

代码检查

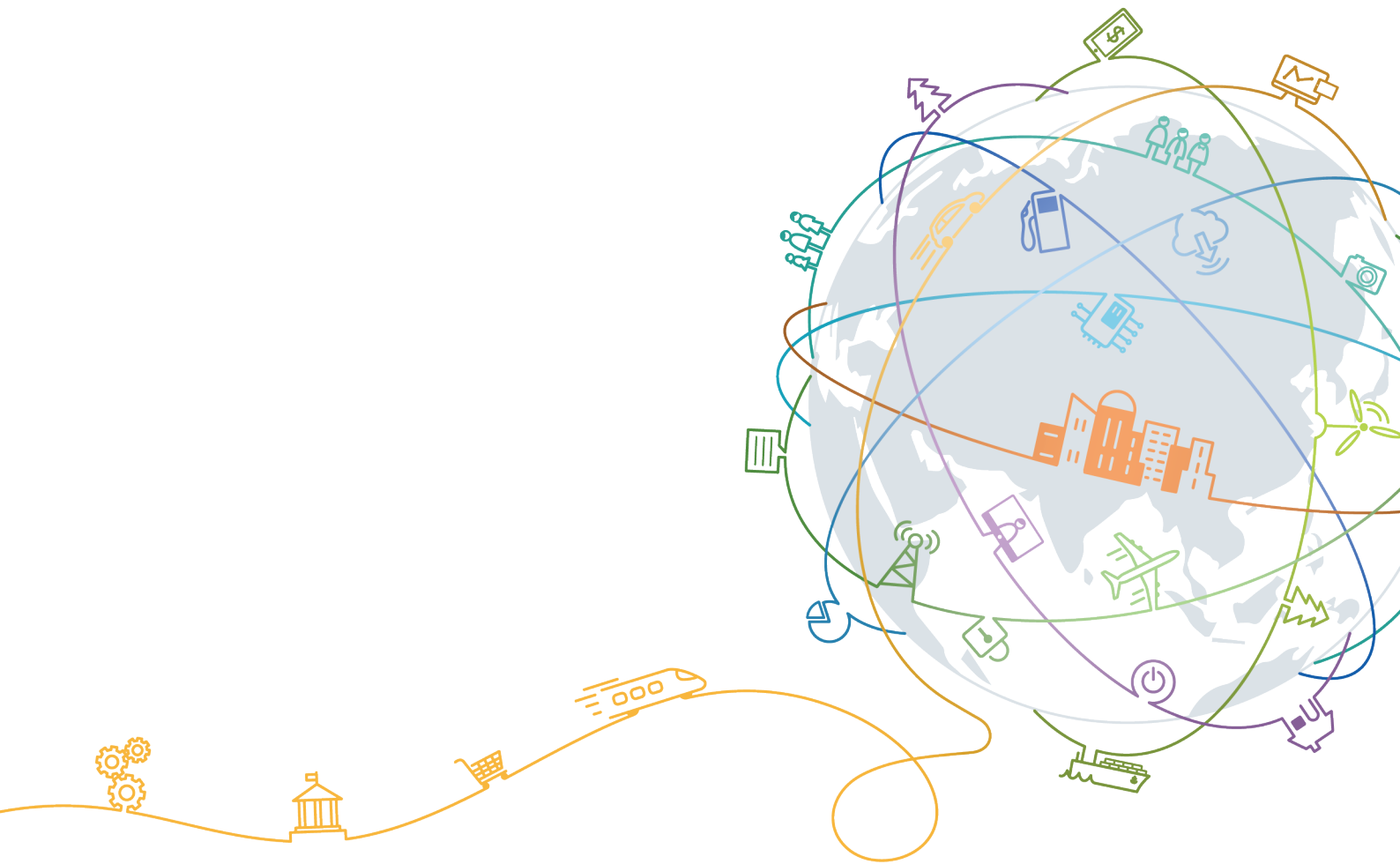
# 最佳实践

文档版本

01

发布日期

2024-05-08



版权所有 © 华为技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

---

## 目录

---

1 使用预置规则检查 GitCode 代码仓中的代码质量.....	1
2 使用预置规则检查通用 Git 代码仓中的代码质量.....	5
3 使用自定义规则检查 CodeArts Repo 代码仓中的代码质量.....	8
4 HE2E DevOps 实践：代码检查.....	13

