

代码检查

用户指南

文档版本 01
发布日期 2025-10-11



版权所有 © 华为技术有限公司 2025。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 购买并授权使用 CodeArts Check | 3 |
| 2.1 购买 CodeArts Check | 3 |
| 3 配置 CodeArts Check 项目级角色权限 | 6 |
| 4 创建代码检查任务前准备 | 9 |
| 4.1 新建任务分组 | 9 |
| 5 创建代码检查任务 | 11 |
| 6 配置代码检查任务 | 17 |
| 6.1 配置代码检查任务规则集 | 17 |
| 6.1.1 配置代码检查任务预置规则集 | 17 |
| 6.1.2 配置代码检查任务自定义规则集 | 24 |
| 6.2 配置代码检查任务模型集 | 30 |
| 6.3 配置代码检查任务质量门禁 | 36 |
| 6.4 配置代码检查任务定时执行 | 37 |
| 6.5 配置代码检查任务消息通知 | 38 |
| 6.6 配置代码检查任务检查范围 | 41 |
| 6.7 查看徽标状态 | 43 |
| 6.8 配置代码检查任务集成服务 | 44 |
| 6.9 配置代码检查任务自定义环境 | 44 |
| 6.10 配置代码检查任务高级选项 | 46 |
| 7 执行代码检查任务 | 48 |
| 8 查看代码检查结果 | 50 |
| 9 变更代码检查服务规格 | 55 |
| 10 查询审计日志 | 57 |
| 11 参考 | 59 |
| 11.1 CodeArts Check 插件使用指南（IntelliJ IDEA 版本） | 59 |

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

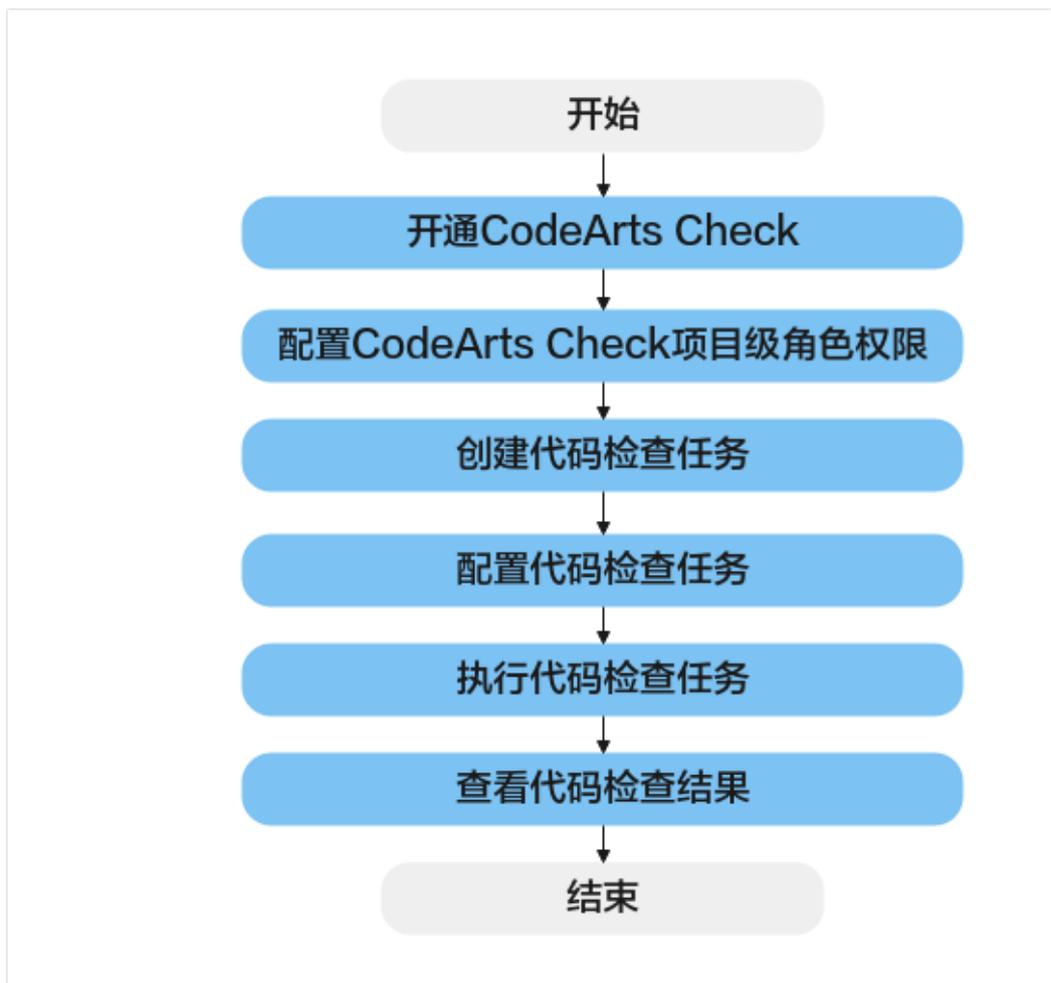


表 1-1 使用流程说明

| 流程 | 说明 |
|---|---|
| 购买 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 购买并授权使用 CodeArts Check

2.1 购买 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[购买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。

购买 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](#)。

购买建议：每5位开发人员购买1个代码检查并发。

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。 |
| 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 | 可选择“基础版”或“专业版”，差异如下。 <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/ArkTS/SQL/仓颉 (Cangjie)10+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

建议根据您的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 form for CodeArts Check. It includes sections for:

- 计费模式** (Billing Mode): 包年/包月 (Annual/Monthly)
- 区域** (Region): Multiple options like 华北-北京一, 华北-北京四, etc.
- 产品** (Product): CodeArts Check套餐
- 规格** (Specification): 基础版 (Basic) or 专业版 (Professional)
- 购买时长** (Purchase Duration): 1, 2, 3, 4, 5, 6, 7, 8, 9个月, 1年, 2年, 3年
- 协议** (Agreement): 我已阅读并同意《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 |
|---------|---------|-------|------|------|------|------|-----|-----|------|------|-------|-----------|
| 检查任务 | 创建 | √ | √ | √ | × | × | × | × | × | × | √ | √ |
| | 执行 | √ | √ | √ | × | × | × | × | × | √ | √ | √ |
| | 查看 | √ | √ | √ | × | × | × | × | × | √ | √ | √ |
| | 编辑 | √ | √ | √ | × | × | × | × | × | √ | √ | √ |
| | 删除 | √ | √ | × | × | × | × | × | × | √ | √ | √ |
| 规则集 | 设置默认规则集 | √ | √ | × | × | × | × | × | × | √ | √ | √ |
| 缺陷报告 | 查看 | √ | √ | √ | × | × | × | × | × | √ | √ | √ |
| | 编辑 | √ | √ | √ | × | × | × | × | × | √ | √ | √ |

前提条件

- 已[购买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，并参考“管理CodeArts权限”章节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：服务入口和项目入口。

- **服务入口**

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**

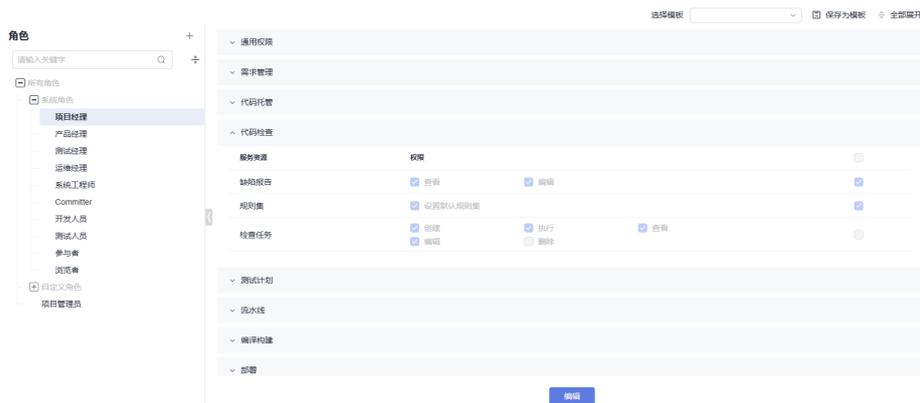
- a. 单击“立即使用”，进入代码检查服务首页。
- b. 单击导航栏“首页”。
- c. 单击需要查看的项目名称。
- d. 选择“代码 > 代码检查”，进入指定项目下代码检查任务列表页。

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 > 权限管理”。
3. 在配置角色页面，为不同的角色配置代码检查服务的各个资源权限。



4 创建代码检查任务前准备

4.1 新建任务分组

对于在同一项目中的不同模块或使用场景不同的代码检查，CodeArts Check支持分组管理。当创建一个分组后，会默认同步创建一个“未分组”分组，若创建代码检查任务时未选择分组，则创建的检查任务归档在“未分组”中。

约束与限制

- 最多支持创建100个分组。
- 创建分组支持多层创建，最多支持3层。

新建分组

1. 通过项目入口方式[访问CodeArts Check服务首页](#)。
2. 单击“新建分组”，在弹出的窗口中单击。

图 4-1 新建分组



3. 根据实际使用情况，设置分组名称，单击，新建完成。

图 4-2 设置分组名称



管理分组

1. 单击 ，打开分组管理。

图 4-3 分组管理



2. 将鼠标放在分组所在行。

单击 ，可修改分组名称。

单击 ，可调整分组顺序和删除分组。

说明

默认存在一个“未分组”分组，当分组删除后，该分组下的任务会被移入“未分组”中。

5 创建代码检查任务

本章节为用户介绍如何单个创建或批量创建不同代码源的代码检查任务。

前提条件

- 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 > 新建Repo代码仓库”，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。

约束与限制

- 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。
- 仅通过项目/租户模板的创建方式，支持批量创建代码检查任务。单次最多可以创建10个代码检查任务。
- 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。

创建代码检查任务

支持创建单个和批量代码检查任务。

创建单个 Repo 代码源检查任务

在CodeArts Repo服务中创建云端仓库时，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CodeArts Repo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 [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5-1](#)配置参数。

表 5-1 Repo 代码源检查任务参数说明

| 参数 | 说明 |
|------|--|
| 所属项目 | 任务所属项目。 <ul style="list-style-type: none">以项目入口方式访问CodeArts Check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。 |
| 新建方式 | 选择创建代码检查任务的模板。创建单个代码检查任务时，选择“空模板”。 |
| 代码源 | 选择检查的代码源。 <ul style="list-style-type: none">Repo：检查当前项目下Repo代码仓中的代码质量。在代码仓和其子模块为同一种开发语言时，支持对子模块进行检查。其他项目Repo：检查其他项目下Repo代码仓的代码质量。 |
| 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3~128。 |
| 项目 | 当“代码源”选择“其他项目Repo”时，选择Repo代码仓所在的项目。 |
| 代码仓 | 选择需要检查的代码仓库。 |
| 默认分支 | 选择需要检查的仓库分支。 |
| 检查语言 | 选择需要检查的代码语言。当前支持检查的语言可参考 支持主流开发语言 。 当代码源是Repo时，检查语言会根据代码仓默认选择，只能减少语言，不能手动新增语言。 |

