

代码检查

用户指南

文档版本 01
发布日期 2024-09-11



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。

安全声明

漏洞处理流程

华为公司对产品漏洞管理的规定以“漏洞处理流程”为准，该流程的详细内容请参见如下网址：

<https://www.huawei.com/cn/psirt/vul-response-process>

如企业客户须获取漏洞信息，请参见如下网址：

<https://securitybulletin.huawei.com/enterprise/cn/security-advisory>

目录

1 代码检查服务(CodeArts Check)使用流程	1
2 开通并授权使用 CodeArts Check	3
2.1 开通 CodeArts Check	3
3 配置 CodeArts Check 项目级角色权限	6
4 创建代码检查任务	9
5 配置代码检查任务	15
5.1 配置代码检查任务规则集	15
5.1.1 配置代码检查任务预置规则集	15
5.1.2 配置代码检查任务自定义规则集	23
5.2 配置代码检查任务质量门禁	25
5.3 配置代码检查任务定时执行	26
5.4 配置代码检查任务消息通知	27
5.5 配置代码检查任务检查模式	28
5.6 查看徽标状态	30
5.7 配置代码检查任务集成服务	31
5.8 配置代码检查任务自定义环境	32
5.9 配置代码检查任务高级选项	33
6 执行代码检查任务	35
7 查看代码检查结果	37
8 变更代码检查服务规格	42
9 查询审计日志	44
10 参考	46
10.1 CodeArts Check 插件使用指南 (IntelliJ IDEA 版本)	46

1 代码检查服务(CodeArts Check)使用流程

代码检查(CodeArts Check)是基于云端实现的代码检查服务。建立在多年自动化源代码静态检查技术积累与企业级应用经验的沉淀之上，为用户提供代码风格、通用质量与网络安全风险等丰富的检查能力，提供全面的质量报告、便捷的问题闭环处理流程，帮助企业有效管控代码质量，助力企业成功。

使用流程

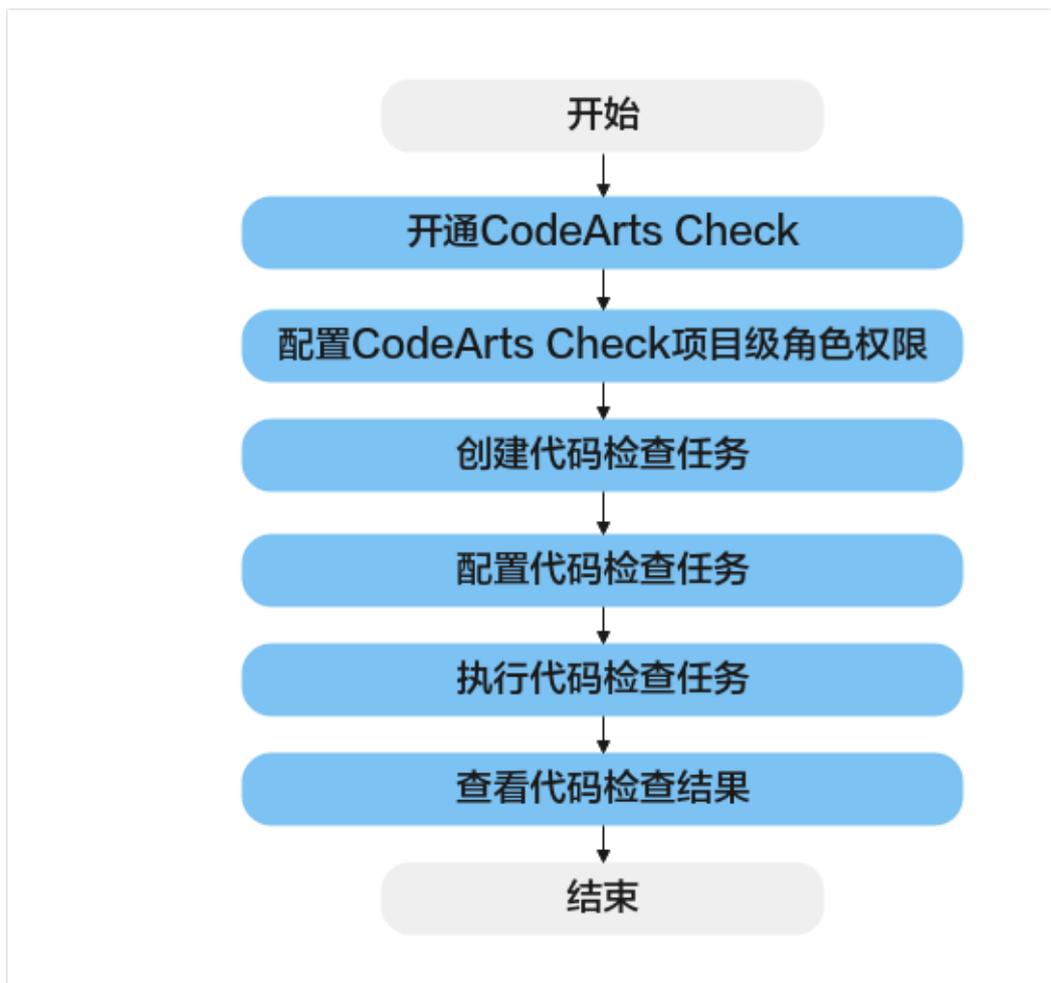


表 1-1 使用流程说明

流程	说明
开通 CodeArts Check	指导您如何开通代码检查服务。
配置 CodeArts Check项目级角色权限	为您介绍如何配置CodeArts Check服务项目级权限以及访问代码检查服务首页的方式。
创建代码检查任务	您可创建不同代码源的代码检查任务。
配置代码检查任务	您可对已有的检查任务进行修改或配置。
执行代码检查任务	您可对目标代码检查任务执行检查，及时发现源代码中的代码问题。
查看代码检查结果	代码检查任务执行完成后，您可查看检查结果并修改代码问题。

2 开通并授权使用 CodeArts Check

2.1 开通 CodeArts Check

前提条件

已注册华为云并实名认证，如果还没有华为账号，请参考以下步骤创建。

1. 打开[华为云网站](#)。
2. 单击“注册”，根据提示信息完成注册。
注册成功后，系统会自动跳转至您的个人信息界面。
3. 参考[实名认证](#)完成个人或企业账号实名认证。

购买 CodeArts Check 须知

- 在[CodeArts支持的区域](#)内，各区域独立开通/购买、独立计费。
- 您可以[开通CodeArts Check](#)，或者[开通/购买软件开发生产线服务组合套餐](#)，体验一站式、全流程、安全可信的软件开发生产线。

说明

若已经购买了CodeArts套餐，则不能再单独购买代码检查服务套餐。

开通 CodeArts Check

代码检查服务采用包年/包月计费模式，提供基础版、专业版套餐，以满足不同规模用户的使用需求。套餐详情介绍请参见[包年/包月套餐说明](#)。

步骤1 进入[购买CodeArts Check套餐页面](#)。

步骤2 参考[表2-1](#)根据实际需要进行购买。

表 2-1 套餐包配置

配置项	配置详情
计费模式	包年/包月。

配置项	配置详情
区域	选择需要使用的区域。不同区域购买的资源不能跨区使用，需慎重选择。
产品	CodeArts Check套餐
规格	<p>可选择“基础版”或“专业版”，差异如下。</p> <ul style="list-style-type: none"> ● 基础版 <ul style="list-style-type: none"> - 支持Java/C++/JavaScript/Go/Python/C#/TypeScript/CSS/HTML/PHP/LUA/RUST/Shell/KOTLIN/SCALA/ArkTS10+种语言。 - 支持质量类、安全类规则。 - 检查并发：5个任务并发数。 ● 专业版 <ul style="list-style-type: none"> - 基础版所有特性 - 提供增强的代码安全检查规则，深度识别代码安全风险和漏洞 - 检查并发：10个任务并发数。
购买时长	可选择1个月~3年。
自动续费	勾选后将开启自动续费。自动续费规则请参考 自动续费规则说明 。

📖 说明

建议您根据您业务所在物理区域就近选择，以减少网络延时。购买的套餐只在对应的区域生效，不能跨区域使用。

步骤3 确认套餐信息，勾选我已经阅读并同意《CodeArts服务使用声明》，单击“下一步”。

图 2-1 确认套餐信息

The screenshot shows a configuration interface for CodeArts Check. It includes the following sections:

- 计费模式:** 包年/包月 (selected)
- 区域:** 华北-北京一, 华北-北京四 (selected), 华北-乌兰察布二零一, 华北-乌兰察布二零二, 华东-上海一, 华东-上海二, 华南-广州, 西南-贵阳一, 亚太-新加坡, 拉美-墨西哥城二. A note below states: "不同区域购买的资源不能跨区域使用，请谨慎选择"
- 产品:** CodeArts Check套餐
- 规格:** 基础版 (selected), 专业版
- 功能特性:** 支持Java、C/C++、Python、JavaScript、C#、Go等10种语言; 支持质量类、安全类规则
- 资源规格:** 检查并发：5个任务并发数
- 购买时长:** 1 (selected), 2, 3, 4, 5, 6, 7, 8, 9个月, 1年, 2年, 3年
- 自动续费:** 自动续费 了解 扣费规则 和 续费时长
- 协议:** 我已经阅读并同意 《CodeArts服务使用声明》

步骤4 若确认无误，继续根据页面提示完成支付。

----结束

3 配置 CodeArts Check 项目级角色权限

新增的成员需赋予指定的角色，不同角色具备的默认权限不同。各角色具备的默认权限如表3-1。

表 3-1 代码检查服务默认角色权限

资源权限/角色		项目管理员	项目经理	开发人员	测试经理	测试人员	参与者	浏览者	运维经理	产品经理	系统工程师	Committer
代码检查任务	创建	√	√	√	×	×	×	×	×	×	√	√
	执行	√	√	√	×	×	×	×	×	√	√	√
	查看	√	√	√	×	×	×	×	×	√	√	√
	修改	√	√	√	×	×	×	×	×	√	√	√
	删除	√	√	×	×	×	×	×	×	√	√	√
规则集	设置默认规则集	√	√	×	×	×	×	×	×	√	√	√
缺陷报告	查看	√	√	√	×	×	×	×	×	√	√	√
	修改	√	√	√	×	×	×	×	×	√	√	√

资源权限/ 角色	项目 管理员	项目 经理	开发 人员	测试 经理	测试 人员	参 与 者	浏 览 者	运 维 经 理	产 品 经 理	系 统 工 程 师	Co m m i t t e r
导出	√	√	√	×	×	×	×	×	√	√	√

前提条件

- 已[开通CodeArts Check](#)。
- 已参考软件开发生产线(CodeArts)的“用户指南 > 软件开发生产线 (CodeArts) 使用前准备 > 添加CodeArts项目成员”章节[添加成员](#)，并为新增的成员[赋予角色](#)。

访问 CodeArts Check 服务首页

步骤1 使用华为云账号[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 代码检查服务页面有两种访问方式：首页入口和项目入口。

