to modify the code in the code repository.

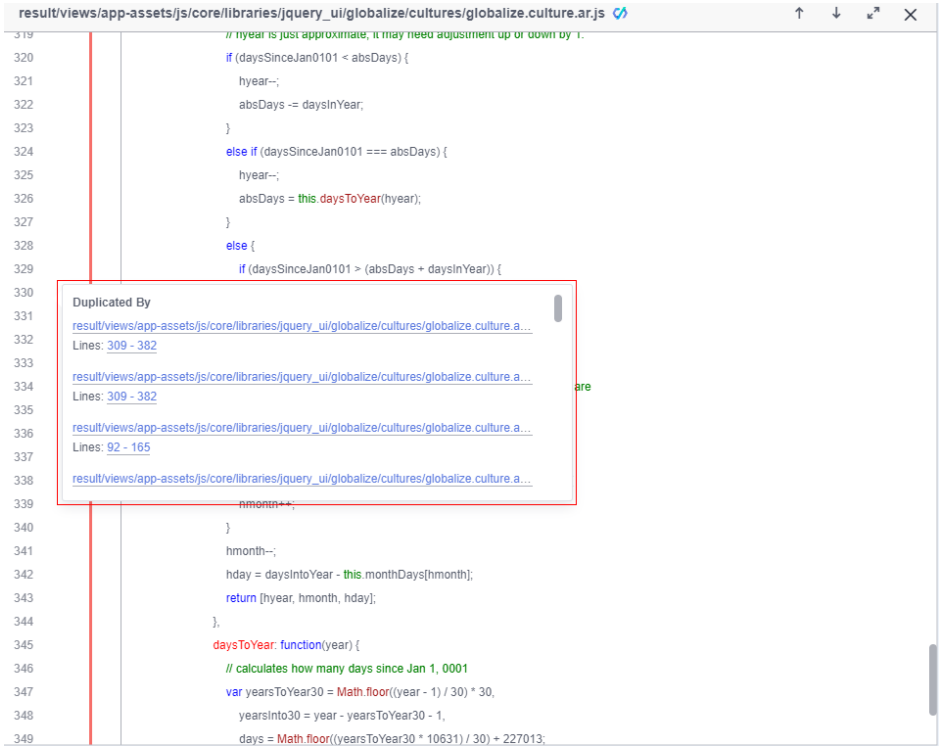


- Duplication rate. CodeArts Check identifies duplicate lines, blocks, and rates.

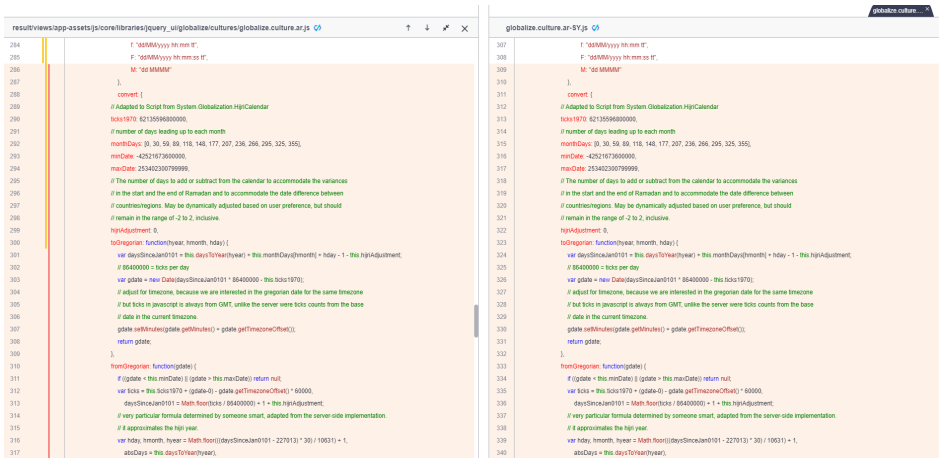
  To solve this problem, perform the following steps:

  a. Click a file name to go to the file details page.

  b. The vertical bars on the left of the file details page identify duplicated code blocks. If you click a vertical bar, the list of files that contain the same code blocks is displayed.



  c. Click a file name to compare the duplicated code blocks of the two files.



d. Click  to modify the code in the code repository.