步骤3 单击“确定”，完成代码检查任务创建。可在代码检查任务列表页查看到该任务。

- 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、分组和默认分支。
- 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

创建单个第三方代码仓的代码检查任务

- 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 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5-2配置参数。

表 5-2 第三方代码源检查任务参数说明

| 参数 | 说明 |
|-------|--|
| 所属项目 | 任务所属项目。 <ul style="list-style-type: none"> 以项目入口方式访问CodeArts Check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。 |
| 代码源 | 选择检查的代码来源。 <ul style="list-style-type: none"> 选择“GitHub”，检查托管在GitHub代码仓中的代码质量。创建GitHub代码源检查任务前，需将Github仓库设置成公开。 选择“GitCode”，检查托管在GitCode代码仓中的代码质量。 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。 选择“码云”，检查托管在码云代码仓中的代码质量。 |
| 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。 |
| 服务扩展点 | 选择连接到第三方代码仓库的服务扩展点。如果没有，可参考新建服务扩展点创建对应代码仓的服务扩展点。 |
| 代码仓 | 选择需要检查的代码仓库。 |
| 默认分支 | 选择需要检查的仓库分支。 |
| 检查语言 | 选择需要检查的代码语言。当前支持检查的语言可参考支持主流开发语言。 |

步骤3 单击“确定”，代码检查任务创建完成。可在代码检查任务列表页查看到该任务。

- 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、分组和默认分支。
- 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

批量创建代码检查任务

仅自定义模板的创建方式，支持批量创建代码检查任务。单次最多可以创建10个代码检查任务。

步骤1 通过项目入口方式[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5-3](#)配置参数。

表 5-3 批量任务参数说明

| 参数 | 说明 |
|------|--|
| 所属项目 | <p>需要使用的模板所属项目。</p> <ul style="list-style-type: none"> 以项目入口方式访问CodeArts Check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。 |
| 新建方式 | <p>选择使用的模板，可选择“项目模板”或“租户模板”。</p> <ul style="list-style-type: none"> 项目模板：使用当前项目下自定义的任务模板。 租户模板：使用当前租户自定义的任务模板。 <p>租户模板和项目模板的差异如下：</p> <ul style="list-style-type: none"> 通知管理：基于租户模板创建任务不继承接受者角色、自定义成员配置信息。 检查范围：基于租户模板创建任务不继承忽略文件的配置信息。 自定义环境：基于租户模板创建任务不继承私有依赖仓扩展点信息。 租户模板不支持修改模板信息。 |
| 模板 | <p>选择对应的自定义模板。若无模板，可参考自定义代码检查任务模板新建模板。</p> |
| 代码源 | <p>选择检查的代码来源。</p> <ul style="list-style-type: none"> 选择“Repo”，检查托管在Repo代码仓中的代码质量。 选择“GitHub”，检查托管在GitHub代码仓中的代码质量。创建GitHub代码源检查任务前，需将Github仓库设置成公开。 选择“GitCode”，检查托管在GitCode代码仓中的代码质量。 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。 选择“码云”，检查托管在码云代码仓中的代码质量。 |
| 任务信息 | <ul style="list-style-type: none"> 项目：选择代码检查任务的所属项目。 服务扩展点：代码源选择非Repo代码仓时需要配置。如果没有服务扩展点，可参考新建服务扩展点创建对应代码仓的服务扩展点。 代码仓：选择代码检查任务使用的代码仓。 默认分支：选择代码检查任务需要检查的仓库分支。 任务名称：填写代码检查任务的名称。支持中英文，数字，点，下划线“_”和连接符“-”，字符长度范围为3~128。 <p>单击“增加任务”，可创建新的代码检查任务，单次最多可创建10个代码检查任务。</p> |

步骤3 单击“确定”，批量代码检查任务创建完成。

可在对应项目下的代码检查任务列表中查看到新建的代码检查任务。

----结束

自定义代码检查任务模板

步骤1 访问[CodeArts Check服务首页](#)。

步骤2 在代码检查任务列表页，选择“配置中心 > 模板管理”。

步骤3 单击“新建模板”，在弹出的窗口按照配置参数。

表 5-4 模板参数说明

| 参数 | 说明 |
|-------|---|
| 所属项目 | 任务所属项目。 <ul style="list-style-type: none"> 以项目入口方式访问CodeArts Check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。 |
| 代码源 | <ul style="list-style-type: none"> 选择“GitHub”，检查托管在GitHub代码仓中的代码质量。创建GitHub代码源检查任务前，需将Github仓库设置成公开。 选择“GitCode”，检查托管在GitCode代码仓中的代码质量。 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。 选择“码云”，检查托管在码云代码仓中的代码质量。 选择“Repo”，检查托管在Repo代码仓中的代码质量。 |
| 模板名称 | 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3~128。 |
| 服务扩展点 | 选择连接到第三方代码仓库的服务扩展点。如果没有，可参考 新建服务扩展点 创建对应代码仓的服务扩展点。 |
| 代码仓 | 选择需要检查的代码仓库。 |
| 默认分支 | 选择需要检查的仓库分支。 |

步骤4 单击“确定”，代码检查任务模板创建完成。

可在代码检查任务模板列表中查看到该模板，并且在基于租户模板或者项目模板新建代码检查任务时可以选到该模板

----结束

管理代码检查任务

CodeArts Check支持批量管理代码检查任务。

步骤1 在代码检查任务列表页，勾选任务名称前的复选框。

步骤2 在弹出的窗口中可对选中的任务进行移动分组，修改配置，删除和导出任务告警的操作。

- “修改配置”支持修改[规则集](#)，[质量门禁](#)，[定时执行](#)，[MR检查模式](#)，[检查目录](#)，[自定义镜像](#)，[配置私有依赖仓扩展点](#)，[检查前置命令](#)，[新问题起始时间](#)，[源文件编码](#)，[问题责任人精准匹配](#)，[消息通知](#)，[企业微信通知](#)，[钉钉通知](#)，[飞书通知](#)。
- “导出任务告警”最多支持20个代码检查任务。
- 最多支持10个任务移动分组，修改配置和删除。

图 5-1 批量管理代码检查任务



----结束

相关操作

- 创建代码检查任务相关API接口可参考[新建检查任务](#)。
- 任务管理的相关API接口可参考[任务管理](#)。
- 使用代码检查服务的相关最佳实践可参考[代码检查最佳实践](#)。
- 后续操作可参考[配置代码检查任务](#)。

6 配置代码检查任务

6.1 配置代码检查任务规则集

6.1.1 配置代码检查任务预置规则集

规则集是一组用于代码检查的规则集合。根据不同的检查目标 and 需求，规则集有相应的分类，比如安全类、安卓应用、编程风格、编码规范等。

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。

约束与限制

- 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他语言规则集进行混合检查。
- 系统预置规则集不支持编辑，如果系统预置规则集不满足用户的需求，可基于预置规则集[创建自定义检查规则集](#)，自定义规则集可进行编辑。
- 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“编译配置”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“编译配置”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“编译配置”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](#)。

查看规则集被任务使用情况时，查看到的任务数超过10000，且该规则集使用情况变化，任务数变化有1小时的延迟。首次查看规则集的代码检查任务列表时，不存在延迟问题。

查看预置规则集

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6-1](#)。

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 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。

其中，规则使用的检查引擎默认不显示，如果需要查看，需在导航栏选择“服务 > 代码检查”，然后单击“配置中心”页签，在“高级设置”中打开显示检查引擎的开关。

注意，IAM3.0场景下需要租户权限或者用户的te_admin角色才可以配置检查引擎开关。

图 6-1 显示检查引擎



规则集列表支持按照“检查规则数量”和“修改时间”进行排序。两者之间相互排斥，例如，通过“修改时间”进行排序后，按照“检查规则数量”的排序会失效。

步骤3 单击规则集名称，进入该规则集详情页面，可查看规则集中的规则使用详情。

- **在线查看：**直接通过网页查看。

图 6-2 查看规则集的规则



- **离线查看：**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

复制预置规则集

在规则集列表页，单击规则集所在行的 按钮，可复制当前规则集，复制后的规则集为自定义规则集，关于自定义规则集的约束与限制可参考[约束与限制](#)。。

查看使用规则集的任务列表

1. 在规则集列表页，单击规则集名称，进入规则集详情页。
2. 单击“*** > 被任务使用的情况”，可以查看使用该规则集的代码检查任务列表。

说明

当查看到的规则集被使用的任务数超过10000，且该规则集使用情况发生变化时，任务数变化存在1小时的延迟。首次查看规则集的代码检查任务列表时，不存在延迟问题。

修改默认规则集

初次使用代码检查服务时，每种编程语言都预置了一个默认的检查规则集，如表6-1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时，单击“编译配置”，配置代码检查扩展参数并开启配置开关，完成后单击“确定”。

配置编译工具时，勾选编译工具后会自动填充默认的工具版本和命令。如需自定义，可手动修改。其中使用GO编译工具时，为保证构建成功，需正常拉取依赖包，如需修改GOPROXY环境变量，请在编译脚本中使用go env -w方式。

图 6-3 以 Java 为例的编译配置参数设置

编译配置 ×

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。

编译工具选项

已关闭

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

确定 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“编译配置”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“编译配置”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“编译配置”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](#)。

步骤5 单击页面右上角“管理规则集”，可跳转至规则集列表页。

----结束

预置规则集清单

表 6-1 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 |
|-----|--|----------------|---|---|
| C++ | <ul style="list-style-type: none"> ● 支付卡行业数据安全标准 (PCI DSS) 3.2.1 ● 支付卡行业数据安全标准 (PCI DSS) 4.0 ● MISRA C 2012规则集 ● AUTOSAR C++14 规则集 ● 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 ● 鸿蒙C/C++语言推荐内容检查规则集 ● 通用检查规则集 | 华为C/C++编码规范规则集 | .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 |

| 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 |
|------|--|-------------|----------|---|
| Java | <ul style="list-style-type: none"> ● 支付卡行业数据安全标准 (PCI DSS) 4.0 ● 支付卡行业数据安全标准 (PCI DSS) 3.2.1 ● P3C规范JAVA推荐规则集 ● 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 ● CodeArts Check JAVA推荐规则集 (旧) ● 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 ● 鸿蒙JAVA语言推荐内容检查规则集 | 华为Java编程规则集 | .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 |

| 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 |
|------------|--|-----------|----------|---|
| 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 | 华为编码规范规则集 | .go | <ul style="list-style-type: none"> go-1.10.3 go-1.11.6 go-1.12.1 |
| 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 鸿蒙PYTHON语言推荐内容检查规则集 | 华为编码规范规则集 | .py | / |
| JavaScript | <ul style="list-style-type: none"> 华为JS编程规则集 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 鸿蒙JAVASCRIPT语言推荐内容检查规则集 | 华为JS编程规则集 | .js/.jsx | / |
| 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"> dotnetcore2.2 dotnetcore3.0 dotnetcore3.1 dotnetcore5.0 dotnetcore6.0 msbuild15-dotnetframework3.5 msbuild15-dotnetframework4.7.2 msbuild16-dotnetframework4.8 |

| 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 |
|-------------|---|-----------|----------|---------------|
| Type Script | <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 | / |
| CSS | <ul style="list-style-type: none"> 通用检查规则集 | 通用检查规则集 | .css | / |
| HTML | <ul style="list-style-type: none"> 通用检查规则集 | 通用检查规则集 | .html | / |
| 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 | / |
| LUA | 通用检查规则集 | 通用检查规则集 | .lua | / |
| RUST | 通用检查规则集 | 通用检查规则集 | .rs | / |
| Shell | 通用检查规则集 | 通用检查规则集 | .sh | / |
| KOTLIN | 通用检查规则集 | 通用检查规则集 | .kt | / |
| SCALA | 通用检查规则集 | 通用检查规则集 | .scala | / |
| ArKTS | 通用检查规则集 | 通用检查规则集 | .ets | / |
| SQL | 通用检查规则集 | 通用检查规则集 | .sql | / |
| CANJIE | 华为仓颉编程规则集 | 华为仓颉编程规则集 | .cj | / |