- **首页入口**

单击“立即使用”，进入代码检查服务首页。该页面展示的是与当前用户相关的代码检查任务列表。



- **项目入口**

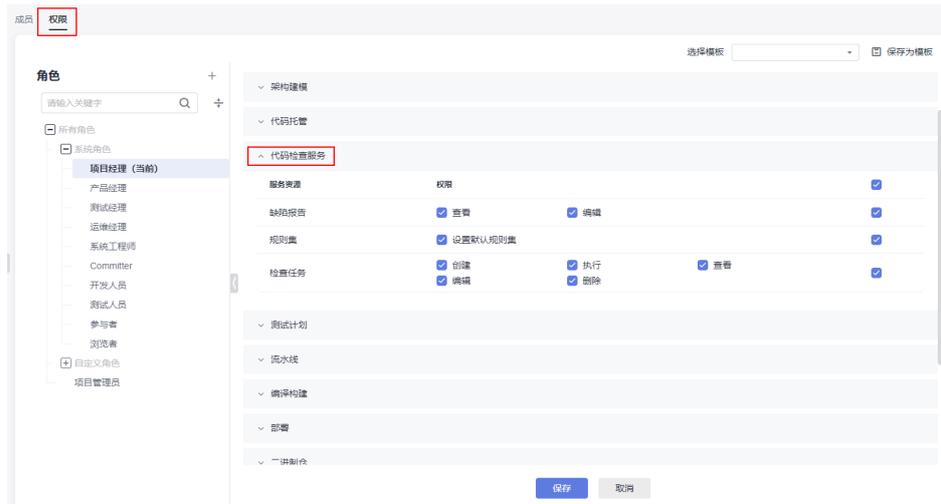
- 单击“立即使用”，进入代码检查服务首页。
- 单击导航栏“首页”。
- 单击需要查看的项目名称。
- 选择“代码 > 代码检查”，进入指定项目下代码检查任务列表页。

单击页面左上角, 可根据需要选择区域。

----结束

配置项目级角色权限

1. 通过项目入口方式[访问CodeArts Check服务首页](#)。
2. 选择导航栏“设置 > 通用设置 > 服务权限管理”，选择“权限”页签。
3. 在配置权限页面，为不同的角色配置编译构建服务的各个资源权限。



4 创建代码检查任务

代码检查服务可检查不同代码仓的代码质量，在创建代码检查任务前，需满足以下条件：

- 已新建CodeArts项目。
- 如果使用的是代码托管服务（CodeArts Repo）的代码仓库，则可以参考代码托管服务的“用户指南 > 创建 > 代码托管仓库”，完成新建代码仓库。
- 如果使用的是第三方代码仓库，需已创建可用的代码仓库。

创建 Repo 代码源检查任务

在代码托管中创建云端仓库时，若勾选了“自动创建代码检查任务”，则代码仓库创建完成后，在代码检查任务列表可查看对应仓库的检查任务。

以下示例为在代码托管服务中创建云端仓库不勾选“自动创建代码检查任务”时，Repo代码源代码检查任务的创建步骤。

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 在代码检查页面，单击“新建任务”，在“新建任务”页面，参考表4-1配置参数后单击“新建任务”。

表 4-1 Repo 代码源检查任务参数说明

参数名称	参数说明	是否必填
归属项目	任务所属项目。默认填写，无需设置。 <ul style="list-style-type: none">● 以项目入口方式访问访问编译构建服务时默认填写，无需手动填写。● 以服务入口访问时需根据实际情况选择新建项目中创建的项目。	是
代码源	选择Repo，检查代码托管服务中的代码质量。	是
任务名称	代码检查任务名称，可自定义。 <ul style="list-style-type: none">● 支持中英文，数字，点，下划线“_”和连接符“-”。● 字符长度范围为1~128。	是

参数名称	参数说明	是否必填
仓库	选择需要检查的代码仓库。	是
分支	选择需要检查的仓库分支。	是
检查语言	选择需要检查的代码语言。	否

📖 说明

- 创建代码检查任务后，如果您需要修改任务名称或默认分支，可在“代码检查详情”页面，选择“设置 > 基本信息”，修改任务名称和默认分支。
- 对于不再继续使用的代码检查任务，可进入“代码检查详情”页面，选择“设置 > 基本信息”，删除当前代码检查任务。也可以单击代码检查任务所在行 *******，在下拉框中，选择“删除”。

若删除的代码检查任务其代码仓为多分支代码仓，分支数量小于200时可删除成功，大于等于200会删除失败。

----结束

创建 GitHub 代码源检查任务

创建GitHub代码源检查任务前，需将Github仓库设置成公开。

📖 说明

- 使用第三方代码仓库可能出现网络不稳定或其他问题，具体使用体验取决于第三方代码仓库网络环境和服务状态。
- 建议使用代码托管的代码导入功能，将代码导入到代码托管，实现安全、稳定、高效下载与构建。

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 在代码检查页面，单击“新建任务”，在“新建任务”页面，参考[表4-2](#)配置参数后单击“新建任务”。

表 4-2 GitHub 代码源检查任务参数说明

参数项	说明	是否必填
归属项目	任务所属项目。默认填写，无需设置。 <ul style="list-style-type: none"> • 以项目入口方式访问访问编译构建服务时默认填写，无需手动填写。 • 以服务入口访问时需根据实际情况选择新建项目中创建的项目。 	是
代码源	选择GitHub，检查托管在GitHub代码仓中的代码质量。	是

参数项	说明	是否必填
任务名称	代码检查任务名称，可自定义。 <ul style="list-style-type: none"> 支持中英文，数字，点，下划线“_”和连接符“-”。 字符长度范围为1~128。 	是
Endpoint实例	选择连接到GitHub代码仓库的服务扩展点。如果没有，可参考 新建服务扩展点 中“新建GitHub服务扩展点”创建。	是
Repository	选择需要检查的代码仓库。	是
Branch	选择需要检查的仓库分支。	是
检查语言	选择需要检查的代码语言。	否

📖 说明

- 创建代码检查任务后，如果您需要修改任务名称或默认分支，可在“代码检查详情”页面，选择“设置 > 基本信息”，修改任务名称和默认分支。
- 对于不再继续使用的代码检查任务，可进入“代码检查详情”页面，选择“设置 > 基本信息”，删除当前代码检查任务。也可以单击代码检查任务所在行^{***}，在下拉框中，选择“删除”。

若删除的代码检查任务其代码仓为多分支代码仓，分支数量小于200时可删除成功，大于等于200会删除失败。

----结束

创建 GitCode 代码源检查任务

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 在代码检查页面，单击“新建任务”，在“新建任务”页面，参考[表4-3](#)配置参数后单击“新建任务”。

表 4-3 GitCode 代码源检查任务参数说明

参数项	说明	是否必填
归属项目	任务所属项目。默认填写，无需设置。 <ul style="list-style-type: none"> 以项目入口方式访问访问编译构建服务时默认填写，无需手动填写。 以服务入口访问时需根据实际情况选择新建项目中创建的项目。 	是
代码源	选择GitCode，检查托管在GitCode代码仓中的代码质量。	是

参数项	说明	是否必填
任务名称	代码检查任务名称，可自定义。 <ul style="list-style-type: none"> 支持中英文，数字，点，下划线“_”和连接符“-”。 字符长度范围为1~128。 	是
Endpoint实例	选择连接到GitCode代码仓库的服务扩展点。如果没有，可参考 新建服务扩展点 中“新建GitCode服务扩展点”创建。	是
仓库	选择需要检查的代码仓库。	是
分支	选择需要检查的仓库分支。	是
检查语言	选择需要检查的代码语言。	否

📖 说明

- 创建代码检查任务后，如果您需要修改任务名称或默认分支，可在“代码检查详情”页面，选择“设置 > 基本信息”，修改任务名称和默认分支。
- 对于不再继续使用的代码检查任务，可进入“代码检查详情”页面，选择“设置 > 基本信息”，删除当前代码检查任务。也可以单击代码检查任务所在行 *******，在下拉框中，选择“删除”。

若删除的代码检查任务其代码仓为多分支代码仓，分支数量小于200时可删除成功，大于等于200会删除失败。

----结束

创建通用 Git 代码源检查任务

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 在代码检查页面，单击“新建任务”，在“新建任务”页面，参考[表4-4](#)配置参数后单击“新建任务”。

表 4-4 通用 Git 代码源检查任务参数说明

参数项	描述	是否必填
归属项目	任务所属项目。默认填写，无需设置。 <ul style="list-style-type: none"> 以项目入口方式访问访问编译构建服务时默认填写，无需手动填写。 以服务入口访问时需根据实际情况选择新建项目中创建的项目。 	是

参数项	描述	是否必填
代码源	选择通用Git。对于托管在其他服务上的代码，可以使用通用Git连接实现代码拉取。	是
任务名称	代码检查任务名称，可自定义。 <ul style="list-style-type: none"> 支持中英文，数字，点，下划线“_”和连接符“-”。 字符长度范围为1~128。 	是
Endpoint实例	选择连接到通用Git代码仓库的服务扩展点。如果没有，可参考#codecheck_ug_0000/section5999559152718中“新建通用Git服务扩展点”创建。	是
仓库	选择需要检查的代码仓库。	是
分支	输入需要检查的仓库分支。	是
检查语言	选择需要检查的代码语言。	否

📖 说明

- 创建代码检查任务后，如果您需要修改任务名称或默认分支，可在“代码检查详情”页面，选择“设置 > 基本信息”，修改任务名称和默认分支。
- 对于不再继续使用的代码检查任务，可进入“代码检查详情”页面，选择“设置 > 基本信息”，删除当前代码检查任务。也可以单击代码检查任务所在行 *******，在下拉框中，选择“删除”。

若删除的代码检查任务其代码仓为多分支代码仓，分支数量小于200时可删除成功，大于等于200会删除失败。

----结束

创建码云代码源检查任务

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 在代码检查页面，单击“新建任务”，在“新建任务”页面，参考[表4-5](#)配置参数后单击“新建任务”。

表 4-5 码云代码源检查任务参数说明

参数项	说明	是否必填
归属项目	任务所属项目。默认填写，无需设置。 <ul style="list-style-type: none"> 以项目入口方式访问访问编译构建服务时默认填写，无需手动填写。 以服务入口访问时需根据实际情况选择新建项目中创建的项目。 	是
代码源	选择码云，检查托管在码云代码仓中的代码质量。	是
任务名称	代码检查任务名称，可自定义。 <ul style="list-style-type: none"> 支持中英文，数字，点，下划线“_”和连接符“-”。 字符长度范围为1~128。 	是
Endpoint实例	选择连接到码云代码仓库的服务扩展点。如果没有，可参考新建服务扩展点中“新建码云服务扩展点”创建。	是
Repository	选择需要检查的代码仓库。	是
Branch	选择需要检查的仓库分支。	是
检查语言	选择需要检查的代码语言。	否

📖 说明

- 创建代码检查任务后，如果您需要修改任务名称或默认分支，可在“代码检查详情”页面，选择“设置 > 基本信息”，修改任务名称和默认分支。
- 对于不再继续使用的代码检查任务，可进入“代码检查详情”页面，选择“设置 > 基本信息”，删除当前代码检查任务。也可以单击代码检查任务所在行 *******，在下拉框中，选择“删除”。