# 1 使用预置规则检查 GitCode 代码仓中的代码质量

## 应用场景

如果您的代码存放在GitCode代码仓中，可参考本实践指导检查代码质量。本实践为您介绍如何使用系统预置规则检查GitCode代码仓中的Java语言代码质量。

## 前提准备


- 已[开通并授权使用代码检查服务](#)。
- GitCode代码仓中已有Java语言的代码。

## 操作流程

流程	说明
<a href="#">访问代码检查服务首页</a>	为您介绍如何访问代码检查服务首页。
<a href="#">创建代码检查任务所属项目</a>	创建代码检查任务所属的项目。
<a href="#">新建GitCode服务扩展点</a>	本实践检查第三方代码仓的代码质量，CodeArts Check服务需通过CodeArts提供的扩展插件连接到第三方代码仓。
<a href="#">新建GitCode代码检查任务</a>	创建代码检查任务。
<a href="#">执行GitCode代码检查任务</a>	执行代码检查任务。
<a href="#">查看GitCode代码检查详情</a>	查看代码检查结果。

## 访问代码检查服务首页

**步骤1** [登录华为云控制台页面](#)。

**步骤2** 单击页面左上角 ，在服务列表中选择“开发与运维 > 代码检查 CodeArts Check”。

**步骤3** 单击“立即使用”，进入代码检查服务首页。

----结束

## 创建代码检查任务所属项目

**步骤1** 单击导航栏“首页”。

**步骤2** 单击“新建项目”。

**步骤3** 单击“Scrum”项目模板。

**步骤4** 项目名称填写“Scrum01”，其它保持默认即可。

**步骤5** 单击“确定”后，进入到“Scrum01”项目下。

----结束

## 新建 GitCode 服务扩展点

服务扩展点（Endpoint）是软件开发生产线的一种扩展插件，为软件开发生产线提供链接第三方代码仓的能力。

代码检查服务默认检查CodeArts Repo服务的代码质量，同时也可使用服务扩展点连接第三方代码仓库检查代码质量。

**步骤1** 在导航栏选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤2** 单击“新建服务扩展点”，在下拉列表中选择“GitCode”。

**步骤3** 在弹框中配置以下信息。

参数名称	功能描述	是否必填
连接名称	服务扩展点的名称，可自定义。例如：GitCode。	是

参数名称	功能描述	是否必填
Token	<p>填写GitCode上获取的Token，获取方法如下：</p> <ol style="list-style-type: none"> <li>1. <a href="#">登录GitCode</a>。</li> <li>2. 单击页面右上角账号名称，选择“个人设置”。</li> <li>3. 单击“+访问令牌”，填写令牌名称以及到期时间。</li> <li>4. 单击“新建访问令牌”，生成“你的个人访问令牌”，即Token。</li> <li>5. 单击, 即可复制生成的Token。</li> </ol> <p><b>说明</b></p> <ul style="list-style-type: none"> <li>- Token生成后，请及时保存，下次刷新页面将无法读取，需要重新生成新Token。</li> <li>- 注意填写有效的Token描述信息，避免误删除导致构建失败。</li> <li>- 无需使用时及时删除Token，避免信息泄露。</li> </ul>	是

**步骤4** 单击“确定”

----结束

## 新建 GitCode 代码检查任务

**步骤1** 选择导航栏“代码 > 代码检查”。

**步骤2** 单击“新建任务”，进入新建任务页面，配置如下参数。

- 归属项目：无需设置，默认填充。
- 代码源：选择“GitCode”。
- 任务名称：自定义任务名称。例如：GitCodeCheckTask。
- Endpoint实例：选择[新建GitCode服务扩展点](#)中创建的扩展点“GitCode”。
- 仓库：选择需要进行代码检查的代码仓库。保持默认即可。
- 分支：选择需要检查的代码仓库的分支。保持默认即可。
- 检查语言：选择“Java”。