相关操作

- 关于规则集的相关API接口可参考[规则管理](#)、[查询任务的已选规则集列表v2](#)、[查询任务的已选规则集列表v3](#)、[查询任务规则集的检查参数v2](#)、[查询任务规则集的检查参数v3](#)和[修改任务规则集](#)。
- 关于预置规则集的使用最佳实践可参考[使用预置规则检查通用Git代码仓中的代码质量](#)和[使用预置规则检查GitCode代码仓中的代码质量](#)。

6.1.2 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外，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，即一个规则集只能配置同一种语言类型的检查规则。

约束与限制

关于自定义规则集的限制：

- 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。
- 自定义规则集中“检查语言”不支持修改。
- 自定义规则集中，规则数量上限为3000。
- te_admin账号、租户空间管理员、租户空间所有者和规则集创建者可以修改自定义规则集。
- te_admin账号、租户空间管理员、租户空间所有者和规则集创建者可以删除自定义规则集。
- 预置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自定义规则集，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他规则集。
- 同项目下账号之间进行资源移交之后，被移交的账号具有移交账号创建的规则集的管理权限。

关于自定义规则的限制：

- 支持的自定义规则数量可参考[规格与限制](#)。
- 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。
- te_admin账号、租户空间管理员、租户空间所有者和规则创建者可以编辑和删除自定义规则。
- 同项目下账号之间进行资源移交之后，被移交的账号具有移交账号创建的规则的管理权限。

关于批量导入规则的限制：

- 批量导入规则为受限功能，如需使用，需联系技术支持。
- 导入文件大小不大于1MB。
- 每次上传的规则条数不大于100。

创建自定义检查规则集

步骤1 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、检查语言和新建方式。

其中新建方式有三种：

- 直接新建：新建一个空的规则集，所有规则需逐一勾选。
- 复制：选择“复制”，在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。
- 继承：选择“继承”，在下拉框中选择已有的规则集，继承其中的规则。单击 **+** 可以添加规则集，最多支持继承5个规则集。

如果继承的多个规则集间有冲突，则以优先级高的为准。数值越小，优先级越高。比如，添加的第二个规则集中的规则和第一个规则集中的冲突，那么以第一个规则集的规则为准。

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、适用范围（版本级、门禁级）和延迟时间，单击右上角“保存”。

在“选项”列中，部分规则可设置参数值，不同配置项可设置的参数类型不同，主要包括INTEGER，FLOAT，STRING和BOOLEAN，用户只能填写对应类型的值，图6-5以INTEGER型为例，表示参数值仅能输入INTEGER型。

图 6-4 可设置参数值的配置项



图 6-5 INTEGER 型参数值



说明

- 门禁级：MR扫描和CR扫描属于门禁级，其适用场景如下。
 - secbrella和SQLGuardian引擎规则不支持门禁级检查，规则集配置界面无门禁级选项。
 - sonarqube java的编译规则，若配置了编译命令，且选择了增量MR检查模式，可能会因为编译失败而导致任务执行失败。可以在规则列表中查看是否为编译型的规则。
- 版本级：除MR扫描和CR扫描外均属于版本级，所有规则都支持版本级检查。
- 延迟时间：自定义规则集支持设置规则的延迟告警开始时间和延迟告警结束时间。规则集延迟时间设置后重新扫描任务，当告警发现时间在设置的延迟时间范围内时，告警会被标识为延迟上线问题。延迟上线问题不会被统计为正式问题，因此不会影响质量门禁。

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。

----结束

查看自定义规则集

步骤1 在“我创建的规则集”页签下，单击规则集名称，进入规则集详情页面可规则集中使用的规则详情。

- **在线查看**

其中，规则使用的检查引擎默认不显示，如果需要查看，需在导航栏选择“服务 > 代码检查”，然后单击“配置中心”页签，在“高级设置”中打开显示检查引擎的开关。

注意，IAM3.0场景下需要租户权限或者用户的te_admin角色才可以配置检查引擎开关。

图 6-6 显示检查引擎



规则集列表支持按照“检查规则数量”和“修改时间”进行排序。两者之间相互排斥，例如，通过“修改时间”进行排序后，按照“检查规则数量”的排序会失效。

- **离线查看**

单击“导出”，可导出该规则集中已启用的规则到本地查看规则详情。

步骤2 在规则集详情页面，单击“修改基本信息”，可修改规则集名称、描述信息和规则集所有者。

说明

- 自定义规则集中“检查语言”不支持修改。
- 系统预置规则集无法修改。
- te_admin账号、租户空间管理员、租户空间所有者和规则集创建者可以修改自定义规则集。

步骤3 在规则集详情页面，单击 **...**，选择“被任务使用的情况”，可以查看使用该规则集的代码检查任务列表（包括模板任务）。

说明

当查看到的任务数超过10000，且该规则集使用情况变化时，任务数变化有1小时的延迟。首次查看规则集的代码检查任务列表时，不存在延迟问题。

----结束

删除自定义规则集

- **服务下删除：**

- a. 通过服务入口方式[访问CodeArts Check服务首页](#)。
- b. 依次单击“规则集 > 我创建的规则集”。

- c. 单击规则集所在行的 ，即可删除自定义检查规则集
- **项目下删除：**
在自定义规则集列表页，单击规则集所在行的 ，选择“删除”即可删除自定义检查规则集。
 - te_admin账号、租户空间管理员、租户空间所有者和规则集创建者可以删除自定义规则集。
 - 预置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自定义规则集，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他规则集。

配置代码检查服务使用自定义规则集

步骤1 在代码检查任务详情页选择“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

- 支持的自定义规则数量可参考[规格与限制](#)。
- 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。
- te_admin账号、租户空间管理员、租户空间所有者和规则创建者可以编辑和删除自定义规则。

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 > 新建自定义规则”，参考[表6-2](#)配置参数。

表 6-2 自定义规则参数配置说明

| 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1~255 | 是 |
| 工具规则名称 | 根据上传的规则源码文件名称自动填充，不可手动修改。 | 是 |

| 参数 | 说明 | 是否必填 |
|------|---|------|
| 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和ArkTS语言。 | 是 |
| 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

批量导入自定义规则

批量导入规则为受限功能，如需使用，需联系技术支持。

步骤1 单击“规则”页签。

步骤2 依次单击“新建规则 > 批量导入规则”，在弹出的窗口中，下载批量导入规则模板。

图 6-7 批量导入规则



步骤3 按照模板要求填写完规则信息后，单击 ，选择本地填写完成的批量导入规则文件。

步骤4 单击“确定”，完成导入。

----结束

相关操作

- 关于规则集的相关API接口可参考[规则管理](#)、[查询任务的已选规则集列表v2](#)、[查询任务的已选规则集列表v3](#)、[查询任务规则集的检查参数v2](#)、[查询任务规则集的检查参数v3](#)和[修改任务规则集](#)。
- 关于自定义规则以及规则集的最佳实践可参考[使用自定义规则检查CodeArts Repo代码仓中的代码质量](#)。

6.2 配置代码检查任务模型集

“模型”可以理解为是对特定编程行为或系统交互的抽象表示，它定义了分析器应该如何理解和处理特定的代码模式或API调用。

与规则不同的是，规则通常是描述“什么是对的，什么是错的”的检查条款，而模型则更关注“代码在特定上下文中的行为特征”。

模型为分析器提供了关于代码行为的知识库，使其能够更准确地进行推理。模型主要是对函数的行为或者变量进行建模，例如malloc/free系列的分配释放行为模型。

约束与限制

- 模型集为受限功能，仅[购买代码检查安全增强包](#)后才可使用该功能。
- 当前仅支持“华北-北京四”。
- 每个租户最多可创建100个模型集和20条模型。
- 模型集当前仅支持C/C++语言和secbrella引擎。
- 新建模型时，上传的文件仅支持json后缀的文件。

查看模型集

用户可提前查看模型集详情，根据实际需要选择对应的模型集。

- 步骤1** 访问[CodeArts Check服务首页](#)。
- 步骤2** 单击“模型集”页签，进入模型集列表页面。
- 步骤3** 单击模型集名称，进入该模型集详情页面，可查看模型集中的模型使用详情。
- 结束

配置代码检查任务使用模型集

- 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 > 模型集”。
- 步骤3** 在右侧区域选择使用的模型集。
- 步骤4** 单击页面右上角“管理模型集”，可跳转至模型集列表页。
- 结束

新建模型集

- 步骤1** 访问[CodeArts Check服务首页](#)。
- 步骤2** 单击“模型集”页签，进入模型集列表页面。
- 步骤3** 单击“新建模型集”，参考[表6-3](#)配置模型集参数。然后单击“确定”，进入选择模型界面。

表 6-3 新建模型集

| 参数 | 说明 |
|-------|---|
| 模型集名称 | 创建的模型集名称，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 |
| 检查语言 | 模型集支持的检查语言，当前仅支持C/C++语言。 |
| 描述 | 对当前模型集的描述，字符长度范围为0~256。 |

- 步骤4** 勾选使用的模型后，单击“保存”即可。若无模型可选，可参考[新建模型](#)添加。
- 结束

修改模型集

- 在模型集列表页，单击模型集所在行 ，进入模型集详情页。
- 可增加或者删除模型。
- 单击“修改基本信息”，可修改模型集名称和描述。

删除模型集

在模型集列表页，单击模型集所在行 ，可删除模型集。

新建模型

步骤1 单击新建的模型集名称，进入模型集详情页面，单击“新建模型”。

步骤2 在新建模型页面，参考表6-4配置模型参数。

表 6-4 新建模型

| 参数 | 说明 |
|------|---|
| 模型文件 | 在本地创建的模型文件，操作可参考 创建模型文件 。 |
| 模型名称 | 上传模型文件后自动生成，无需手动填写。 |
| 检查语言 | 当前仅支持C/C++语言。 |
| 执行工具 | 当前仅支持secbrella引擎。 |
| 标签 | 对当前模型的标签，用于识别当前模型的分类，字符长度范围为0~256。 |
| 描述 | 对当前模型描述，字符长度范围为0~256。 |