若删除的代码检查任务其代码仓为多分支代码仓，分支数量小于200时可删除成功，大于等于200会删除失败。

----结束

5 配置代码检查任务

5.1 配置代码检查任务规则集

5.1.1 配置代码检查任务预置规则集

规则集则是一组用于做代码检查的规则组成的集合。一般根据不同的检查目标和需求，规则集也有相应的分类，比如对安全类、安卓应用、编程风格，使用这些类别进行检查的时候，规则集内的规则也会有所不同。

用户可根据实际需要更换代码检查任务使用的规则集，新启用的规则集将在下一次执行检查时生效。

代码检查服务支持多种语言的规则集，且每种语言类型对应多种不同级别的规则集，系统规则集列表请参见[表5-1](#)。

须知

多个编译语言的规则集不能选择一起同时检查。例如：C#语言规则集不可与其它语言规则集进行混合检查。

表 5-1 预置规则集

语言	系统规则集类型	默认规则集	支持扫描文件后缀	扩展参数支持的编译工具版本	检查引擎
C++	<ul style="list-style-type: none"> 华为C编程规则集 华为C增强编程规则集 华为C++增强编程规则集 华为C++编程规则集 CodeArts Check C推荐规则集 CodeArts Check C++推荐规则集 华为C编码规范规则集 华为CPP编码规范规则集 华为C/C++编码规范规则集 全面检查规则集 关键检查规则集 通用检查规则集 	通用检查规则集	.c/.cc/.cpp /.cxx/.cp/.c+ +/.inc/.inl/ .mm/.h/.hh/.hpp/.hxx/.h++	<ul style="list-style-type: none"> cmake3.15.5-gcc7.5.0 cmake3.15.5-gcc8.3.0 cmake3.10.1-gcc4.8.5 cmake3.10.1-gcc5.5.0 cmake3.10.1-gcc6.4.0 cmake3.10.1-gcc7.3.0 cmake2.8.12-gcc4.8.5 cmake2.8.12-gcc5.5.0 cmake2.8.12-gcc6.4.0 cmake2.8.12-gcc7.3.0 gfortran-gcc4.8.5-cmake2.8.12.2 	<ul style="list-style-type: none"> cppcheck wordstool secbrella clangtidy cmetrics fixbotengine-cxx codemars apiengine oat dupf FossScan

语言	系统规则集类型	默认规则集	支持扫描文件后缀	扩展参数支持的编译工具版本	检查引擎
Java	<ul style="list-style-type: none"> ● 华为Java编程规则集 ● 华为Java增强编程规则集 ● CodeArts Check JAVA推荐规则集 ● 华为编码规范规则集 ● DevCloud推荐规则集 ● 安全检查规则集 ● 移动领域-Android规则集 ● 通用检查规则集 ● 关键检查规则集 ● Google编码规范规则集 ● 全面检查规则集 	通用检查规则集	.java	<ul style="list-style-type: none"> ● maven3.5.3-jdk1.7 ● maven3.5.3-jdk1.8 ● maven3.5.3-jdk1.9 ● maven3.5.3-jdk1.11 ● gradle2.13-jdk1.8 ● gradle2.14.1-jdk1.8 ● gradle3.3-jdk1.8 ● gradle4.1-jdk1.8 ● gradle4.2-jdk1.8 ● gradle4.3-jdk1.8 ● gradle4.4-jdk1.8 ● gradle4.5-jdk1.8 ● gradle4.6-jdk1.8 ● gradle4.7-jdk1.8 ● gradle4.8-jdk1.8 ● gradle4.9-jdk1.8 ● gradle4.8-jdk1.9 ● gradle4.9-jdk1.9 	<ul style="list-style-type: none"> ● sonarqube ● androidlint ● wordstool ● fixbottengine-java ● secbrela ● cmetrics ● codemars ● oat ● Infer_Java ● dupf ● FossScan

语言	系统规则集类型	默认规则集	支持扫描文件后缀	扩展参数支持的编译工具版本	检查引擎
Go	<ul style="list-style-type: none"> CodeArts Check GO推荐规则集 华为编码规范规则集 关键检查规则集 通用检查规则集 全面检查规则集 	通用检查规则集	.go	<ul style="list-style-type: none"> go-1.10.3 go-1.11.6 go-1.12.1 	<ul style="list-style-type: none"> wordstool sonarqube codemars cmetrics codecheckrevive oat secbrella dupf FossScan
Python	<ul style="list-style-type: none"> CodeArts Check PYTHON推荐规则集 华为编码规范规则集 关键检查规则集 通用检查规则集 	通用检查规则集	.py	/	<ul style="list-style-type: none"> wordstool sonarqube pylint codemars cmetrics flake8 secbrella oat dupf FossScan

语言	系统规则集类型	默认规则集	支持扫描文件后缀	扩展参数支持的编译工具版本	检查引擎
JavaScript	<ul style="list-style-type: none"> 华为编码规范规则集 关键检查规则集 通用检查规则集 全面检查规则集 	通用检查规则集	.js/.jsx	/	<ul style="list-style-type: none"> wordstool sonarqube codemars eslint cmetrics oat dupf FossScan
C#	<ul style="list-style-type: none"> 关键检查规则集 通用检查规则集 全面检查规则集 	通用检查规则集	.cs	<ul style="list-style-type: none"> dotnetcore 2.2 dotnetcore 3.0 dotnetcore 3.1 dotnetcore 5.0 dotnetcore 6.0 msbuild15-dotnetframework3.5 msbuild15-dotnetframework4.7.2 msbuild16-dotnetframework4.8 	sonarqube

语言	系统规则集类型	默认规则集	支持扫描文件后缀	扩展参数支持的编译工具版本	检查引擎
TypeScript	<ul style="list-style-type: none"> ● 关键检查规则集 ● 通用检查规则集 ● 全面检查规则集 	通用检查规则集	.ts/.tsx	/	<ul style="list-style-type: none"> ● wordstool ● sonarqube ● apiengine ● codemars ● cmetrics ● oat ● eslint ● dupf ● FossScan
CSS	<ul style="list-style-type: none"> ● 通用检查规则集 	通用检查规则集	.css	/	<ul style="list-style-type: none"> ● wordstool ● sonarqube ● oat ● dupf ● FossScan
HTML	<ul style="list-style-type: none"> ● 通用检查规则集 	通用检查规则集	.html	/	<ul style="list-style-type: none"> ● wordstool ● sonarqube ● oat ● dupf ● FossScan
PHP	<ul style="list-style-type: none"> ● 关键检查规则集 ● 通用检查规则集 	通用检查规则集	.php	/	<ul style="list-style-type: none"> ● wordstool ● sonarqube ● oat ● dupf ● FossScan

语言	系统规则集类型	默认规则集	支持扫描文件后缀	扩展参数支持的编译工具版本	检查引擎
LUA	通用检查规则集	通用检查规则集	.lua	/	anystandard
RUST	通用检查规则集	通用检查规则集	.rs	/	clippy
Shell	通用检查规则集	通用检查规则集	.sh	/	codemars
KOTLIN	通用检查规则集	通用检查规则集	.kt	/	detektout
SCALA	通用检查规则集	通用检查规则集	.scala	/	sonarqube
Arks	通用检查规则集	通用检查规则集	.ets	/	codemars

查看规则集中规则详情

用户可优先了解各个规则集中的规则详情，选择符合检查要求的规则集。可查看的规则详情包含：问题级别、正确示例、错误示例和修复建议等信息。

- 步骤1** 基于项目入口[访问CodeArts Check服务首页](#)。
- 步骤2** 单击“规则集”页签，进入检查规则集列表页面。
- 步骤3** 单击规则集名称，进入到该规则集的规则详情页面可在线查看规则详情。



您也可以单击“导出”，导出该规则集中已启用的规则到本地查看规则详情。

----结束

修改默认规则集

初次使用代码检查服务时，每种编程语言都设置一个默认的检查规则集，如表5-1所示。如果用户需要修改各个语言的默认检查规则集，可参考以下指导操作。

说明

以项目入口访问代码检查服务，规则集列表中才会为每种检查语言预置默认规则集。

步骤1 在“规则集”页面的“语言”列筛选需要配置的语言。

步骤2 单击规则集所在行的“...”，选择“默认规则集”，即可将当前的规则集设置为默认规则集。

----结束

配置代码检查任务使用的规则集

如果用户需要在代码检查任务中使用其他的预置规则集，可参考以下指导操作。如果预置规则集不满足用户的检查需求，可参考[配置代码检查任务自定义规则集](#)。

步骤1 进入代码检查页面，在任务列表中，单击任务名称。

步骤2 进入“代码检查详情”页面，选择“设置 > 规则集”。

步骤3 在“已包含语言”区域，打开目标语言开关。

步骤4 在“启用规则集”区域单击 ，选择需要使用的规则集。

当选中的规则集需配置扩展参数（此处以Java为例）时，单击“检查参数”，配置代码检查扩展参数并开启配置开关，完成后单击“确认”。

检查参数

代码检查扩展参数配置：请您在使用中保护自己的敏感信息。

编译工具

maven

编译工具版本

maven3.5.3-jdk1.8

编译命令

mvn clean compile -Dmaven.test.skip=true -B -U -q

确认

取消

📖 说明

用户购买了代码安全检查增强包后，使用代码安全检查增强包规则集必须设置编译脚本检查参数。

- 如果是ARM类型机器，请在“检查参数”页面的“执行机”中选择对应架构类型。
- 选中的规则集使用了secbrella检查引擎时，必须要设置“检查参数”并开启配置开关，代码检查扩展参数请根据实际情况进行配置。
- 选中的规则集使用了oat检查引擎时，必须要设置“检查参数”并开启配置开关（默认已开启）。其中仓库地址为oat工具-n参数，用于匹配默认策略，可以使用默认值或根据实际情况进行配置。详见[oat开源项目](#)。

----结束

5.1.2 配置代码检查任务自定义规则集

代码检查服务除了可以使用预置规则集类型，用户还可以根据实际需要自定义代码检查规则集，每个规则集中最少设置一条规则。

目前仅支持配置单语言的规则集，即一种规则集只能配置同一种语言类型的检查规则。

📖 说明

代码检查规则集提供了代码安全检查增强包功能，使用前请务必按需购买。购买了代码安全检查增强包后，Java、C++、Go和Python语言可以使用安全度更高的缺陷扫描规则。

创建自定义检查规则集

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 单击“规则集”页签，进入规则集列表页面。

步骤3 单击“新建规则集”，在弹出的“新建规则集”窗口设置规则集名称、描述以及检查语言。

您也可以单击“复制于”下拉框选择复制于已有的规则集，基于复制的规则集配置使用的规则。

步骤4 单击“确定”。

步骤5 根据需要勾选规则名称、设置规则级别，单击右上角“保存”。

如果系统预置的规则不满足用户的使用要求，可[自定义规则](#)使用。

- 单击规则集名称，进入规则集详情页面可在线查看规则集中使用的规则详情。
- 单击规则集名称，进入规则集详情页面，单击“导出”按钮可导出该规则集中已启用的规则到本地查看规则详情。
- 单击规则集名称，进入规则集详情页面，单击“修改基本信息”，可修改规则集名称或描述信息。