**步骤3** 单击“新建任务”，完成检查任务的创建。

----结束

## 执行 GitCode 代码检查任务

**步骤1** 在代码检查页面任务列表中，单击“GitCodeCheckTask”代码检查任务所在行 。

**步骤2** 根据页面提示等待任务执行完成。

----结束

## 查看 GitCode 代码检查详情

**步骤1** 在代码检查页面任务列表中，搜索[新建GitCode代码检查任务](#)创建的任务名称。

**步骤2** 单击任务名称，查看代码检查详情，包括概览、代码问题、代码度量、检查日志等。

----结束

# 2 使用预置规则检查通用 Git 代码仓中的代码质量

## 应用场景

如果您的代码存放在通用Git代码仓中，可参考本实践指导检查代码质量。本实践为您介绍如何使用系统预置规则检查通用Git代码仓中的Java语言代码质量。

## 前提准备

- 已[开通并授权使用代码检查服务](#)。
- 通用Git代码仓中已有Java语言的代码。


## 操作流程

流程	说明
<a href="#">访问代码检查服务首页</a>	为您介绍如何访问代码检查服务首页。
<a href="#">创建代码检查任务所属项目</a>	创建代码检查任务所属的项目。
<a href="#">新建通用Git服务扩展点</a>	本实践检查第三方代码仓的代码质量，代码检查服务需通过CodeArts提供的扩展插件连接到第三方代码仓。
<a href="#">创建通用Git代码检查任务</a>	创建代码检查任务。
<a href="#">执行通用Git代码检查任务</a>	执行代码检查任务。
<a href="#">查看代码检查结果</a>	查看代码检查结果。



## 访问代码检查服务首页

**步骤1** 登录[华为云控制台页面](#)。

**步骤2** 单击页面左上角 ，在服务列表中选择“开发与运维 > 代码检查 CodeArts Check”。

**步骤3** 单击“立即使用”，进入代码检查服务首页。

----结束

## 创建代码检查任务所属项目

**步骤1** 单击导航栏“首页”。

**步骤2** 单击“新建项目”。

**步骤3** 单击“Scrum”项目模板。

**步骤4** 项目名称填写“Scrum01”，其它保持默认即可。

**步骤5** 单击“确定”后，进入到“Scrum01”项目下。

----结束

## 新建通用 Git 服务扩展点

服务扩展点（Endpoint）是软件开发生产线的一种扩展插件，为软件开发生产线提供链接第三方代码仓的能力。

代码检查服务默认检查CodeArts Repo服务的代码质量，同时也可使用服务扩展点连接第三方代码仓库检查代码质量。

**步骤1** 在导航栏选择“设置 > 通用设置 > 服务扩展点管理”。

**步骤2** 单击“新建服务扩展点”，在下拉列表中选择“通用Git”。

**步骤3** 在弹框中配置以下信息，单击“确定”。

表 2-1 新建通用 Git 服务扩展点

参数	说明	是否必填
连接名称	自定义，支持中文、英文、数字、“-”、“_”、“.”、空格，不超过256个字符。例如：Endpoint01。	是
Git仓库Url	输入待连接的Git仓库的https地址。	是
用户名	输入待连接的Git仓库的用户名，不超过300个字符。	否
密码或Access Token	输入待连接的Git仓库的密码，不超过300个字符。	否

----结束

## 创建通用 Git 代码检查任务

**步骤1** 在导航栏选择“代码 > 代码检查”。


**步骤2** 单击“新建任务”，进入“新建任务”页面，按照下面配置参数。

参数项	描述
归属项目	任务所属项目。默认填写为 <a href="#">创建代码检查任务所属项目</a> 中的项目名称“Scrum01”，无需手动填写。
代码源	选择“通用Git”。
任务名称	代码检查任务名称，可自定义。例如：CheckTask01。
Endpoint实例	选择 <a href="#">新建通用Git服务扩展点</a> 中创建的服务扩展点“Endpoint01”。
仓库	选择需要检查的代码仓库，保持默认即可。
分支	选择需要检查的仓库分支，保持默认即可。
检查语言	选择需要检查的代码语言，选择“Java”。