Viewing Check Logs

Click the **Logs** tab to view the logs and check parameters.

Figure 7-1 Logs

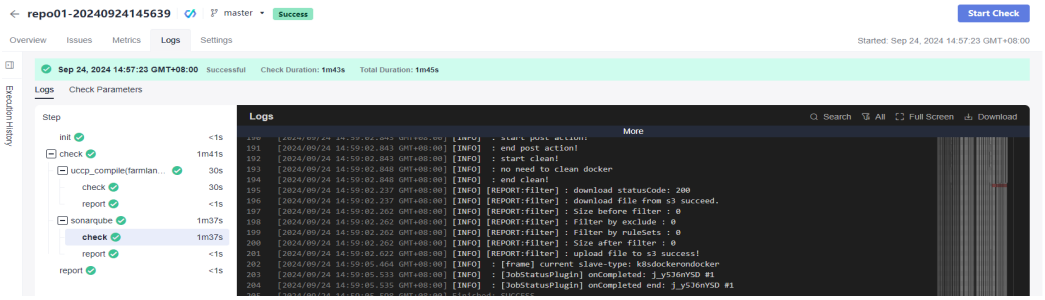
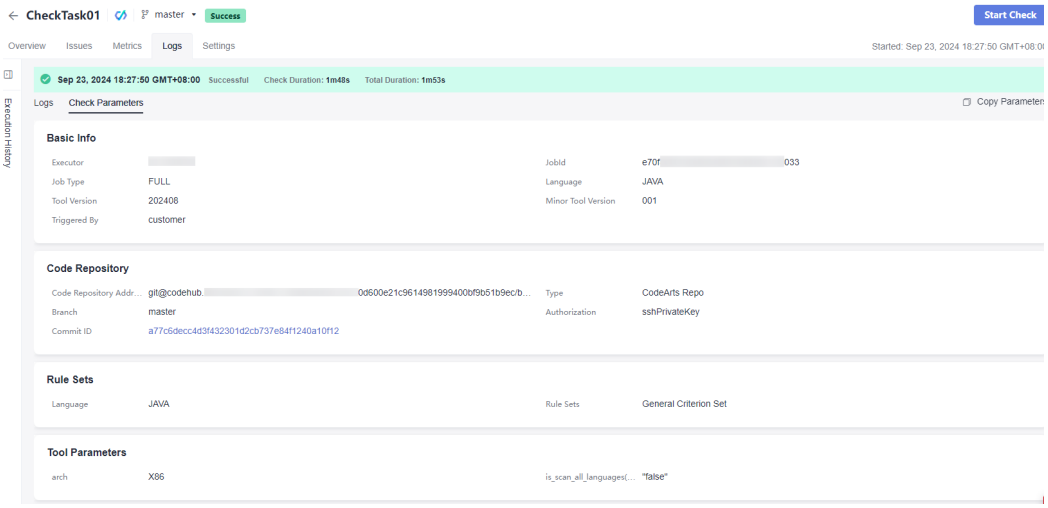


Figure 7-2 Check parameters



8 Querying Audit Logs

Cloud Trace Service (CTS) records operations on CodeArts Check for query, audit, and backtrack.

Operations Recorded by CTS

Table 8-1 Operations recorded by CTS

Operation	Resource Type	Event
Creating a task	task	create
Modifying a task	task	modify
Deleting a task	task	delete
Executing a task	job	create
Canceling a task	job	modify_job
Favoriting a task	followingTask	create_followingTask
Unfavoriting a task	followingTask	delete_followingTask
Modifying rule set information	ruleset	modify_ruleset
Modifying rules in a rule set	ruleset	modify_ruleset
Deleting a rule set	ruleset	delete_ruleset
Setting a rule set as default	ruleset	set_default_ruleset
Exporting a rule set	ruleset	export_ruleset
Exporting issues	defectExcel	export_defectExcel
Downloading issues	defectExcel	downloadAndExport_defectExcel

Operation	Resource Type	Event
Modifying an issue status	issueStatus	update_issueStatus
Changing the owner	issueOwner	update_issueOwner
Modifying an issue comment	issueComment	update_issueComment

Viewing Audit Logs

Query CodeArts Check traces on the CTS console. For details, see **Viewing Audit Logs**.