步骤3 单击“确定”。

步骤4 新建完成后，在模型集下可以对模型进行如下操作。

- 单击模型所在行 ，可在弹框中重新上传模型文件和修改用户标签以及描述。
- 单击模型所在行 ，可下载模型文件。
- 单击模型所在行 ，可删除模型。

----结束

创建模型文件

代码示例如下：

```
{
  "RulePackage": {
    "Language": "C++" # 必填，C或者C++
  },
  "Rules": {
    "TaintSourceRule": [ # 污点源配置
      {
        "Function": { # 污点源函数配置1
          "Namespace": {
            "Value": "TestSpace" # 命令空间，如果没有可以不写
          },
          "Class": {
            "Value": "TestClass" # 类名，如果没有可以不写
          },
          "Function": {
            "Value": "TaintCase1" # 函数名，必填
          }
        }
      }
    ],
    "Source": { # 污点源函数参数配置
      "InArgs": "0,2-3", # InArgs配置函数形参位置（参数具体配置看基本概念）
      "TaintFlags": "UntrustedSource" # 污点标识，固定值
    }
  }
}
```

```
},
{
  "Function": { # 污点源函数配置2
    "Namespace": {
      "Value": "TestSpace" # 命令空间, 如果没有可以不写
    },
    "Class": {
      "Value": "TestClass" # 类名, 如果没有可以不写
    },
    "Function": {
      "Value": "TaintCase2" # 函数名, 必填
    }
  },
  "Source": { # 污点源函数参数配置
    "OutArgs": "return,2-3", # OutArgs在配置函数实参位置或者配置函数返回值
    "TaintFlags": "UntrustedSource" # 污点标识, 固定值
  }
},
{
  "GlobalVariable": { # 污点源全局变量配置
    "Name": {
      "Value": "g_xxxx" # 全局变量名, 必填
    }
  },
  "Source": {
    "TaintFlags": "UntrustedSource" # 污点标识, 固定值
  }
},
{
  "TaintVariable": { # 污点源局部变量配置
    "Name": {
      "Value": "local_var" # 局部变量名, 必填
    }
  },
  "Source": {
    "TaintFlags": "UntrustedSource" # 污点标识, 固定值
  }
}
],
"TaintSinkRule": [ # 污点爆发配置
{
  "Function": { # 污点爆发函数配置
    "Namespace": {
      "Value": "TestSpace" # 命令空间, 如果没有可以不写
    },
    "Class": {
      "Value": "TestClass" # 类名, 如果没有可以不写
    },
    "Function": {
      "Value": "TaintSink" # 函数名, 必填
    }
  },
  "Sink": {
    "InArgs": "0", # InArgs配置函数爆发参数位置, TaintSinkRule配置只有InArgs
    "TaintFlags": "UntrustedSink" # 污点标识, 固定值
  }
}
],
"TaintPassthroughRule": [ # 污点传播配置
{
  "Function": { # 污点传播函数配置1
    "Namespace": {
      "Value": "TestSpace" # 命令空间, 如果没有可以不写
    },
    "Class": {
      "Value": "TestClass" # 类名, 如果没有可以不写
    },
    "Function": {
      "Value": "TaintPass" # 函数名, 必填
    }
  }
}
]
```

```

    }
  },
  "Passthrough": {
    "InArgs": "0", # 配置位置在InArgs的污点参数传播到位置是OutArgs的参数或者返回值
    "OutArgs": "1"
  }
},
{
  "Function": { # 污点传播函数配置2
    "Function": {
      "Value": "TaintPass2" # 函数名, 必填
    }
  },
  "Passthrough": {
    "InArgs": "0", # 配置位置在InArgs的污点参数传播到位置是OutArgs的参数或者返回值
    "OutArgs": "return"
  }
}
],
"TaintCleanseRule": [ # 污点净化配置
{
  "Function": { # 污点净化函数配置
    "Namespace": {
      "Value": "TestSpace" # 命令空间, 如果没有可以不写
    },
    "Class": {
      "Value": "TestClass" # 类名, 如果没有可以不写
    },
    "Function": {
      "Value": "Sanitize" # 函数名, 必填
    }
  },
  "Cleanse": {
    "InArgs": "0" # InArgs配置函数净化参数位置, TaintCleanseRule配置只有InArgs
  }
}
],
"TaintAnnotations": { # 内存申请释放, 空指针类配置
  "SpaceA::ClassA::test_malloc": [ # 函数名, 必填, 格式: 命令空间:(没有就不写)类名:(没有就不写)函数名
    ["AllocSource::1", "InitNull"], # 第一个[], 代表返回值, 没有配置就为空[], 这里配置了"AllocSource::1"代表返回值有返回内存, 且1代表id, 该内存只能用相同id的释放函数释放;还配置了"InitNull"代表可能返回空指针
    [], # 第二个[], 如果是类函数, 代表this指针, 没有配置就为空[]
    [] # 第三个[], 如果是类函数, 代表第一个参数
  ],
  "malloc": [ # 函数名
    ["AllocSource::1", "InitNull"], # 第一个[], 代表返回值, 没有配置就为空[], 这里配置了"AllocSource::1"代表返回值有返回内存, 且1代表id, 该内存只能用相同id的释放函数释放;还配置了"InitNull"代表可能返回空指针
    [ # 第二个[], 如果是普通函数, 代表第一个参数, 没有配置就为空[], 这里配置了"AllocSize"代表该参数是申请内存的长度
      "AllocSize"
    ]
  ],
  "free": [ # 函数名
    [], # 第一个[], 代表返回值, 没有配置就为空[]
    [ # 第二个[], 如果是普通函数, 代表第一个参数, 没有配置就为空[], 这里配置了"FreeSink::1"代表该参数有释放行为, 且1代表id, 只有用于释放来自相同id的申请内存函数的内存, 如果希望该释放函数不需要用于指定的申请函数, 那么可以配置"FreeSink"
      "FreeSink::1"
    ]
  ]
}
}
}
}

```

表 6-5 模型影响的规则

| 模型类型 | 模型生效的规则 |
|----------------------------|---|
| 申请函数 TaintAnnotations | G.VAR.08: 资源不再使用时应予以关闭或释放,内存不再使用时应予以释放。 |
| 内存释放函数 TaintAnnotations | <ul style="list-style-type: none"> G.MEM.05: 不访问已释放的内存,不使用已释放的内存或重复释放内存。 G.VAR.08: 资源不再使用时应予以关闭或释放,内存不再使用时应予以释放。 |
| 返回空指针 TaintAnnotations | <ul style="list-style-type: none"> G.FUD.08: 将字符串或指针作为函数参数时,在函数体中应检查参数是否为NULL。 G.STD.03-CPP: 禁止从空指针创建std::string,解引用空指针。 |
| 污点源函数 TaintSourceRule | <p>G.STD.08-CPP: 外部数据用于容器索引或迭代器时必须确保在有效范围内。</p> <p>G.STD.05-CPP: 确保用于字符串操作的缓冲区有足够的空间容纳字符数据和结束符,并且字符串以null结束符结束,缓冲区越界访问。</p> <p>G.STR.01: 确保字符串存储有足够的空间容纳字符数据和null结束符。</p> <p>G.INT.06: 校验外部数据中整数值的合法性。</p> <p>G.MEM.03: 外部输入作为内存操作相关函数的复制长度时,需要校验其合法性。</p> <p>G.FUU.18: 禁止直接使用外部数据拼接SQL命令、不可信的输入可能引起命令注入、格式化注入、SQL注入等注入类问题。</p> <p>G.FUU.16: 禁止外部可控数据作为进程启动函数的参数。</p> |
| 污点净化函数 TaintCleanseRule | <p>G.STD.08-CPP: 外部数据用于容器索引或迭代器时必须确保在有效范围内。</p> <p>G.STD.05-CPP: 确保用于字符串操作的缓冲区有足够的空间容纳字符数据和结束符,并且字符串以null结束符结束,缓冲区越界访问。</p> <p>G.STR.01: 确保字符串存储有足够的空间容纳字符数据和null结束符。</p> <p>G.INT.06: 校验外部数据中整数值的合法性。</p> <p>G.MEM.0: 外部输入作为内存操作相关函数的复制长度时,需要校验其合法性。</p> <p>G.FUU.18: 禁止直接使用外部数据拼接SQL命令、不可信的输入可能引起命令注入、格式化注入、SQL注入等注入类问题。</p> <p>G.FUU.1: 禁止外部可控数据作为进程启动函数的参数。</p> |

| 模型类型 | 模型生效的规则 |
|--------------------------------|--|
| 污点传播函数 TaintPassthroughRule | <p>G.STD.08-CPP: 外部数据用于容器索引或迭代器时必须确保在有效范围内。</p> <p>G.STD.05-CPP: 确保用于字符串操作的缓冲区有足够的空间容纳字符数据和结束符, 并且字符串以null结束符结束, 缓冲区越界访问。</p> <p>G.STR.01: 确保字符串存储有足够的空间容纳字符数据和null结束符。</p> <p>G.INT.06: 校验外部数据中整数值的合法性。</p> <p>G.MEM.03: 外部输入作为内存操作相关函数的复制长度时, 需要校验其合法性。</p> <p>G.FUU.18: 禁止直接使用外部数据拼接SQL命令、不可信的输入可能引起命令注入、格式化注入、SQL注入等注入类问题。</p> <p>G.FUU.16: 禁止外部可控数据作为进程启动函数的参数。</p> |
| 污点爆发函数 TaintSinkRule | <p>G.FUU.18: 禁止直接使用外部数据拼接SQL命令、不可信的输入可能引起命令注入、格式化注入、SQL注入等注入类问题。</p> <p>G.FUU.16: 禁止外部可控数据作为进程启动函数的参数。</p> |

6.3 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门禁阈值, 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](#)。
- te_admin账号、租户空间管理员和租户空间所有者, 具有更新租户级门禁配置信息的权限。

本节以任务级门禁设置为例介绍操作步骤。

步骤1 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 6-8 配置质量门禁

质量门禁 

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

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

6.4 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。

约束与限制

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 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 6-9 配置执行计划

定时执行

星期日 星期一 星期二 星期三 星期四 星期五 星期六

步骤5 完成后，单击“保存”。

----结束

6.5 配置代码检查任务消息通知

CodeArts Check的通知设置支持租户级、系统项目级、服务项目级和任务级，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。入口分别如下：

- 服务项目级：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。

配置消息通知

代码检查服务支持发送服务动态/邮件通知、企业微信消息通知，钉钉消息通知和飞书消息通知。

配置代码检查任务消息通知

步骤1 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 选择“钉钉”，参考[表6-6](#)配置参数。