📖 说明

- “检查语言”不支持修改。
- 系统预置规则集无法修改。
- 只有创建者可以修改自定义规则集。
- 单击规则集所在行的 \dots ，选择“删除”即可删除检查规则集。

📖 说明

- 只有创建者可以删除自定义规则集。
- 系统规则集和已被使用的自定义规则集无法删除。
- 如果需要删除已被使用的规则集，需将关联使用该规则集的代码检查任务删除或将代码检查任务关联到其它规则集。

----结束

配置代码检查服务使用自定义规则集

步骤1 在代码检查详情页选择“设置 > 规则集”。

步骤2 如果创建代码检查任务后，代码仓的代码有修改，则需单击“已包含语言”所在行重新获取代码仓语言，将目标语言的开关设置为开启状态。

步骤3 单击 ，选择已创建的自定义规则集。

----结束

自定义规则

- 最多支持创建10条自定义规则。
- 使用自定义规则扫描代码最多支持扫描10万行。仅适用于使用的规则集中只包含自定义规则。
- 使用自定义规则扫描10万行代码行最多时长1.5小时。仅适用于使用的规则集中只包含自定义规则。

如果需要支持扫描更多行数的代码或需要创建更多数量的规则，可咨询华为技术支持。

步骤1 单击“规则”页签。

步骤2 单击“新建规则”，参考表5-2配置参数。

表 5-2 自定义规则参数配置说明

参数名称	参数说明	是否必填
规则名称	创建的规则名称，可自定义。 <ul style="list-style-type: none"> • 支持中英文，数字，点，下划线“_”和连接符“-” • 字符长度范围为1~128 	是
工具规则名称	默认填充为规则源码文件名，不可手动修改。	是
检查工具	该检查规则使用的检查工具，当前仅支持SecBrella。	是
语言	该规则检查的编译语言，当前仅支持Java。	是

参数名称	参数说明	是否必填
规则源码	该规则的源码文件。 由用户创建“.kirin”文件编写dsl规则，然后通过运行本地插件在“OutputReport”目录下生成“SecH_自定义规则名.json”的源码文件，上传json文件即可。使用插件生成规则文件可参考 插件使用指南 。	是
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。	是
标签	为该规则设置标签，便于使用时区分该规则使用场景。 添加多个标签时，需使用英文逗号分隔。	否
描述	对该规则的使用描述。如果描述的内容需要粘贴代码，代码内容支持MarkDown格式。字符数限制为0~10000。	是
正确示例	该规则对应的正确代码示例。代码内容支持使用MarkDown格式呈现。字符数限制为0~10000。	否
错误示例	不满足该规则的错误代码示例。代码内容支持使用MarkDown格式呈现。字符数限制为0~10000。	否
修复建议	对于该规则检查出的问题修改建议。如果修复建议中需要粘贴代码，代码内容支持MarkDown格式。字符数限制为0~10000。	否

步骤3 单击“新建规则”，完成创建。

----结束

5.2 配置代码检查任务质量门禁

根据质量要求设置门禁阈值，即可了解您的项目是否已准备好投入生产。例如，实际的检查结果超出设置的质量阈值，则表示当前项目还不具备生产条件。

门禁质量设置支持租户级、项目级以及任务级，优先级为租户级 > 项目级 > 任务级。入口分别如下：

- 租户级：进入代码检查服务首页，选择“配置中心”，默认显示质量门禁详情。
- 项目级：进入项目详情页面，选择“代码 > 代码检查 > 配置中心 > 质量门禁”。
- 任务级：进入代码检查详情页面，选择“设置 > 质量门禁”。

说明

项目级和任务级支持事件[通知设置](#)。

本节以任务级门禁设置为例介绍操作步骤。

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 进入代码检查页面，在任务列表中，单击任务名称。

步骤3 进入“代码检查详情”页面，选择“设置 > 质量门禁”。

步骤4 在“质量门禁”区域，根据质量要求开启门禁开关，并设置门禁阈值。

图 5-1 配置质量门禁

门禁项	运算逻辑	门禁阈值	门禁开关
致命问题数	≤	0	<input checked="" type="checkbox"/>
严重问题数	≤	0	<input checked="" type="checkbox"/>
一般问题数	≤	0	<input type="checkbox"/>
提示问题数	≤	0	<input type="checkbox"/>

保存

步骤5 完成后，单击“保存”。

您可单击“开始检查”执行检查任务并查看检查结果。

若问题级别数超过设定阈值，任务概览页面提示门禁结果为“不通过”，需要您根据检查结果优化代码。

----结束

5.3 配置代码检查任务定时执行

根据实际需要，设置代码检查任务的执行计划，定时执行默认分支的检查任务。

说明

- 建议在非流水线使用情况下进行配置。
- 配置当天时间间隔必须在5分钟以上。

配置执行计划

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 进入代码检查页面，在任务列表中，单击任务名称。

步骤3 进入“代码检查详情”页面，选择“设置 > 执行计划”。

步骤4 开启“定时执行”开关，并设置自动执行时间。

图 5-2 配置执行计划

执行计划

定时执行

* 自动执行时间:

日 一 二 三 四 五 六 00:00

保存

步骤5 完成后，单击“保存”。

----结束

5.4 配置代码检查任务消息通知

您可以根据实际需求，设置代码检查任务事件类型的通知方式。

例如，当检查任务的门禁失败时，可以设置以服务动态或邮件方式通知用户，并可配置通知的接收者角色和自定义成员。若不配置接收者角色及成员，门禁失败时向任务执行者发送通知；若配置，则向任务执行者、配置的接收者角色以及自定义成员发送通知。

通知设置支持项目级和任务级，入口分别如下：

- 项目级：进入项目详情页面，选择“代码 > 代码检查 > 配置中心 > 通知管理”。
- 任务级：进入代码检查详情页面，选择“设置 > 通知管理”。本节以任务级通知设置为例介绍操作步骤。

配置代码检查任务消息通知

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 进入代码检查页面，在任务列表中，单击任务名称。

步骤3 进入“代码检查详情”页面，选择“设置 > 通知管理”。

步骤4 根据需要开启或关闭事件类型的通知状态。

- 开启服务动态通知，即通知当前任务所属项目的特定成员，可在消息中心查看消息。
- 开启邮件通知，即通知当前任务所属项目的特定成员，可在邮件中查看消息。

----结束

配置代码检查服务钉钉通知

步骤1 进入钉钉群，找到“群设置 > 智能群助手”，然后添加机器人（选择自定义类型）。

步骤2 填写机器人名字，选择群组，完成安全设置（需勾选“加签”，并单击加签文本框旁的“复制”获取加签密钥）。

步骤3 已阅读并同意相关协议后，单击“完成”。

步骤4 单击Webhook文本框旁的“复制”，获取钉钉Webhook地址。

步骤5 将“钉钉通知”开关设置为  状态。

步骤6 选择通知场景，并填写Webhook地址，单击“测试”确保Webhook地址可用。

步骤7 单击“保存”。

----结束

配置完成后，当任务运行结果满足事件类型时，代码检查服务会发送消息到指定的钉钉群。

配置代码检查服务企业微信通知

步骤1 在企业微信群群里添加群机器人。（以手机客户端为例，详细指导请参考[如何设置群关系机器人](#)。）

1. 运行企业微信客户端，选中需要接受消息推送的群聊，单击右上角三个点按钮。
2. 单击“群机器人”。
3. 单击“添加”。
4. 单击“新建”。
5. 填写机器人名字，单击“添加”。
6. 单击Webhook文本框旁的“复制”获取企业微信Webhook地址。

步骤2 在代码检查任务中配置企业微信群通知服务。

1. 将“企业微信通知”开关设置为  状态。
2. 设置通知场景，并填写[步骤1](#)中获取的“Webhook URL”。
3. 单击“测试”，验证Webhook地址是否正确可用。确认Webhook URL正确后，单击“保存”。

当代码检查任务完成时，代码检查服务会发送检查结果到指定的企业微信群。

----结束

5.5 配置代码检查任务检查模式

您可以根据实际需求设置检查模式和检查目录，提高检查效率。

须知

C#语言和secbrella引擎均不支持MR合入。

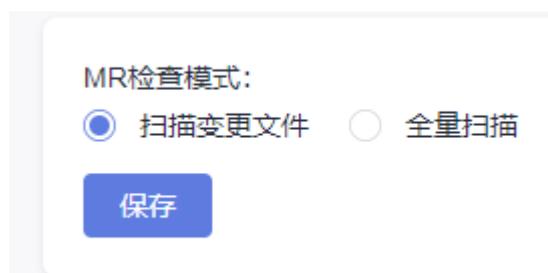
设置检查模式

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 进入代码检查页面，在任务列表中，单击任务名称。

步骤3 进入“代码检查详情”页面，选择“设置 > 检查模式”。

步骤4 在“MR检查模式”区域，选择“扫描变更文件”或“全量扫描”。默认选择“扫描变更文件”。



步骤5 完成后，单击“保存”。

----结束

设置检查目录

支持检查代码仓指定目录。

- 不配置检查目录时，检查代码仓中所有目录下的代码文件。
- 配置检查目录时，只检查配置目录中的文件。

检查目录 

输入需要检查的目录，多个目录使用逗号隔开

保存

说明

支持配置多个检查目录，多个目录需要用英文逗号分隔。例如：

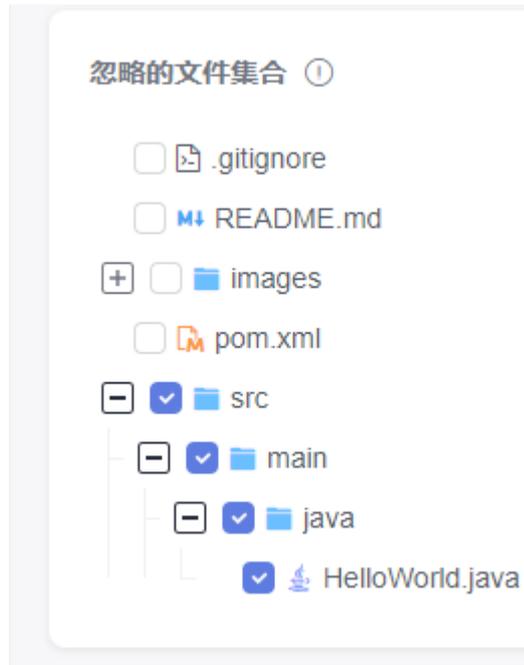
- 如果要检查dir1和dir2，检查目录配置为“dir1,dir2”。
- 如果要检查dir1目录下的dir3，检查目录配置为“dir1/dir3”。
- 配置项为目录且不支持文件，默认检查代码仓中所有代码。
- 配置的目录层级中不能包含"."开头的目录。
- 如果检查目录输入框路径为空，则默认全部检查。如果有指定检查目录，则只检查检查目录中的文件，如果指定的检查目录中有勾选[设置忽略文件](#)，则被忽略的文件不参与检查。
- 检查目录只生效于不需要编译的文件。例如，A文件编译后产生了B文件，则不会输出B文件上的告警。