**步骤3** 单击“确定”，完成检查任务的创建。

----结束

## 执行通用 Git 代码检查任务

**步骤1** 在代码检查页面任务列表中，单击代码检查任务所在行  执行任务。

**步骤2** 根据页面提示等待任务执行完成。

----结束

## 查看代码检查结果

**步骤1** 在代码检查页面任务列表中，搜索[创建通用Git代码检查任务](#)创建的任务名称“CheckTask01”。

**步骤2** 单击任务名称，查看代码检查详情，包括概览、代码问题、代码度量、检查日志等。  
至此，我们完成了一次基本的通用Git源码源常用检查流程。

----结束

# 3 使用自定义规则检查 CodeArts Repo 代码仓中的代码质量

## 应用场景

随着现在项目的代码量越来越大，以及现有开发框架的增加。静态分析所需要覆盖的场景也日益增多。

- 传统静态分析引擎靠工具提供通用规则来对常见安全，风格和质量场景下的编码问题进行扫描分析，难以实时覆盖不同用户特定场景下的代码问题。
- 用户往往不熟悉工具提供的通用规则所能覆盖的全部场景。在为新开发的业务场景寻找适用的规则时需要耗费大量的时间，增加开发成本。
- 不同用户之间的代码差异大，研发无法实时掌握不同用户的具体需求，在开发规则过程中难以针对不同用户的业务代码场景来制定全面有效的工具规则。

本示例为您介绍如何使用自定义规则检查代码中存在调试代码的问题。

## 前提准备

- 已[开通并授权使用代码检查服务](#)。
- 通用Git代码仓中已有Java语言的代码。


## 操作流程

流程	说明
<a href="#">访问代码检查服务首页</a>	为您介绍如何访问代码检查服务首页。
<a href="#">创建代码检查任务所属项目</a>	创建代码检查任务所属的项目。
<a href="#">创建CodeArts Repo代码仓</a>	创建代码检查任务使用的代码仓。
<a href="#">创建规则文件</a>	创建自定义规则时需要上传使用的规则文件。
<a href="#">创建自定义规则</a>	在代码检查服务页面创建自定义规则。

流程	说明
<a href="#">创建自定义规则集</a>	自定义规则无法直接使用，需要通过创建自定义规则集使用。
<a href="#">创建代码检查任务</a>	创建使用自定义规则检查的代码检查任务。
<a href="#">配置代码检查任务使用自定义规则集并执行检查</a>	配置代码检查任务使用带有自定义规则的自定义规则集。
<a href="#">查看检查结果</a>	查看检查结果，确认使用的规则是否生效。

## 访问代码检查服务首页

步骤1 [登录华为云控制台页面](#)。

步骤2 单击页面左上角 ，在服务列表中选择“开发与运维 > 代码检查 CodeArts Check”。

步骤3 单击“立即使用”，进入代码检查服务首页。

----结束

## 创建代码检查任务所属项目

步骤1 单击导航栏“首页”。

步骤2 单击“新建项目”。

步骤3 单击“Scrum”项目模板。

步骤4 项目名称填写“Scrum01”，其它保持默认即可。

步骤5 单击“确定”后，进入到“Scrum01”项目下。

----结束

## 创建 CodeArts Repo 代码仓

步骤1 在页面导航栏中选择“代码 > 代码托管”。

步骤2 进入代码托管页面，单击“新建仓库”，选择“模板仓库”。

步骤3 单击“下一步”，在“按模板新建 > 选择仓库模板”，搜索并选择“Java Ant Demo”模板。

步骤4 单击“下一步”，仓库名称填写“Repo01”，“自动创建代码检查任务”参数需要去除勾选。其他参数保持默认即可。

步骤5 后单击“确定”完成仓库创建。

步骤6 在“com/huawei”目录下的“HelloWorld.java”文件中修改后的代码信息如下。

```
package com.huawei;  
/**
```

```
* Generate a unique number
*
*/
public class HelloWorld
{
//用于打印日志
public void debugLog(List<String> msg) {
    for (String msg0 : msg) {
        System.out.println("DEBUG:" + msg0);
    }
}

public static void main( String[] args )
{
    System.out.println("Hello World!");
}
}
```