9 Reference

9.1 Using CodeArts Check Plug-in in IntelliJ IDEA

This plug-in is a powerful assistant for developers to protect the code quality. It provides industry standard check, one-click code style formatting, and automatic code fixing, adhering to the concepts of simplicity, high speed, and real-time monitoring.

Installing the Plug-in

Step 1 On the IntelliJ IDEA editor, choose **File** > **Settings** on the top menu bar.

Step 2 Click **Plugins** in the navigation pane on the left and click **Marketplace** in the right pane.

Step 3 Enter **Huawei Cloud CodeArts Check** or **CodeArts Check** in the search box.

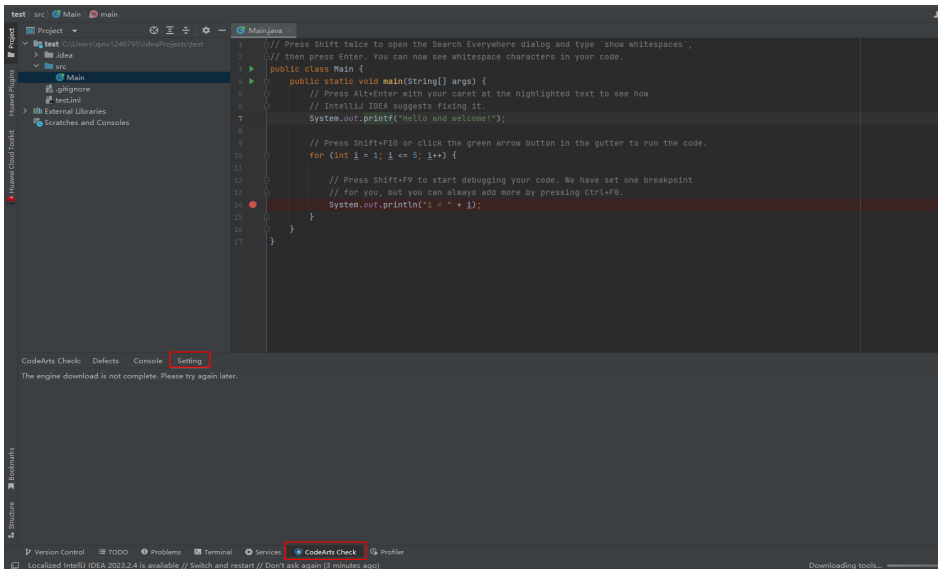
Step 4 Click **Install**. Then IntelliJ IDEA restarts. After the restart, IntelliJ IDEA automatically downloads the engine package and the JDK package.

Step 5 Once the download is finished, a message appears in the lower right corner confirming the successful download.

----End

Configuring the Plug-in

Step 1 Click **CodeArts Check** at the bottom of the IntelliJ IDEA editor and click **Setting** on the toolbar.



Step 2 In the displayed window, click  on the left to view the rule details.

Step 3 Decide whether to enable automatic code check, enter the masked directory, and specify directory to be checked.

Step 4 See the activated or applied rule details (including status, level, language, engine tag, and rule type) or filter rules by all/recommended rule set.

Step 5 After the configuration, click **Apply**.

----End

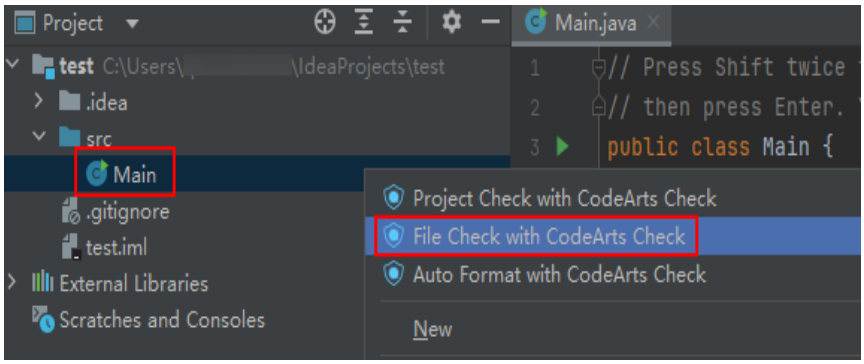
Checking and Fixing Files by the Plug-in

Take a local Java project as an example.