设置忽略的文件集合

代码检查服务支持用户以任务为单元，设置代码检查的文件范围。

- 不勾选文件路径时，则检查所选仓库分支的所有代码。
- 勾选指定文件路径时，则该指定路径下的文件将排除检查。

在“忽略的文件集合”区域，勾选无需进行检查的文件路径。设置完成后，执行代码检查任务，将按照设置排除选中的文件路径执行检查。



说明

勾选的忽略文件对需要编译的文件和不需要编译的文件均生效。

- 不需要编译的文件：勾选忽略文件后，被忽略的文件不参与检查。
- 需要编译的文件：例如A文件与B文件互为依赖，即使把A文件忽略，但A文件依然会参与检查前的编译过程，编译完成后不参与检查。

5.6 查看徽标状态

执行完代码检查任务后，您可通过徽标状态查看代码健康度，并标识在代码仓库中。

代码检查服务中设置各个告警级别的影响度如下：

致命：5

严重：3

一般：1

提示：0.1

健康度 = $\text{SUM} \sum (\text{告警数量} * \text{告警级别影响度}) / \text{代码量}$ ，各个健康度对应的标识如下：

A: 健康度 ≤ 0.001

B: $0.001 < \text{健康度} \leq 0.005$

C: $0.005 < \text{健康度} \leq 0.01$

D: 健康度 > 0.01

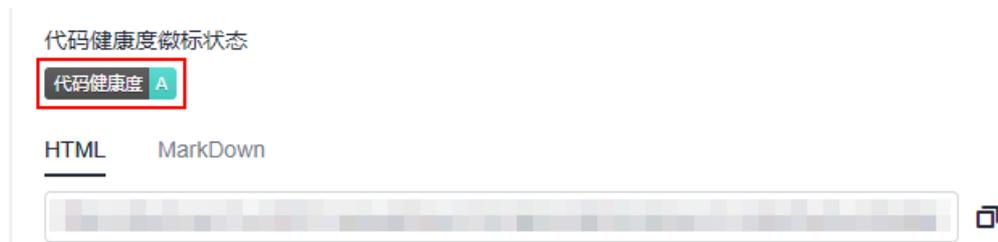
查看徽标状态

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 在任务列表中，单击任务名称。

步骤3 进入“代码检查详情”页面，选择“设置 > 徽标状态”。

当前仅支持“HTML”和“Markdown”两种查看方式。



----结束

5.7 配置代码检查任务集成服务

当代码托管中有分支合并时，在代码检查服务中可以设置自动检查代码检查任务，并生成一个子任务，即代码仓库中有分支合并，就会重新生成一个代码检查子任务。

配置 MR 检查状态

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 进入代码检查页面，在任务列表中，单击任务名称。

步骤3 进入“代码检查详情”页面，选择“设置 > 集成服务 > CodeArts Repo”。

步骤4 选择是否勾选“MR检查状态”，勾选即表示开启代码评审（MR）的自动化检查。

步骤5 当所在任务的代码仓库中有分支合并时，代码检查服务会自动生成一个代码检查子任务。

1. 进入代码检查任务详情页面。
2. 单击，在代码托管中新建MR请求。

说明

合并分支相关操作请参见[代码托管用户指南](#)。

3. 单击 master ▾，通过下拉框中，可查看所有的MR触发代码检查的历史记录。

----结束

配置代码提交时执行

步骤1 勾选“代码提交时执行”，表示对代码仓提交代码时会触发执行代码检查任务。

说明

仅对已创建任务的分支有效。

步骤2 单击“保存”。

----结束

5.8 配置代码检查任务自定义环境

当使用代码检查服务时，若常用的编译构建环境缺少您需要的依赖包、工具，无法满足您的使用需求时，您可通过自定义构建环境提供的基础镜像执行代码检查任务。

前提条件

已将自定义环境推送到镜像仓中，具体操作请参见[制作镜像并推送到SWR仓库](#)。

配置自定义镜像

- 步骤1** 基于项目入口[访问CodeArts Check服务首页](#)。
- 步骤2** 进入代码检查页面，在任务列表中，单击任务名称。
- 步骤3** 在代码检查详情页面，选择“设置 > 自定义环境”。
- 步骤4** 在“自定义镜像”中，输入以“域名称/组织名称/仓名称:tag名称”格式的内容或容器镜像服务SWR内镜像链接。
- 步骤5** 完成后，单击“保存”。

----结束

配置私有依赖仓扩展点

您可通过配置私有依赖库作为私有依赖下载源，当代码检查任务需要连接到远程第三方服务时，进行数据获取。

- 步骤1** 在“配置私有依赖仓扩展点”区域，单击“扩展点管理”，具体操作请参见[服务扩展点](#)。



- 步骤2** 配置完成后，单击“扩展点”下拉框，选择新建的扩展点。

📖 说明

最多只能添加7个扩展点。

- 步骤3** 完成后，单击“保存”。

----结束

配置检查前置命令

您可以在执行代码检查之前，根据实际情况设置“检查前置命令”，运行该命令可以搭建用户自定义的扫描环境。

步骤1 在“检查前置命令”区域，输入检查前置命令。

步骤2 单击“保存”。

----结束

5.9 配置代码检查任务高级选项

配置新问题起始时间

在每个任务的基础上可以设置新问题起始时间。起始时间设置之后要重新扫描，问题的发现时间大于设置时间将被划分为新问题。如果没有设置起始时间，连续两次的检查结果对比，新增的就是新问题数。

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 进入代码检查页面，在任务列表中，单击任务名称。

步骤3 进入“代码检查详情”页面，选择“设置 > 高级选项”。

步骤4 在“新问题起始时间”区域，单击设置新问题起始时间。

说明

默认起始时间为当前任务第一次扫描成功的时间，起始时间之后扫描出的所有问题都将被划分为新问题。

----结束

配置源文件编码

设置文件的编码方式，源文件编码会影响系统识别的源码内容的完整性，系统默认编码为UTF-8。

步骤1 在“源文件编码”区域，单击下拉框选择源文件编码。



步骤2 完成后，提示修改成功。

----结束

配置问题责任人精准匹配

您可根据实际需求，选择是否开启问题责任人精准匹配，将问题分配给不同的成员处理。

步骤1 在“问题责任人精准匹配”区域，开启开关更新问题责任人匹配策略。

步骤2 任务检查完成后，您可在“代码问题”页签下通过选择“负责人”过滤问题。

----**结束**

6 执行代码检查任务

用户可以对目标代码检查任务执行检查，及时发现源代码中的代码问题。

执行代码检查有两种方式：

- 手动执行：创建完代码检查任务后手动执行代码检查任务。本章节介绍的为手动执行代码检查任务的操作指导。
- 定时执行：多用于日常执行代码检查，配置详情可参考[配置代码检查任务定时执行](#)。

前提条件

- 已[创建代码检查任务](#)。
- 如果创建的是[Repo代码源检查任务](#)，则需要有代码仓库的访问权限。参考[成员管理](#)，可添加代码仓访问权限。

执行代码检查任务

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 单击代码检查任务所在行的 ，根据页面提示等待任务执行完成。

----结束

检查代码仓其它分支的代码

在代码检查详情页面可以手动切换“已检查分支列表”和“未检查分支列表”进行检查。

步骤1 单击代码检查任务名称。

步骤2 在“代码检查详情”页面，默认显示“概览”页签，单击  master ▼ 可以切换不同分支。

步骤3 切换目标分支后，单击“开始检查”，等待代码检查任务执行完成。

再次单击代码检查任务所在行的  执行检查任务时，检查的代码分支会自动变更为创建检查任务时设置的默认分支。

---结束

7 查看代码检查结果

前提条件

您已[创建并执行代码检查任务](#)。

查看代码检查详情

步骤1 基于项目入口[访问CodeArts Check服务首页](#)。

步骤2 进入代码检查页面，搜索目标任务。在任务列表页，可以查看检查任务的状态。

- 新问题：表示本次代码检查任务执行后新增的问题。
- 未解决：表示该代码检查任务未解决的问题。
- 已解决：表示该代码检查任务已解决的问题。

步骤3 单击检查任务名称，进入代码检查详情概览页，可查看的信息如下：

- 原始代码行数。
- 门禁结果：检查结果分为“通过”和“不通过”，同时会展示“致命问题数”和“严重问题数”。

代码检查是否通过是由设置的质量门禁值判断，门禁值设置请参考[配置代码检查任务质量门禁](#)。

- 问题数：检查出来的代码质量问题数。
- 延迟上线问题数：由于检查引擎的能力提升，出现的新问题数。
- 未解决新问题数：当前的代码检查结果中，未解决的问题个数。
- 已解决问题数：当前的代码检查结果中，已解决的问题个数
- 代码平均圈复杂度：代码的平均[圈复杂度](#)。
- 代码重复率：代码的[重复率](#)。
- NBNC代码行：非空非注释的代码行数。
- 问题数最多Top10检查规则：问题数Top10的检查规则以及每条规则对应的问题数。
- 待处理问题严重程度：待处理的各个等级问题数量分布情况。
- 问题指派分配：检查出来的代码问题修改分配情况。如果已指派的数量与问题数不一致，说明存在某代码问题还未分配修改责任人。

- 历史趋势：默认展示近一个月内该代码检查任务的检查结果变化趋势，包括问题数、圈复杂度和重复率。

----结束

查看代码问题

可查看代码问题详情有两种查看方式：导出至本地和在线查看。

- 导出至本地。

方式一：

- 返回至代码检查任务列表页，勾选代码检查任务名称前的复选框，可选择的任务数为1~20。
- 选择“更多操作 > 导出任务告警”。
- 在弹出的窗口中选择需要导出的选项，单击“开始导出”。
- 单击“查看下载”，即可查看导出进度。
- 待导出完成后，单击“下载”，可将代码检查问题信息导出至本地。

方式二：

- 单击“代码问题”页签。
- 选择“全量操作 > 全量导出”，可将代码检查问题信息导出至本地。

- 在线查看。

- 单击“代码问题”页签，可以在“过滤器”区域，根据过滤条件筛选问题，可设置的过滤条件如下表。

过滤条件	说明
问题级别	分为致命、严重、一般、提示四个级别。
问题状态	包括待处理、已解决、已忽略、已修复四种状态。
延迟问题	查看当前检查出的问题中正式版本问题总数和延迟上线版本问题总数。
检测时间	根据检测时间区分新问题和存量问题。 说明 设置 新问题起始时间 ，问题发现时间在起始时间之后的则为新问题。
文件名	按照源文件名称筛选代码问题。
语言	按照检查的语言筛选代码问题。
规则	按照问题命中的规则筛选代码问题，便于用户一次性解决同一类问题。
标签	按照规则的用户标签筛选代码问题。
负责人	按照问题分配负责人筛选代码问题。 说明 代码检查过程中，开启精准匹配自动将新问题分配给问题代码行的最后一次提交者。
CWE	按照安全类规则名筛选代码问题。