表 6-6 配置钉钉

| 配置项 | 是否必填 | 说明 |
|-------------|------|--|
| Webhook URL | 是 | 输入钉钉机器人Webhook地址，长度不超过256个字符。 获取钉钉机器人Webhook地址的方式，请参考 钉钉开放平台官网 。 |
| 密钥 | 否 | 如果钉钉机器人的安全设置中开启了加签，需输入加签密钥，长度不超过256个字符。 获取钉钉机器人加签密钥的方式，请参考 钉钉开放平台官网 。 |

| 配置项 | 是否必填 | 说明 |
|----------|------|--|
| 事件类型 | 是 | 勾选事件类型。 <ul style="list-style-type: none"> 全部：所有事件均发送通知。 门禁通过：仅门禁通过的事件发送通知。 门禁不通过：仅门禁未通过的事件发送通知。 |
| 通知内容 | 是 | 勾选需要通知内容，包括检查结果，门禁结果，问题数，跳转链接，执行人和完成时间。 |
| @user_id | 否 | 输入被@成员的钉钉user_id，多个user_id之间使用“,”隔开，长度不超过1024个字符。 |

步骤2 配置完成后，单击“保存”。

步骤3 将“钉钉”开关设置为  状态。

当开关状态为“开启”且任务运行结果满足事件类型设置集时，代码检查服务会发送消息到指定的钉钉群并@到指定用户。

----结束

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. 选择“企业微信”，参考表6-7配置参数。

表 6-7 配置企业微信

| 配置项 | 是否必填 | 说明 |
|-------------|------|----------------------------------|
| Webhook URL | 是 | 填写步骤1中获取到的Webhook地址。长度不超过256个字符。 |

| 配置项 | 是否必填 | 说明 |
|----------|------|--|
| 事件类型 | 是 | 勾选事件类型。 - 全部：所有事件均发送通知。 - 门禁通过：仅门禁通过的事件发送通知。 - 门禁不通过：仅门禁未通过的事件发送通知。 |
| 通知内容 | 是 | 勾选需要通知内容，包括检查结果，门禁结果，问题数，跳转链接，执行人和完成时间。 |
| @user_id | 否 | 输入被@成员的企业微信user_id，多个user_id之间使用“，”隔开，长度不超过1024个字符。 |

2. 配置完成后，单击“保存”。

3. 将“企业微信”开关设置为  状态。

当开关状态为“开启”且任务运行结果满足事件类型设置集时，代码检查服务会发送消息到指定的企业微信群并@到指定企业微信名。

----结束

配置飞书通知

1. 选择“飞书”，参考表6-8配置参数。

表 6-8 配置飞书

| 配置项 | 是否必填 | 说明 |
|-------------|------|--|
| Webhook URL | 是 | 输入飞书机器人Webhook地址，长度不超过256个字符。 获取飞书机器人Webhook地址的方式，请参考 飞书开放平台官网 。 |
| 密钥 | 否 | 如果飞书机器人的安全设置中开启了加签，需输入加签密钥，长度不超过256个字符。 获取飞书机器人加签密钥的方式，请参考 飞书开放平台官网 。 |
| 事件类型 | 是 | 勾选事件类型。 • 全部：所有事件均发送通知。 • 门禁通过：仅门禁通过的事件发送通知。 • 门禁不通过：仅门禁未通过的事件发送通知。 |
| 通知内容 | 否 | 勾选需要通知内容，包括检查结果，门禁结果，问题数，跳转链接，执行人和完成时间。 |
| @user_id | 否 | 输入被@成员的飞书user_id，多个user_id之间使用“，”隔开，长度不超过1024个字符。 |

2. 配置完成后，单击“保存”。

3. 开关设置为  状态。

当开关状态为“开启”且任务运行结果满足事件类型设置集时，代码检查服务会发送消息到指定的飞书用户。

6.6 配置代码检查任务检查范围

用户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。

约束与限制

- secbrella引擎均不支持MR合入。
- C#语言仅支持MR全量扫描模式。

设置 MR 检查模式

步骤1 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 > 检查范围”。

步骤4 在“MR检查模式”区域选择扫描模式。默认为“扫描变更文件”。

- 扫描变更文件：对MR中产生变更的文件进行检查，统计这些文件中的所有代码问题，含未修改部分的代码问题。
- 全量扫描：对源分支的全部文件进行检查，统计源分支的所有代码问题。
- 扫描变更代码：分别扫描源分支及目标分支的全部文件，统计源分支相对目标分支新增的代码问题。

MR检查模式

扫描变更文件 全量扫描 扫描变更代码

保存

步骤5 选择完成后，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 ?

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。

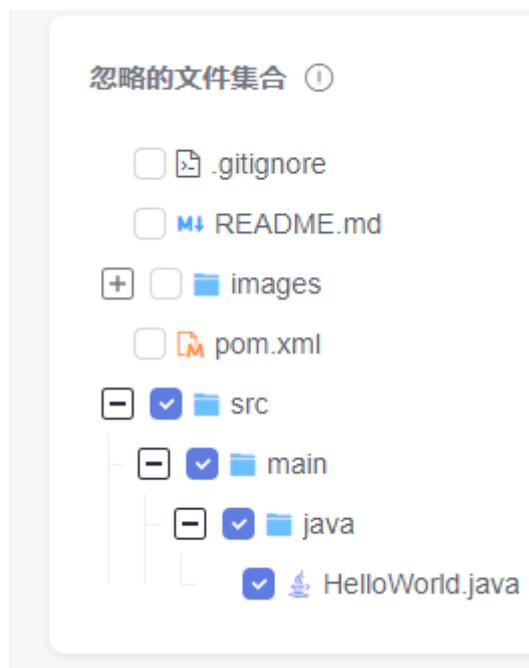
设置忽略文件模式

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。

忽略的文件集合：通过勾选文件的形式设置不检查的文件。

- 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。



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。

正则表达式：使用正则表达式排查不需要检查的文件，配置后单击“正则校验”，在弹出的窗口中，对填写的表达式进行测试匹配。

表达式要求如下：

- 该值是一个使用括号括起来的正则表达式，正则表达式元字符需要转义。
- 过滤目录或者文件，只需要写出过滤的特征字符串即可，只要满足特征字符串就可以达到过滤的目的。
举例：
 - 过滤包含test、llt字符的目录或文件：(test|llt);过滤一段路径：(apps/syn/test/)
 - 只检查包含src、config的目录或文件：^(?!(src|config)).*\$)
 - 只检查包含src、config的目录或文件，但排除掉包含test、llt字符的目录或文件：((^(?!(src|config)).*\$)|test|llt)
- 不使用“|”结尾，不允许使用空格。否则，会把所有文件全部排除。

配置完成后需单击“保存”后生效。

6.7 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=SUM Σ(告警数量*告警级别影响度)/代码量，各个健康度对应的标识如下：

A: 健康度≤0.001

B: 0.001 < 健康度 ≤0.005

C: 0.005 < 健康度 ≤0.01

D: 健康度 > 0.01

查看徽标状态

步骤1 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

6.8 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。

配置合并请求时触发

- 步骤1** 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“合并请求时触发”，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

配置代码提交时触发

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

仅对已创建任务的分支有效。
- 步骤2** 单击“保存”。

----结束

6.9 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** 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

同时支持用户通过配置私有依赖库作为私有依赖下载源。当CodeArts Check需要引用第三方服务的私有依赖仓时，需通过服务扩展点连接，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[新建nexus repository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。
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

配置执行主机

配置代码检查任务运行的主机。可根据实际情况选择资源池，自定义资源池和专属资源池为在资源池中添加的代理执行机，具体自定义操作可参考[资源池管理](#)。

参考[表6-9](#)选择资源池后，单击“保存”即可。

表 6-9 执行主机

| 主机类型 | 说明 |
|--------|--|
| 默认资源池 | 代码检查服务自身提供的执行主机，用户无需配置即可开箱即用。执行机默认规格为8U32G。 |
| 专属资源池 | 由CodeArts自动创建并托管的用户专享的任务运行环境，支持内网连通用户私有VPC，提供安全的全内网运行环境。 在“代理资源池”的下拉框中选择对应的资源池。 |
| 自定义资源池 | 用户自行提供的计算资源，通过注册的方式托管到代码检查服务中，通过代码检查服务进行调度并执行代码检查任务。代码检查服务目前仅支持LINUX和LINUX_DOCKER自定义执行机。 在“代理资源池”的下拉框中选择对应的资源池。 |

配置第三方引擎

如果CodeArts Check当前提供的检查引擎不满足用户的使用需求，可使用第三方引擎检查工具执行代码检查任务。

第三方引擎仅支持在自定义执行机上使用，在配置第三方引擎前，需在自定义执行机上导入第三方引擎和相关插件。

- 引擎名称：选择导入的第三方引擎。
- 引擎执行命令：输入对应开发语言的引擎命令，字符范围为0~1024。
- 报告路径：填写执行报告在自定义执行机上的存放路径。

相关操作

- 关于自定义执行机的最佳实践可参考[使用自定义执行机执行代码检查任务](#)。
- 关于第三方引擎的最佳实践可参考[基于第三方引擎执行代码检查](#)。

6.10 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 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。



The screenshot shows a configuration panel with two sections. The top section is titled '新问题起始时间' (New Issue Start Time) and includes a text input field containing '2023/02/16' and a calendar icon. Below this is a section titled '源文件编码' (Source File Encoding) with a text input field containing 'UTF-8' and a dropdown arrow. A red box highlights the '源文件编码' section.