----结束

## 创建规则文件

**步骤1** 在[Visual Studio Code官网](#)下载并安装1.67.0以上版本的IDE编辑器。

**步骤2** 在VSCode IDE编辑器界面上，单击左侧按钮，在弹出的窗口中搜索“Huawei Cloud CodeNavi”。

**步骤3** 单击“Install”，安装代码检查插件。

**步骤4** 安装后，在VSCode IDE编辑器工作空间新建后缀为“.kirin”的文件，文件名为自定义规则的规则名，例如：CheckDebugCode.kirin。文件内容如下所示。

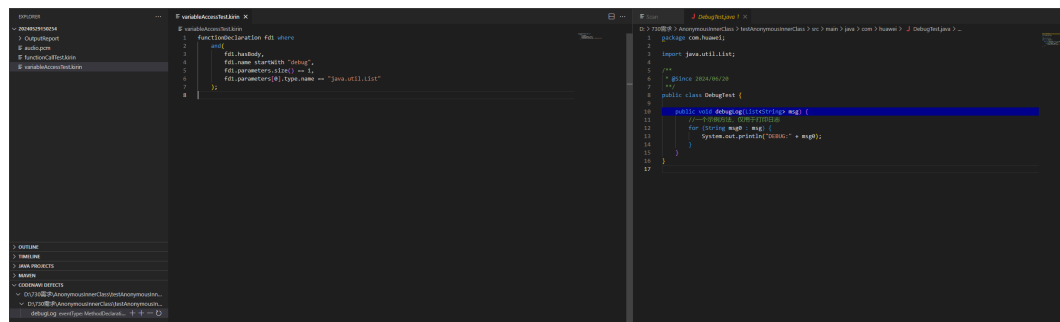
```
functionDeclaration fd1 where
    and(
        fd1.hasBody,
        fd1.name startWith "debug",
        fd1.parameters.size() == 1,
        fd1.parameters[0].type.name == "java.util.List"
    );
```

**步骤5** 在规则文件界面上鼠标右键，选择“CodeNavi > Format格式化”校验语法正确性。

**步骤6** 在规则文件界面上鼠标右键，选择“CodeNavi > Scan扫描”。

**步骤7** 在弹出的窗口中选择需要检查的文件或者目录，单击“Scan”。

**步骤8** 扫描完成后，单击界面左下角的告警显示具体的代码片段。同时，会在同目录的OutputReport文件中生成“.json”格式的规则文件。



----结束

## 创建自定义规则

**步骤1** 在页面导航栏中选择“代码 > 代码检查”。

**步骤2** 单击“规则”页签。

**步骤3** 单击“新建规则”，参考表3-1配置参数。

表 3-1 自定义规则参数配置说明

参数	说明
规则名称	创建的规则名称，可自定义。例如：检查调试代码。
工具规则名称	默认填充为规则源码文件名，不可手动修改。
检查工具	该检查规则使用的检查工具，当前仅支持SecBrella。
语言	该规则检查的编译语言，当前仅支持Java。
规则源码	该规则的源码文件。上传 <a href="#">创建规则文件</a> 中生成的文件。
严重级别	该规则检查出的代码问题的严重级别，分为致命、严重、一般和提示。这里设置为“提示”。
标签	为该规则设置标签，便于使用时区分该规则使用场景。非必填，无需填写。 <b>说明</b> 添加多个标签时，需使用英文逗号分隔。
描述	对该规则的使用描述。支持Markdown语言，字符数限制为0~10000。例如：检查是否存在调试代码。
正确示例	该规则对应的正确代码示例。支持Markdown语言，字符数限制为0~10000。非必填，无需填写。
错误示例	不满足该规则的错误代码示例。支持Markdown语言，字符数限制为0~10000。非必填，无需填写。
修复建议	对于该规则检查出的问题修改建议。支持Markdown语言，字符数限制为0~10000。非必填，无需填写。