- b. 设置完筛选条件后，在右侧区域对检查问题可进行以下操作。
- 如果该问题已解决，单击“待处理”选择“已解决”；如果该问题无需处理，单击“待处理”选择“忽略问题”，忽略问题时需提交评论。
 - 单击“负责人”，可将此问题提交至其他负责人。
 - 单击“问题帮助”，可查看该问题对应的规则详情。
 - 单击“查看上下文”，在问题详情页面中，可修改问题状态、转移负责人和根据“问题帮助”单击前往代码仓对代码进行修改。
 - 单击“创建问题单”，提交问题单给指定的责任人。该功能仅Scrum类型项目支持。
 - 单击“查看问题单”，查看问题单详情。该功能仅Scrum类型项目支持。
 - 单击“生成报告”，可将代码问题生成PDF格式的文件报告。
 - 单击“全量操作 > 全量修改问题状态”，可修改筛选出来的全部问题的问题状态。

说明

如果代码问题数是0，则不显示该内容。

- 单击“全量操作 > 全量转移责任人”，可将筛选出来的全部问题转移给指定责任人。

查看代码度量

代码度量可准确的反映提交代码的质量，助力用户及时发现并修改问题，提升研发效能。代码度量包括已下两种类型的问题：

- 圈复杂度：代码的复杂度。

说明

扫描Shell语言代码时，度量结果不支持圈复杂度计算。

单击文件名称进入文件详情页面，单击可进入到对应的代码仓修改代码。

```
worker/src/main/java/worker/Worker.java
1 package worker;
2
3 import redis.clients.jedis.Jedis;
4 import redis.clients.jedis.exceptions.JedisConnectionException;
5 import java.sql.*;
6 import org.json.JSONObject;
7
8 class Worker {
9     public static void main(String[] args) {
10        try {
11            Jedis redis = connectToRedis("redis");
12            Connection dbConn = connectToDB("db");
13
14            System.err.println("Watching vote queue");
15
16            while (true) {
17                String voteJSON = redis.blpop(0, "votes").get(1);
18                JSONObject voteData = new JSONObject(voteJSON);
19                String voterID = voteData.getString("voter_id");
20                String vote = voteData.getString("vote");
21
22                System.err.printf("Processing vote for '%s' by '%s'\n", vote, voterID);
23                updateVote(dbConn, voterID, vote);
24            }
25        }
26    }
27 }
```

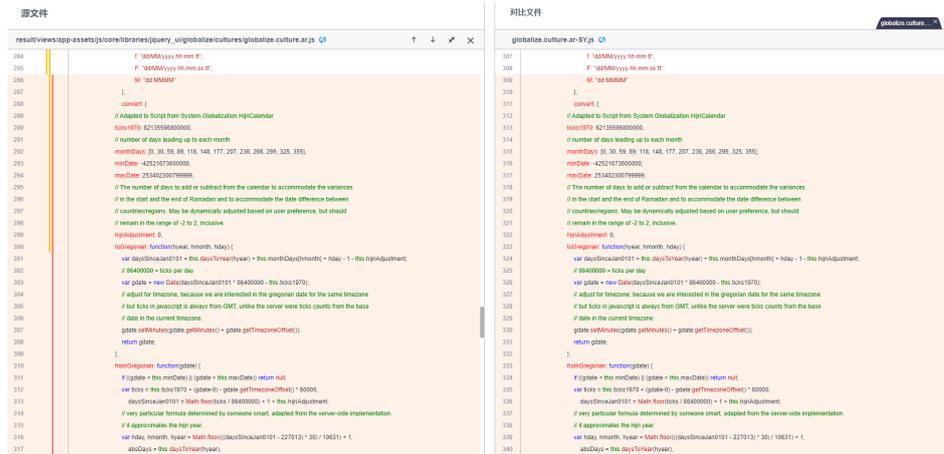
- 重复率：代码检查服务通过重复块、重复行、重复率来识别被测代码的重复情况。

修改重复率问题操作指导如下：

- a. 单击文件名称进入文件详情页面。
- b. 文件详情页面左侧的竖线标识了重复的代码块，单击标识竖线，将会显示与该段代码重复的文件列表。

```
result/views/app-assets/js/core/libraries/jquery_ui/globalize/cultures/globalize.culture.ar.js
319 // year is just approximate, it may need adjustment up or down by 1.
320 if (daysSinceJan0101 < absDays) {
321     hyear--;
322     absDays -= daysInYear;
323 }
324 else if (daysSinceJan0101 === absDays) {
325     hyear--;
326     absDays = this.daysToYear(hyear);
327 }
328 else {
329     if (daysSinceJan0101 > (absDays + daysInYear)) {
330
331         Duplicated By
332         result/views/app-assets/js/core/libraries/jquery_ui/globalize/cultures/globalize.culture.a...
333         Lines: 309 - 382
334
335         result/views/app-assets/js/core/libraries/jquery_ui/globalize/cultures/globalize.culture.a...
336         Lines: 309 - 382
337
338         result/views/app-assets/js/core/libraries/jquery_ui/globalize/cultures/globalize.culture.a...
339         Lines: 92 - 165
340
341         result/views/app-assets/js/core/libraries/jquery_ui/globalize/cultures/globalize.culture.a...
342
343         hmonth++;
344     }
345     hmonth--;
346     hday = daysIntoYear - this.monthDays[hmonth];
347     return [hyear, hmonth, hday];
348 },
349 daysToYear: function(year) {
350     // calculates how many days since Jan 1, 0001
351     var yearsToYear30 = Math.floor((year - 1) / 30) * 30,
352         yearsInto30 = year - yearsToYear30 - 1,
353         days = Math.floor((yearsToYear30 * 10631) / 30) + 227013;
```

- c. 单击文件名称，可对比查看两个文件的重复代码块。



d. 单击  可进入到对应的代码仓修改代码。

查看检查日志

单击“检查日志”页签，可查看检查步骤日志和检查参数。

图 7-1 步骤日志

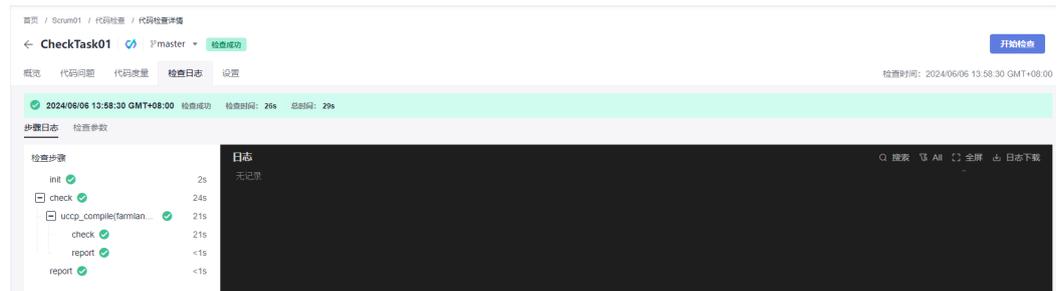
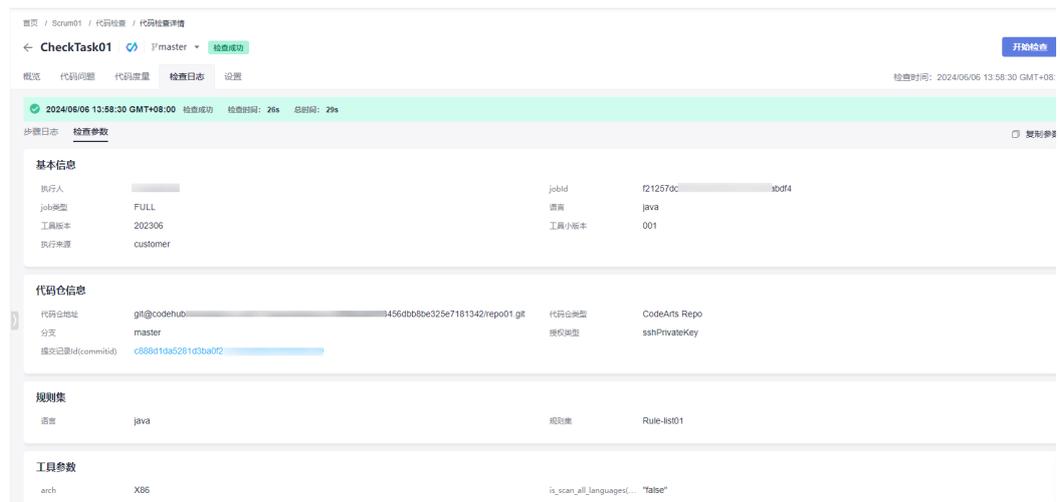


图 7-2 检查参数



8 变更代码检查服务规格

变更代码检查服务套餐规格

代码检查服务支持变更套餐规格，变更影响请参见[变更配置后对计费的影响](#)。

步骤1 登录代码检查控制台。

步骤2 找到代码检查服务套餐，单击“变更”。

步骤3 根据需要选择变更规格、变更类型，勾选同意声明，单击“下一步”。

说明

若变更类型选择“续费变更”，则还需要选择续费时长。

步骤4 确认订单内容：若需要修改，单击“上一步”。

步骤5 若确认无误，继续根据页面提示完成支付。

----结束

变更并发数

代码检查并发扩展支持变更并发数。

步骤1 登录代码检查控制台。

步骤2 找到待变更的资源扩展记录，单击“变更”。

步骤3 根据需要设置并发数，勾选同意声明，单击“下一步”。

步骤4 确认订单内容：若需要修改，单击“上一步”。

步骤5 若确认无误，继续根据页面提示完成支付。

----结束

购买资源扩展

代码检查支持并发扩展，详情介绍请参见[资源扩展包](#)。

步骤1 登录代码检查控制台。

步骤2 找到代码检查服务套餐，单击“Check扩展资源”区域的“购买”。

步骤3 根据需要选择区域、代码检查并发数、购买时长、是否自动续费，勾选同意声明后单击“下一步”。

步骤4 确认订单内容：若需要修改，单击“上一步”。

步骤5 若确认无误，继续根据页面提示完成支付。

----结束

9 查询审计日志

云审计服务是安全解决方案中专业的日志审计服务，记录了应用与数据集成平台的相关操作事件，方便您日后的查询、审计和回溯。

支持审计日志的操作

表 9-1 云审计服务支持 CodeArts Check 服务操作

操作名称	资源类型	事件名称
创建代码检查任务	task	create
修改代码检查任务	task	modify
删除代码检查任务	task	delete
执行代码检查任务	job	create
取消执行代码检查任务	job	modify_job
关注代码检查任务	followingTask	create_followingTask
取消关注代码检查任务	followingTask	delete_followingTask
修改规则集信息	ruleset	modify_ruleset
修改规则集的规则	ruleset	modify_ruleset
删除规则集	ruleset	delete_ruleset
配置默认规则集	ruleset	set_default_ruleset
导出规则集表格	ruleset	export_ruleset
导出问题表格	defectExcel	export_defectExcel
下载问题表格	defectExcel	downloadAndExport_defectExcel
修改问题状态	issueStatus	update_issueStatus
修改问题责任人	issueOwner	update_issueOwner