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

7 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 [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

检查代码仓其他分支的代码

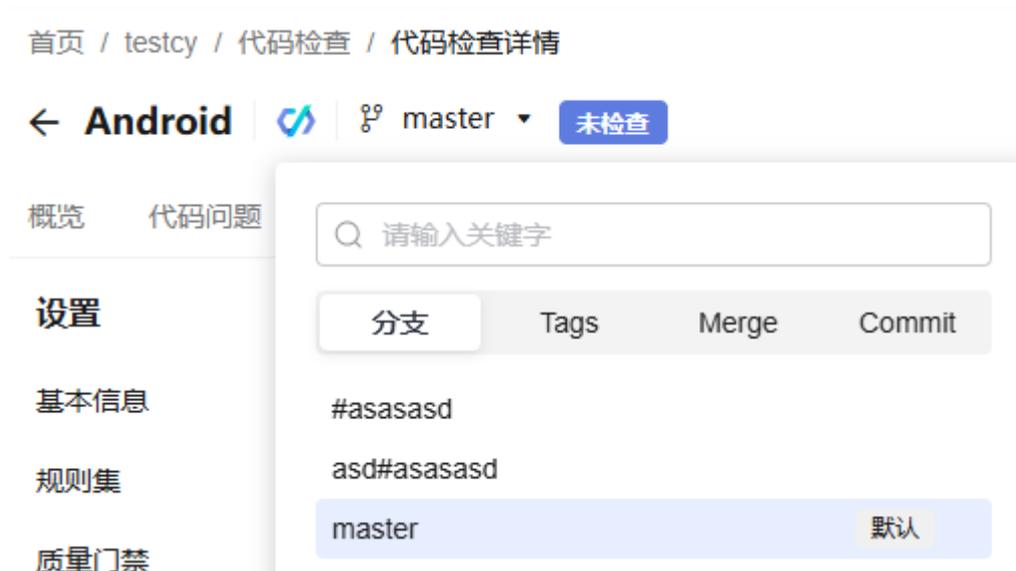
代码检查服务支持按分支、仓库标签、MR请求和CommitID筛选代码仓，执行代码检查任务。

在代码检查详情页面可以手动选择代码仓。具体操作指导如下：

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  可以通过分支、仓库标签、MR请求和CommitID筛选代码仓进行代码检查。

图 7-1 筛选代码仓



步骤3 设置筛选条件后，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

相关操作

- 执行代码检查任务时常见问题请参考[代码检查常见问题](#)。
- 执行检查任务相关API接口可参考[执行检查任务](#)。
- 后续操作可参考[查看代码检查结果](#)。

8 查看代码检查结果

为了确保代码质量，及时发现并修复潜在的问题，代码检查任务执行完成后，需及时查看代码检查结果。

前提条件

您已[创建并执行代码检查任务](#)。

查看代码检查结果

代码检查结果包括检查详情、代码问题、代码度量和检查日志。

查看代码检查详情

步骤1 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。
- 未闭环问题级别：未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。**

批量任务导出：

- 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。
- 在页面下方弹框中单击“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。

单个任务导出：

- 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作，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。 |

| 过滤条件 | 说明 |
|------|--|
| 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。该功能支持Scrum类型项目和IPD类型项目。
- 单击“查看问题单”，查看问题单详情。该功能支持Scrum类型项目和IPD类型项目。
- 单击“删除问题单”，在弹框中单击“确定”，可删除当前问题单。
- 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。代码度量包括以下两种类型的问题：