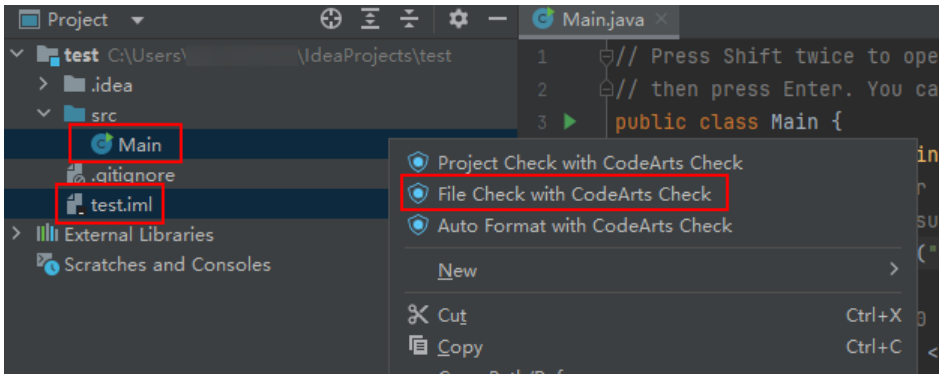
The CodeArts Check plug-in allows you to check one or multiple Java files, as well as the entire project. You can also correct code style issues with just one click. For details, see **Plug-in Functions**, **One-Click Formatting and Automatic Fix**, and **Rule Configuration**.

Plug-in Functions

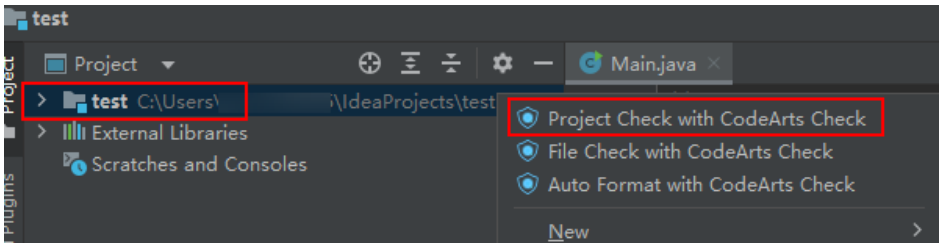
- **Checking one or multiple files, as well as the entire project, and viewing the results.**
  - Checking one file: In the editor area, right-click and click **File Check with CodeArts Check**. Alternatively, in the **Project** window, right-click the file to be checked and click **File Check with CodeArts Check**.



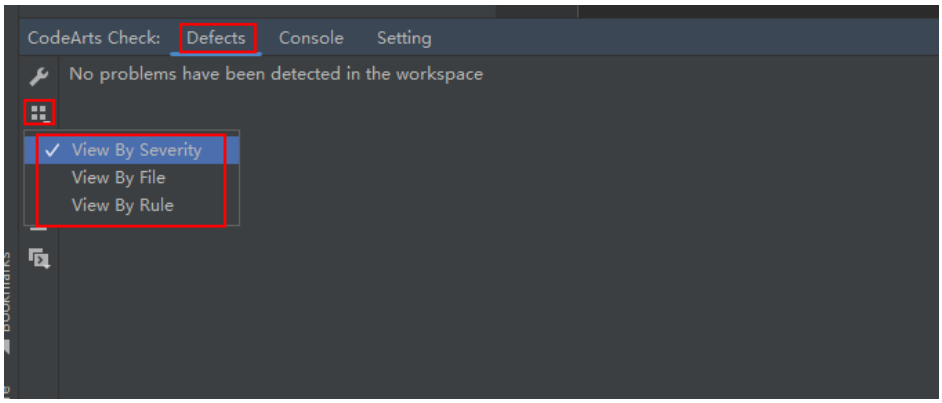
- Checking multiple files: In the **Project** window, hold down **Ctrl** and select multiple files. Then, right-click and click **File Check with CodeArts Check**.