操作名称	资源类型	事件名称
修改问题评论	issueComment	update_issueComment

查看审计日志

用户需要在云审计服务CTS的管理控制台查询CodeArts Check服务的事件列表。详情请参考[查看审计事件](#)。

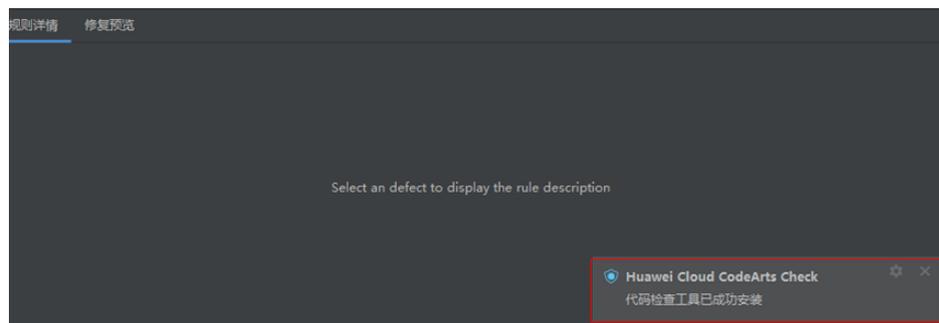
10 参考

10.1 CodeArts Check 插件使用指南（IntelliJ IDEA 版本）

本插件致力于守护开发人员代码质量，成为开发人员的助手和利器。秉承极简、极速、即时看护的理念，提供业界规范的检查、代码风格一键格式化及代码自动修复功能。

安装 CodeArts Check 插件

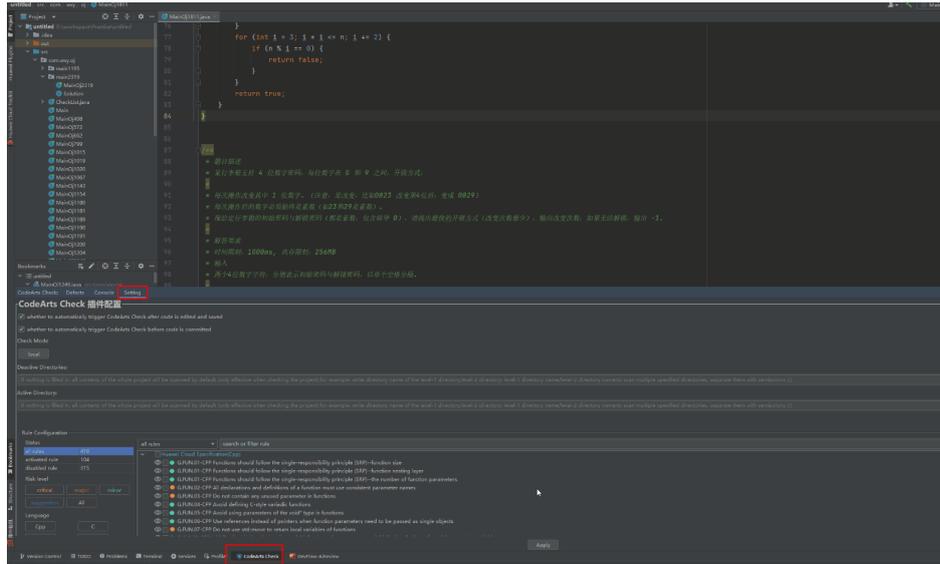
- 步骤1** 在IDEA IDE编辑器顶部菜单栏选择“File > Settings”，进入IDEA设置界面。
- 步骤2** 左侧菜单中单击“Plugins”，右侧窗口切换到“Marketplace”。
- 步骤3** 在搜索输入框内搜索“Huawei Cloud CodeArts Check”或“CodeArts Check”可看到CodeArts Check代码检查插件。
- 步骤4** 单击“安装”，IDEA会重启，重启之后，会自动开始下载引擎包和JDK。
- 步骤5** 下载完成之后，在IDEA IDE编辑器右下角会有下载安装成功的提示信息。



----结束

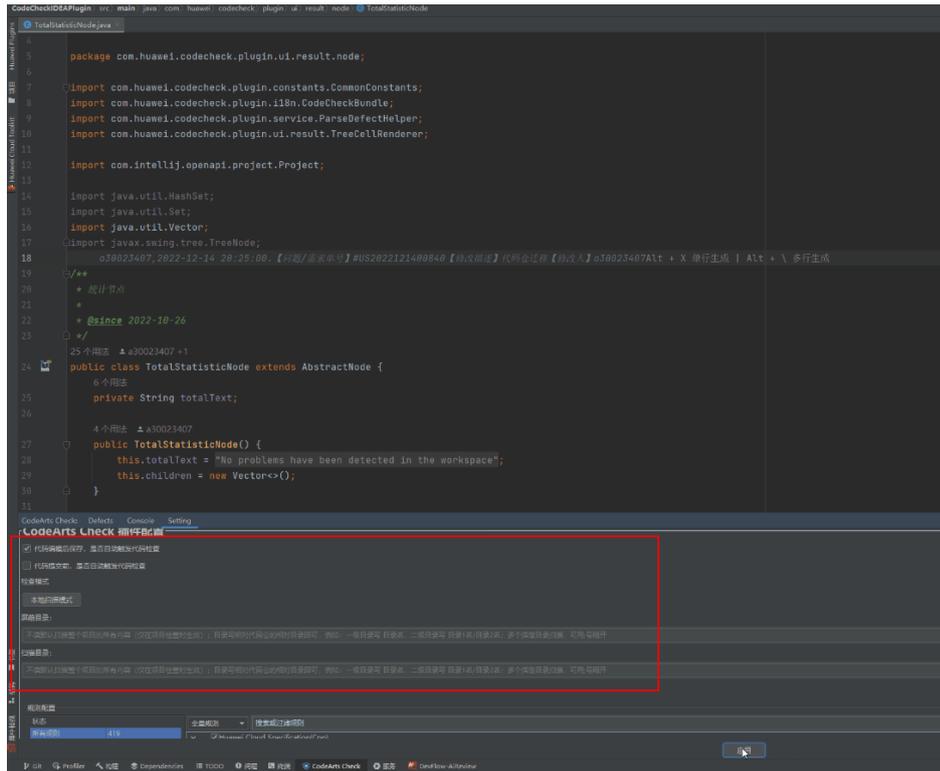
配置 CodeArts Check 插件

- 步骤1** 编辑器下方切换到“CodeArts Check”窗口，在底部工具栏单击“Setting”，弹出配置窗口。

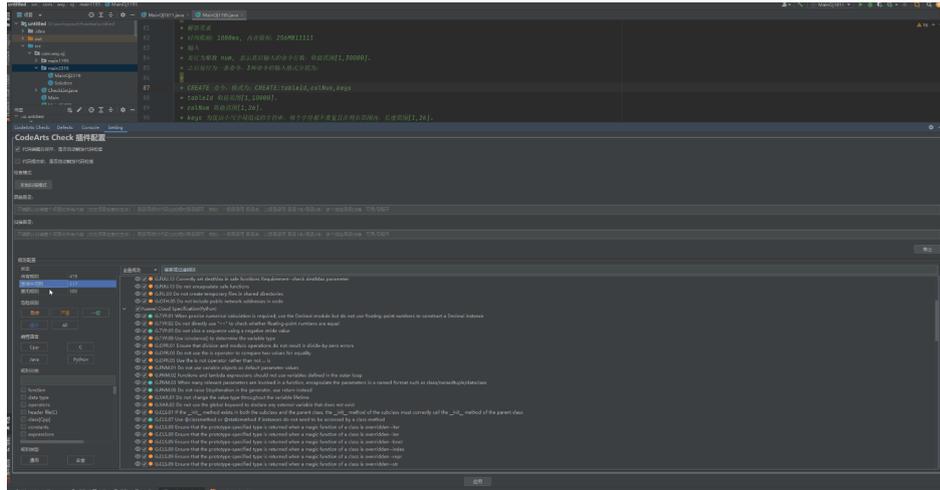


步骤2 单击配置窗口中“规则”左侧的图标可以查看规则详细信息。

步骤3 根据实际需要确定勾选是否自动触发代码检查，然后输入检查的屏蔽目录和指定的扫描目录。



步骤4 可以查看当前被激活/应用到的规则（包括状态、危险级别、编程语言、引擎标签、规则类型）或者根据规则集筛选规则（所有规则集/推荐规则集）。



步骤5 配置后，单击“应用”。

----结束

使用 CodeArts Check 插件检查和修复文件

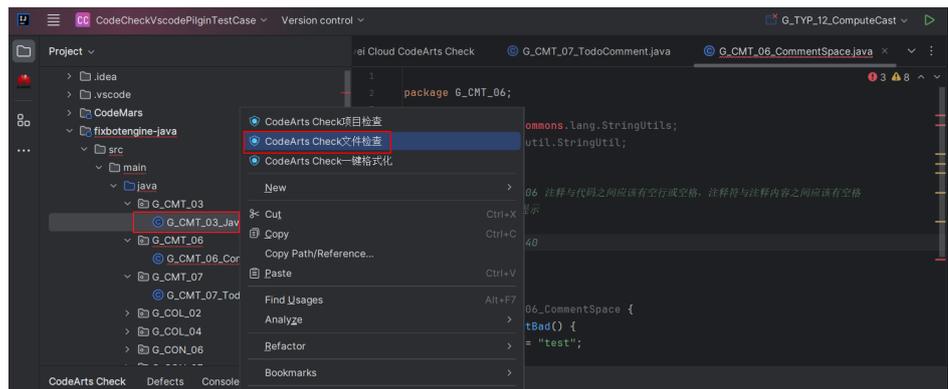
以Java项目为例。

本地选择一个Java项目。支持对单个Java文件或多个Java文件进行扫描检查，或对整个项目进行工程检查，还可以通过一键格式化对文件进行自动代码风格问题纠正等，详细介绍请参见[CodeArts Check插件检查功能介绍](#)、[CodeArts Check一键格式化及自动修复功能介绍](#)和[CodeArts Check规则配置功能介绍](#)。

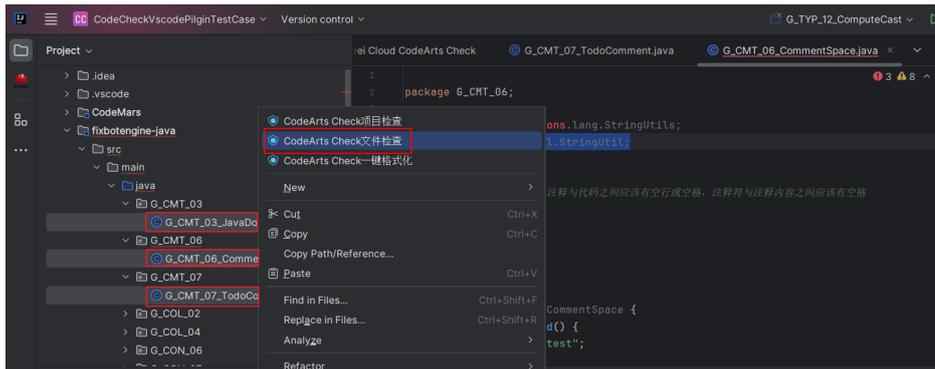
CodeArts Check 插件检查功能介绍

- 扫描检查单文件、多文件、整个项目并查看扫描结果

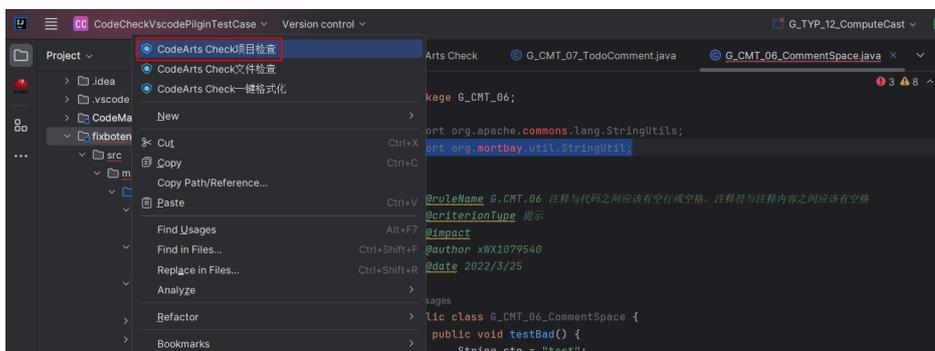
- 单文件扫描检查：在当前打开的文件代码编辑区，单击鼠标右键，选择“CodeArts Check文件检查”。或者在项目的资源管理器界面里，鼠标移动到需要扫描的文件，单击鼠标右键，选择“CodeArts Check文件检查”。