- 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 }
```

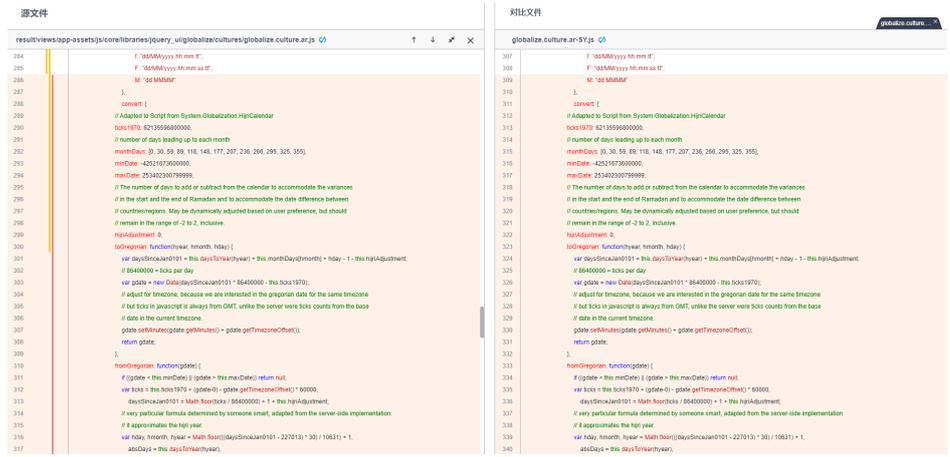
- 重复率：代码检查服务通过重复块、重复行、重复率来识别被测代码的重复情况。

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         Lines: 92 - 165
343
344         hmonth++;
345     }
346     hmonth--;
347     hday = daysInYear - this.monthDays(hmonth);
348     return [hyear, hmonth, hday];
349 }
350 },
351 daysToYear: function(year) {
352     // calculates how many days since Jan 1, 0001
353     var yearsToYear30 = Math.floor((year - 1) / 30) * 30,
354         yearsInto30 = year - yearsToYear30 - 1,
355         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 8-1 步骤日志

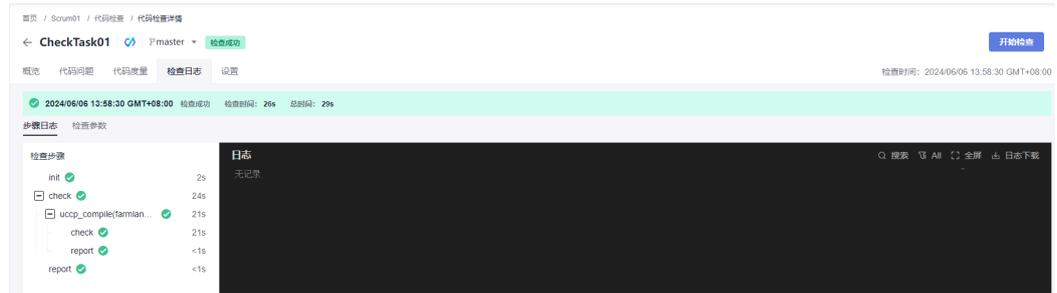
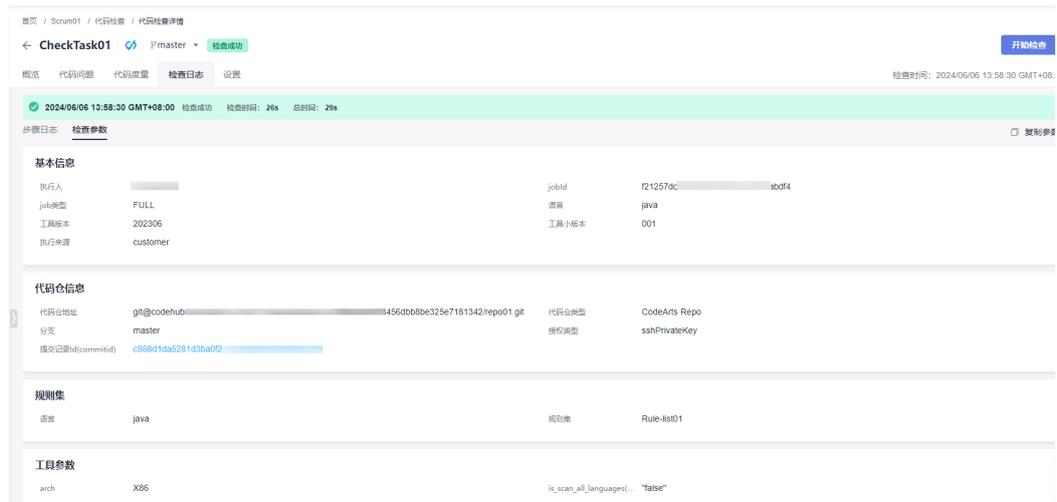


图 8-2 检查参数



相关操作

相关API接口可参考[缺陷管理](#)和[代码度量](#)。

9 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

10 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CodeArts Check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 10-1 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 |
| 创建模型 | model | create |
| 修改模型 | model | modify |
| 删除模型 | model | delete |
| 创建模型集 | modelset | create |
| 修改模型集 | modelset | modify |
| 删除模型集 | modelset | delete |

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](#)。

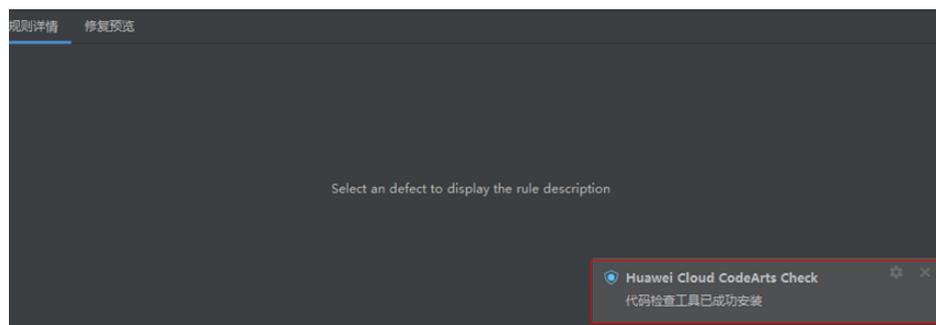
11 参考

11.1 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。

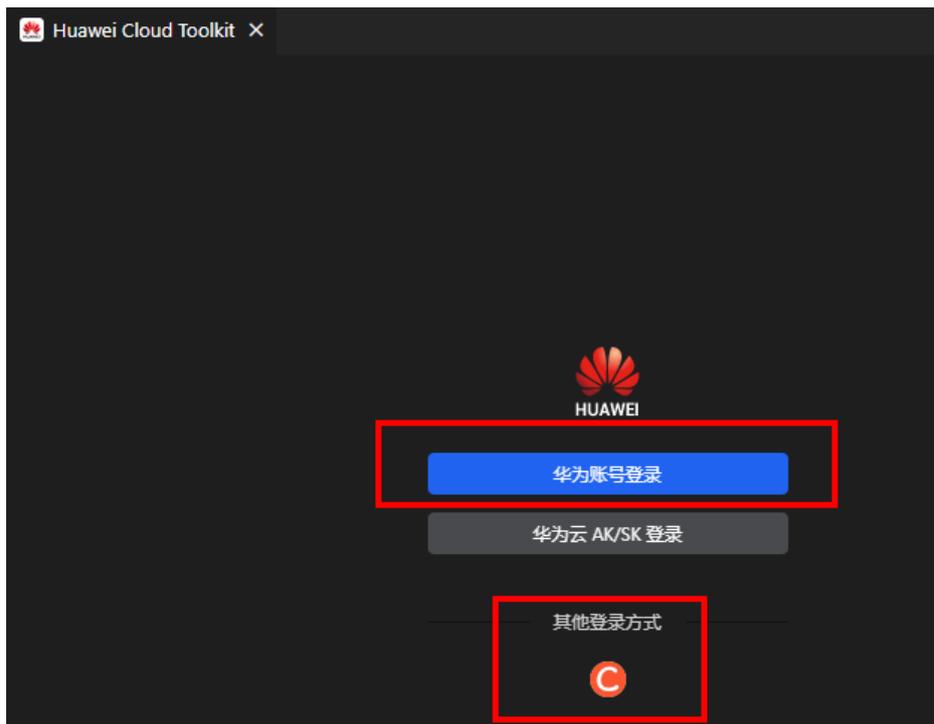


----结束

登录 IntelliJ IDEA 版本插件（可选）

分为两种登录方式。

- **方式一：**通过华为云账号直接登录或CSDN账号授权登录CodeArts Check IDE插件。



- **方式二：通过华为云账号的AK/SK登录。**
 - a. 获取华为云账号的AK/SK。
 - i. 登录华为云[访问密钥界面](#)。
 - ii. 单击“新增访问密钥”，在弹出的窗口中根据实际需要填写描述信息。
 - iii. 单击“确定”，弹出“身份验证”窗口，通过手机的验证方式进行身份验证。

身份验证

×

i 您已开启操作保护，为了保障您的账号和资源安全，请进行身份验证。如需关闭操作保护，请在“账号安全设置>敏感操作”中关闭。 [关闭操作保护](#)

验证方式 手机 邮箱 虚拟MFA ?

手机号码 C 修改

验证码

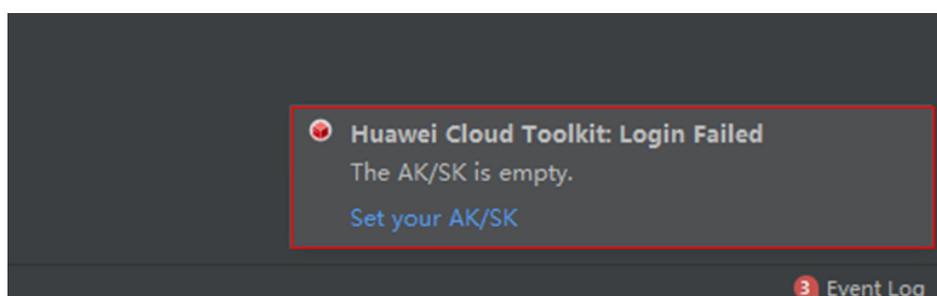
- iv. 输入验证码后单击“确定”，会弹出窗口提示“创建成功”，单击“立即下载”按钮。

 **创建成功**

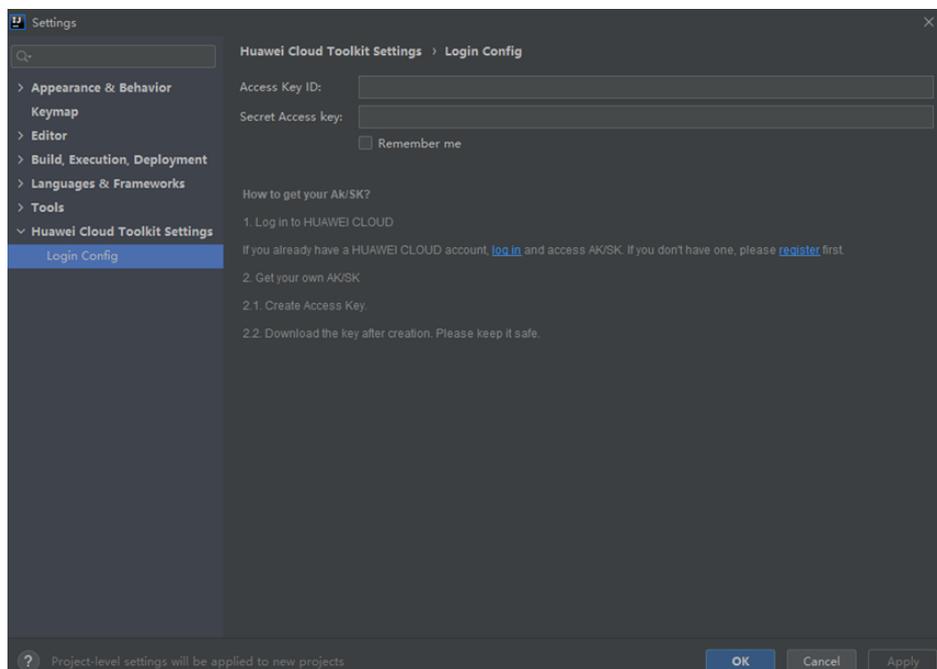
请及时下载保存，弹窗关闭后将无法再次获取该密钥信息，但您可重新创建新的密钥。

[立即下载](#)[取消](#)

- v. 打开下载的“.csv”文件，查看User Name/Access Key/Secret Key信息。
- b. IDEA IDE编辑器右下角提示“Huawei Cloud ToolKit: Login Failed”。