- Checking the entire project: In the editor area, right-click and click **Project Check with CodeArts Check**. Alternatively, in the **Project** window, right-click the file and click **Project Check with CodeArts Check**.



- **Viewing defect details, severity, type, compliant example, non-compliant example, and fixing suggestion. Filtering defects by severity, rule, and file.**
  - Viewing defects by file: Open the **CodeArts Check** window, click **Defects** >  > **View By File**.



- Viewing defects by severity: Open the **CodeArts Check** window, click **Defects** > , and select a defect type as required.
- **Triggering automatic check and defect update.**

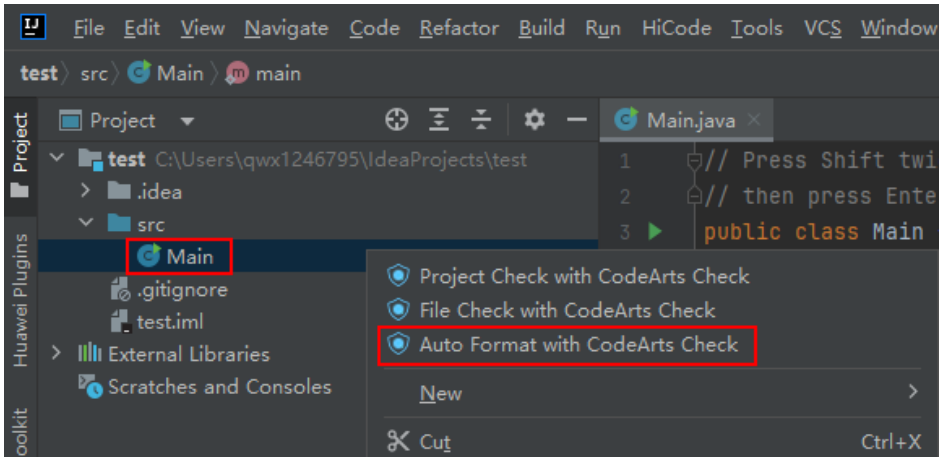
  The plug-in automatically checks code in real time and updates defects after you fix the code defects based on fixing suggestions.

One-Click Formatting and Automatic Fix

- One-click formatting on a single file, multiple files, a folder, or an entire project.

  With one-click formatting, you can easily resolve common code specification issues in your project, such as extra spaces or blank lines.

  In the editor area, right-click and click **Auto Format with CodeArts Check**. Alternatively, in the **Project** window, right-click the file and click **Auto Format with CodeArts Check**.



- Automatic fix for a single defect checked by the plug-in.

  The plug-in provides a preview window with suggestions on how to fix and the automatic fix function.

Rule Configuration

Open the CodeArts Check window, click **Setting**.

Table 9-1 Configurations

No.	Configuration	Description
1	Automatic code check after editing and before commit	• Automatic code check after saved editing: Select **whether to automatically trigger CodeArts Check after code is edited and saved**. Automatic check does not affect the current window. Once you modify and save the code, the latest check results will be automatically updated in the **Defects** tab. • Automatic code check before commit: Select **whether to automatically trigger CodeArts Check before code is committed**.
2	Check mode	The **local** mode is used by default.
3	Excluded directory	Specify multiple directories by separating them with commas (,). To check the dir1 directory, enter **dir1**. To check dir2 in the dir1 directory, enter **dir1/dir2**. If no directory is specified, all content of the entire project is checked by default. This parameter is valid only during project check.
4	Search by category	Search for or filter rules.
5	Setting rule	Select or deselect rules.
6	Uploading/ Exporting rule	Export the rule details to an XLSX file on the local PC. Upload the rule file in XLSX format prepared on the local PC.