**步骤4** 单击“新建规则”，完成创建。

----结束

## 创建自定义规则集

**步骤1** 在代码检查任务列表页，单击“规则集”页签，进入规则集列表页面。

**步骤2** 单击“新建规则集”，在弹出的窗口设置“规则集名称”为“RuleList”，“检查语言”为“Java”。

**步骤3** 单击“确定”，进入到“规则集配置”页面。

**步骤4** 勾选**创建自定义规则**中创建的规则，单击右上角“保存”。

----结束

## 创建代码检查任务

**步骤1** 在代码检查任务列表页，单击“新建任务”，按照如下表格配置参数。

参数项	说明
归属项目	<b>创建代码检查任务所属项目</b> 中创建的项目名称“Scrum01”。默认填写，无需配置。
代码源	选择“Repo”。
任务名称	代码检查任务名称，可自定义。例如：CheckTask01。
仓库	选择 <b>创建CodeArts Repo代码仓</b> 中创建的代码仓“Repo01”。
分支	保持默认“mater”即可。
检查语言	选择“Java”。

**步骤2** 单击“新建任务”，完成代码检查任务的创建。

----结束

## 配置代码检查任务使用自定义规则集并执行检查

**步骤1** 在代码检查任务列表页单击代码检查任务名称，进入代码检查任务详情页。

**步骤2** 单击“设置”。

**步骤3** 单击“规则集”，在右侧区域单击  选择**创建自定义规则集**中创建的规则集“RuleList”。

**步骤4** 单击右上角“开始检查”。

----结束

## 查看检查结果

**步骤1** 在代码检查页面任务列表中，搜索**创建代码检查任务**创建的任务名称“CheckTask01”。

**步骤2** 单击任务名称，查看代码检查详情，包括概览、代码问题、代码度量、检查日志等。

----结束

# 4 HE2E DevOps 实践：代码检查

本文以“DevOps全流程示例项目”为例，介绍如何在项目中完成代码检查配置。  
开展实践前，需要完成[创建项目](#)和[创建仓库](#)。

## 预置任务简介

样例项目中预置了以下4个代码检查任务。

表 4-1 预置任务

预置任务	任务说明
phoenix-codecheck-worker	检查Worker功能对应代码的任务。
phoenix-codecheck-result	检查Result功能对应代码的任务。
phoenix-codecheck-vote	检查Vote功能对应代码的任务。
phoenix-sample-javas	检查整个代码仓库对应的JavaScript代码的任务。

本章节以任务“phoenix-codecheck-worker”为例进行讲解。


## 配置并执行任务

开发人员可以对样例项目中预置的任务做一些简单的配置，增加Python语言检查规则集，使检查更全面。

### 步骤1 编辑任务。


1. 进入“凤凰商城”项目，单击导航“代码 > 代码检查”，页面中显示样例项目内置的4个任务。
2. 在列表中找到任务“phoenix-codecheck-worker”。
3. 单击任务名称进入详情页，选择“设置”页签。



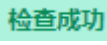
4. 单击导航“规则集”，规则集中默认包含的语言是“JAVA”。
5. 增加Python语言检查规则集。
  - a. 单击“已包含语言”之后的图标，重新获取代码仓库语言，刷新后的列表新增了多种语言。

#### 说明

如果页面中已显示“PYTHON”，则忽略此步骤。

- b. 将PYTHON语言对应的开关打开.
- c. 在弹框中单击“确定”。

#### 步骤2 执行任务。

1. 单击“开始检查”，启动任务。
2. 当页面显示  **检查成功**，表示任务执行成功。  
如果任务执行失败，请根据页面弹出报错提示排查修改。

----结束

## 查看检查结果

代码检查服务提供检查结果统计，并对检查出的问题提供修改建议，可以根据修改建议优化项目代码。

**步骤1** 在代码检查任务中，选择“概览”页签，即可查看任务执行结果统计。

**步骤2** 单击“代码问题”页签，即可看到问题列表。

单击问题框中的“问题帮助”，可以查看系统对此问题的修改建议。可以根据需要在代码仓库中找到对应文件及代码位置，参考修改建议优化代码。

----结束