- 多文件扫描检查：在项目的资源管理器里，按住Ctrl键，鼠标左键选择需要扫描的几个文件，单击鼠标右键，选择“CodeArts Check文件检查”。

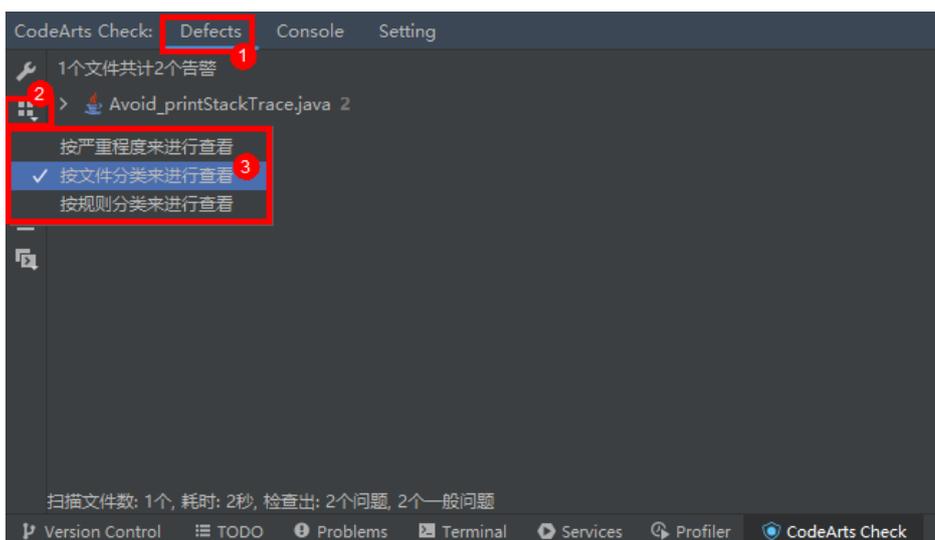


- 整个项目扫描检查：在当前打开的文件代码编辑区，单击鼠标右键，选择“CodeArts Check项目检查”。或者在项目的资源管理器界面里，单击鼠标右键，选择“CodeArts Check项目检查”。

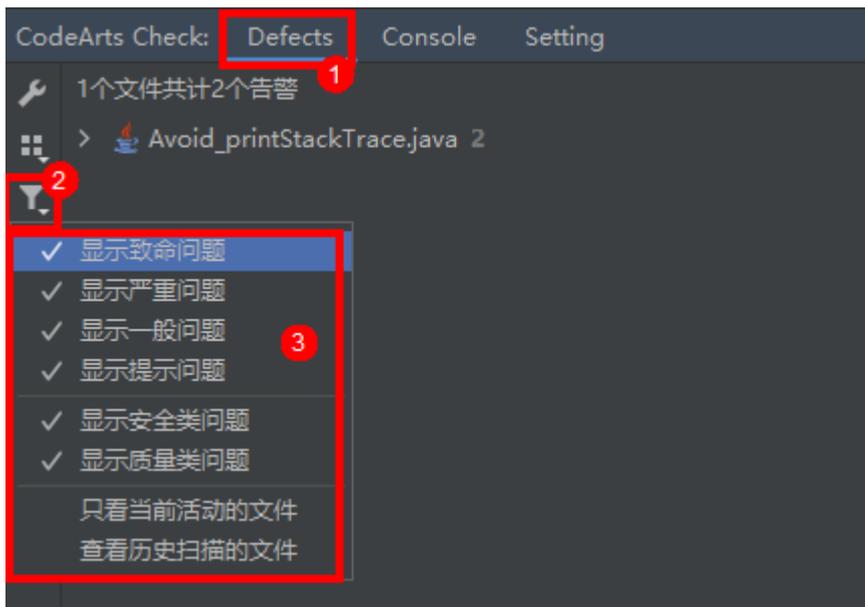


- 告警帮助提供告警详细描述、严重级别、告警类型、正确示例、错误示例及修复建议，您可以根据“严重程度/文件分类/规则分类”筛选查看告警。

- 按文件分类查看：打开CodeArts Check窗口，选择“Defects”页签，在检查结果区域左侧工具栏，单击 （分类），选择“按文件分类进行查看”。



- 按严重级别后告警类型查看：打开CodeArts Check窗口，选择“Defects”页签，在检查结果区域左侧工具栏，单击 ，根据需要选择显示问题的类别。



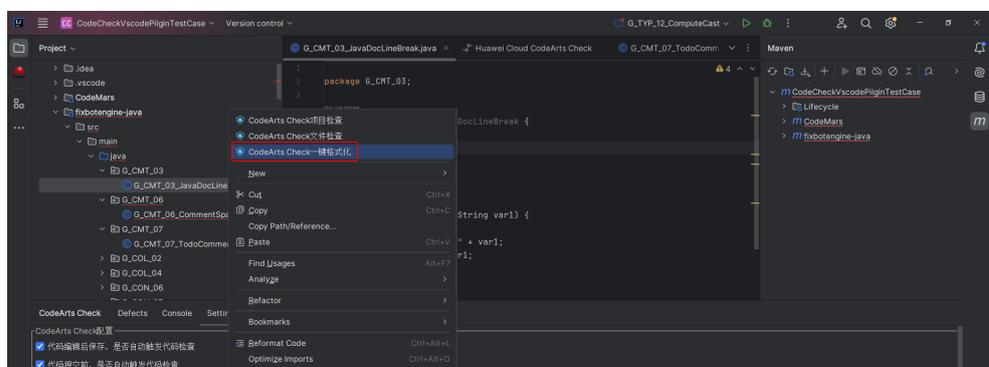
- 根据告警指导修改后，插件将实时自动检查，自动更新告警信息
用户根据插件告警指导修改后，插件将实时自动进行检查，自动更新告警信息。



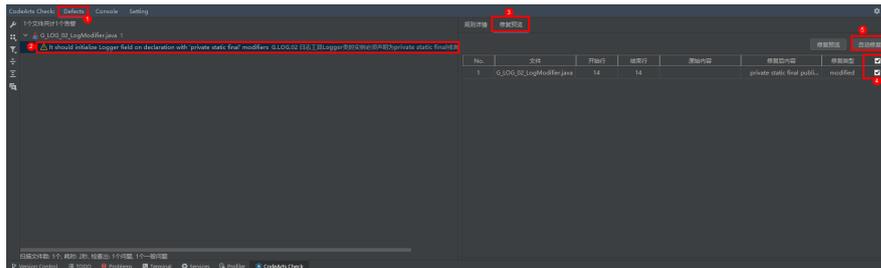
CodeArts Check 一键格式化及自动修复功能介绍

- 一键格式化：支持对单文件/多文件/文件夹/整个项目进行一键格式化。一键格式化可帮助您自动解决项目里基础的代码规范问题，如：多余的空格或空行。操作方法如下：

在当前打开的文件代码编辑区或在项目的资源管理器界面里，单击鼠标右键，选择“CodeArts Check—一键格式化”。



- 自动修复：针对插件扫描出来的单问题，插件提供建议如何修复的预览界面及自动修复。



CodeArts Check 规则配置功能介绍

打开CodeArts Check窗口，选择“Setting”页签，查看配置。

“CodeArts Check配置”界面功能说明请参见图10-1和表10-1。

图 10-1 CodeArts Check 配置界面

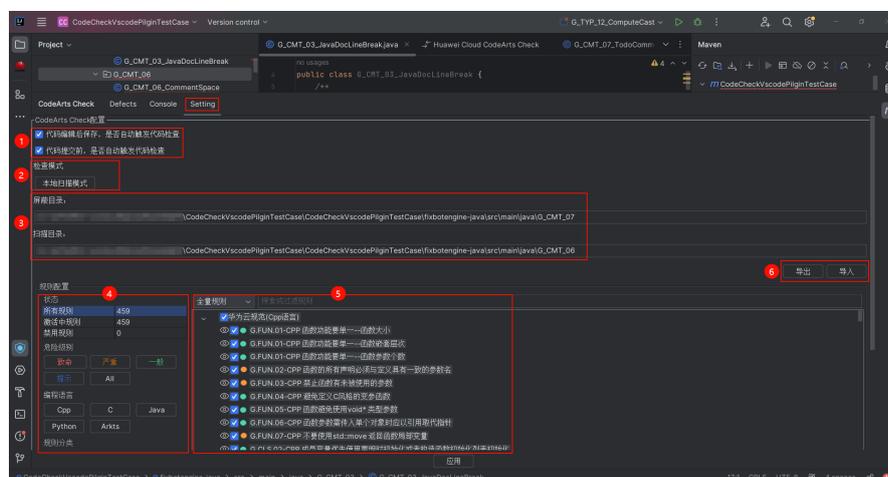


表 10-1 CodeArts Check 配置界面功能说明

序号	功能项	功能描述
1	开启/关闭无感扫描和git提交自动扫描	<ul style="list-style-type: none"> 配置无感扫描：即勾选“代码编辑后保存，是否自动触发代码检查”。扫描完成不影响用户当前作业窗口，修改保存自动扫描、结果自动刷新在CodeArts Check窗口“Defects”页签下区域；取消勾选“代码编辑后保存，是否自动触发代码检查”则关闭该功能。 配置git代码提交时自动扫描：即勾选“代码提交前，是否自动触发代码检查”；取消勾选“代码提交前，是否自动触发代码检查”则关闭该功能。
2	检查模式	无需配置，默认为本地扫描模式。

序号	功能项	功能描述
3	配置扫描排除目录	支持填写相对代码仓的相对目录，例如：一级目录写 目录名，二级目录写 目录1名/目录2名；多个指定目录扫描，可用;号隔开。 不填写目录默认扫描整个项目的所有内容（仅在项目检查时生效）。
4	分类检索	支持规则的搜索或过滤。
5	快速设置规则	支持规则的快速勾选/取消勾选。
6	导入/导出配置规则	将规则详情信息以xlsx格式的文件导出至本地。或将本地准备好的xlsx格式的规则文件导入至工具。