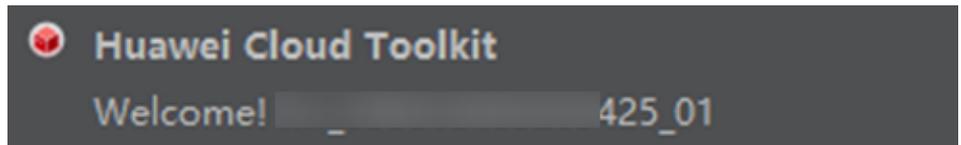
- c. 单击“Set your AK/SK”，打开华为云ToolKit的登录界面窗，输入a中获取华为云账号的AK/SK。
需要勾选“Remember me”。



- d. 单击“Apply”，在弹出的窗口中单击“同意并继续使用”。

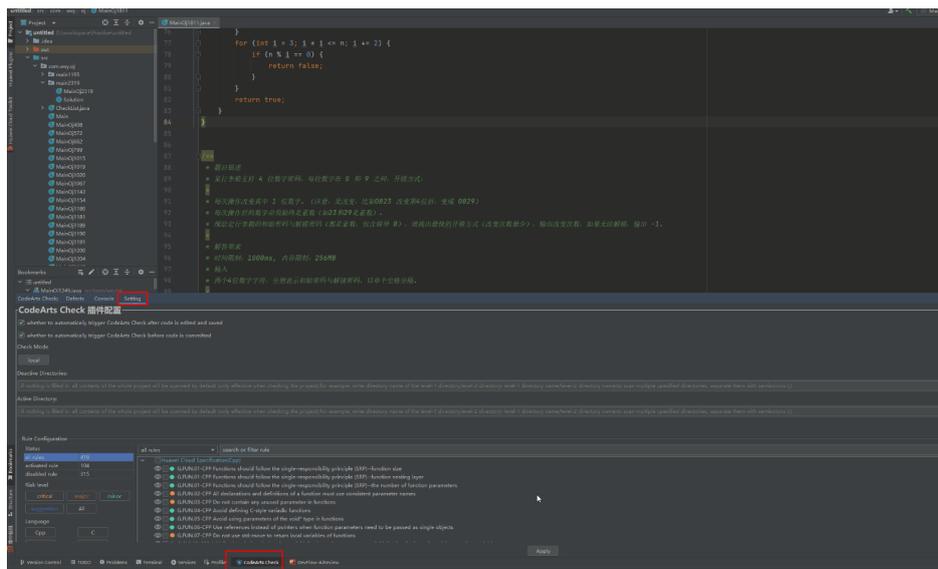


e. IDEA IDE右下角会有登录的用户名提示信息, 表示已经登录成功。



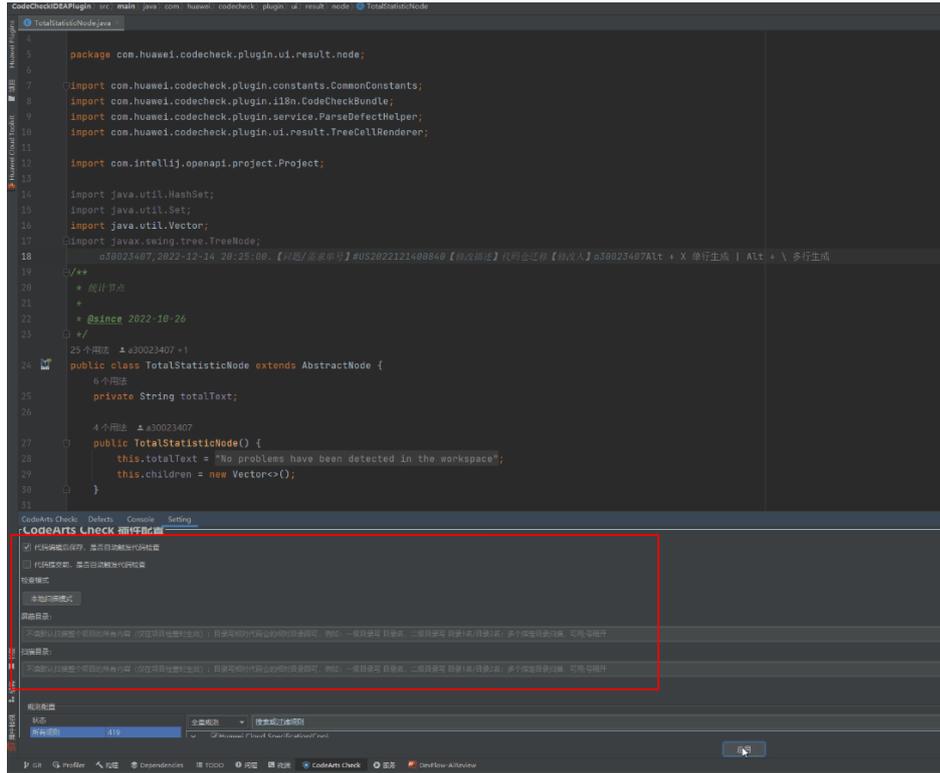
配置 CodeArts Check 插件

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。

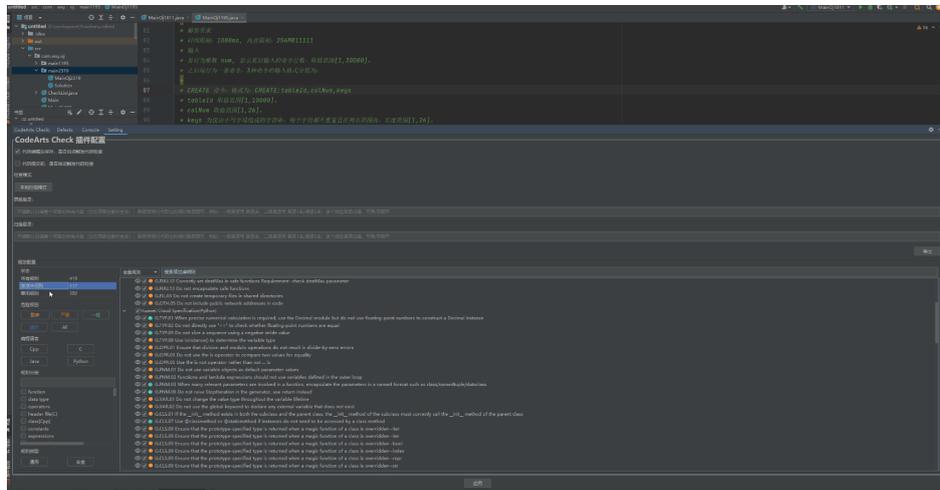


步骤2 单击配置窗口中“规则”左侧的图标可以查看规则详细信息。

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

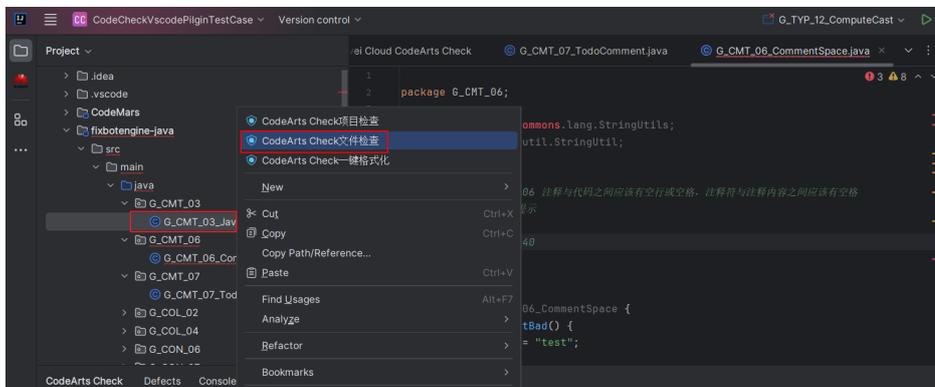
以Java项目为例。

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](#)。

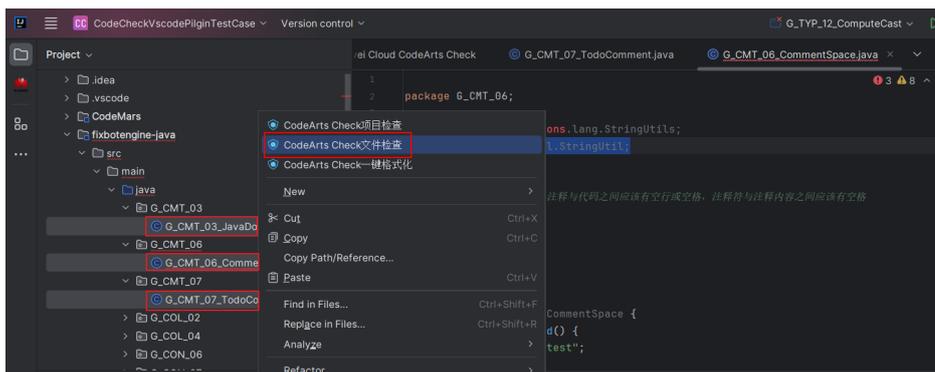
CodeArts Check 插件检查功能介绍

- 扫描检查单文件、多文件、整个项目并查看扫描结果

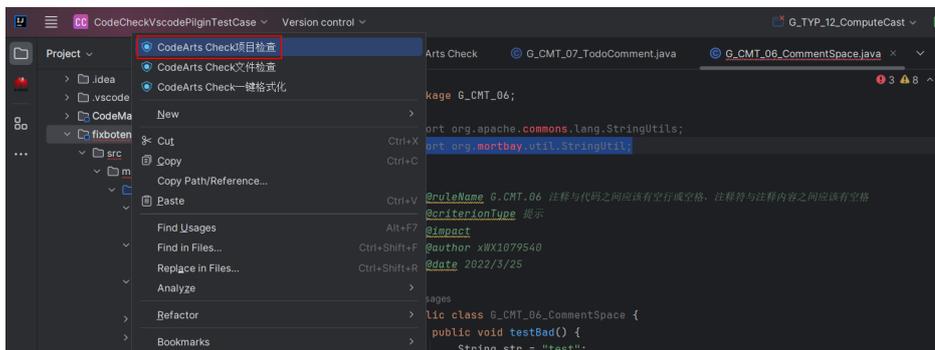
- 单文件扫描检查：在当前打开的文件代码编辑区，单击鼠标右键，选择“CodeArts Check文件检查”。或者在项目的资源管理器界面里，鼠标移动到需要扫描的文件，单击鼠标右键，选择“CodeArts Check文件检查”。



- 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”。

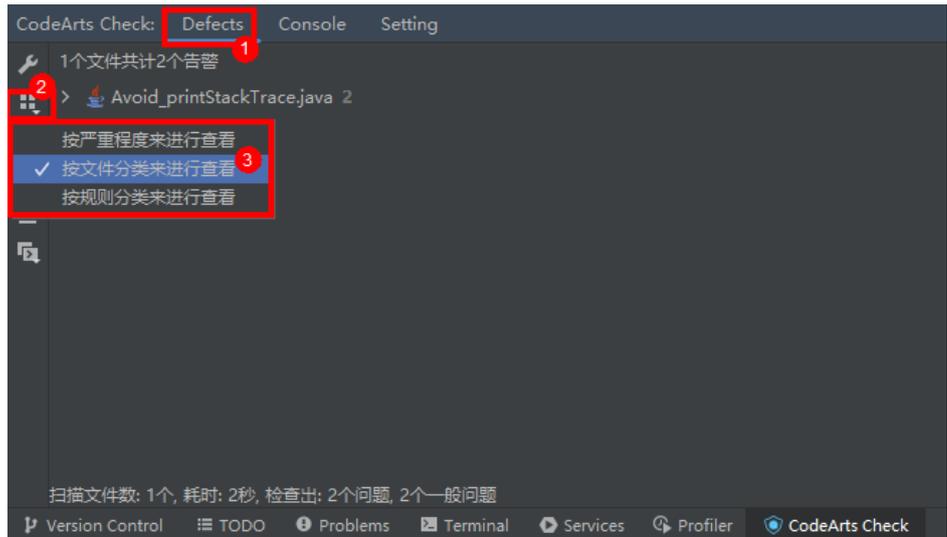


- 整个项目扫描检查：在当前打开的文件代码编辑区，单击鼠标右键，选择“CodeArts Check项目检查”。或者在项目的资源管理器界面里，单击鼠标右键，选择“CodeArts Check项目检查”。

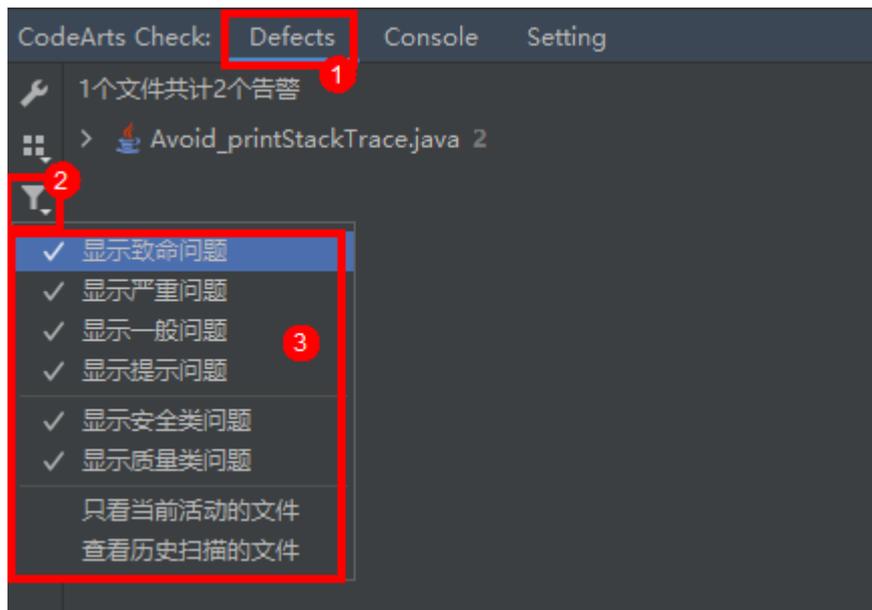


- 告警帮助提供告警详细描述、严重级别、告警类型、正确示例、错误示例及修复建议，您可以根据“严重程度/文件分类/规则分类”筛选查看告警。

- 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。



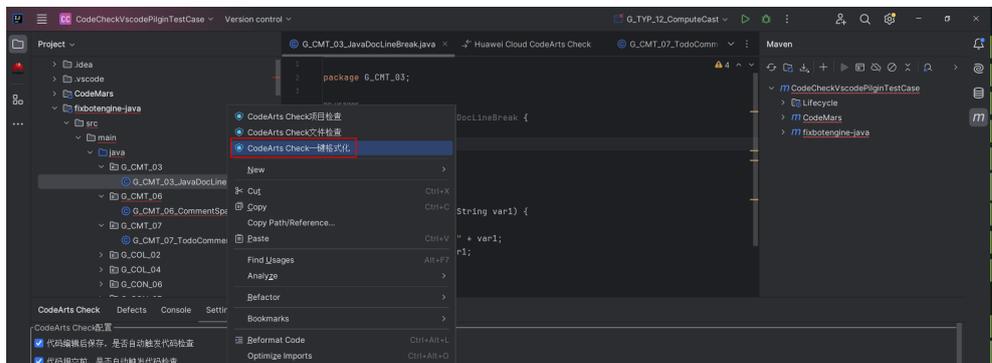
- 根据告警指导修改后，插件将实时自动检查，自动更新告警信息
用户根据插件告警指导修改后，插件将实时自动进行检查，自动更新告警信息。



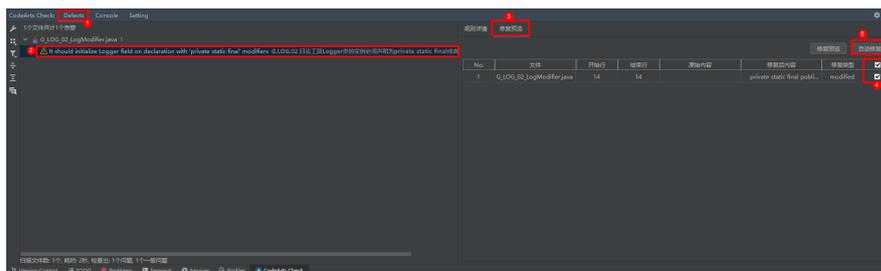
CodeArts Check 一键格式化及自动修复功能介绍

- 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个问题，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。

若未登录，请先登录再使用查看配置，单击打开“Setting”配置界面。若未登录，“Setting”界面提示“单击链接登录后获取配置数据(…)”等相关信息，单击提示下链接“Login config”跳转到登录页面登录。

若已登录，“Setting”页面提示“设置页面已移动到底部CodeArts Check > Setting面板”，单击提示下方链接打开底部Setting面板查看配置。

“CodeArts Check配置”界面功能说明请参见图11-1和表11-1。

图 11-1 CodeArts Check 配置界面

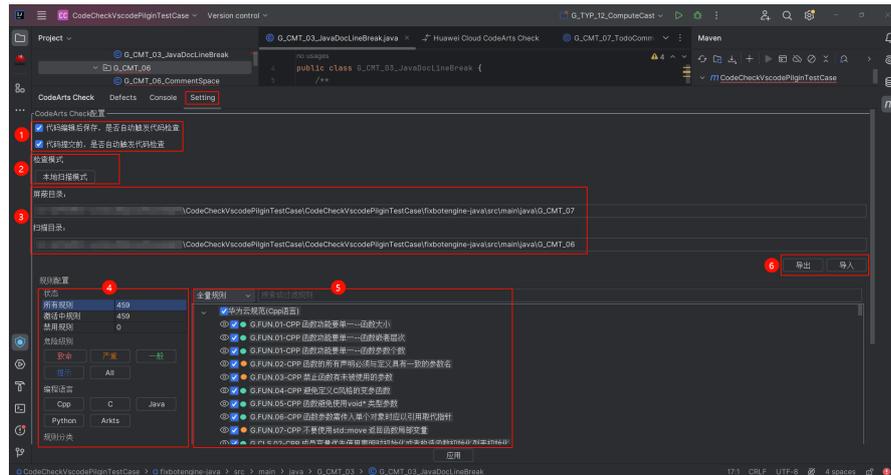


表 11-1 CodeArts Check 配置界面功能说明

| 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。 |
| 3 | 配置扫描排除目录 | <p>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。</p> <p>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）。</p> |
| 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。 |