

Day05.实践打卡任务

使用 DevCloud 静态代码检查确保编码规范有效落地

实践背景

项目经理：随着凤凰商城越来越庞大，线上出现的缺陷以及安全问题也越来越多，修复的成本太大了，其实很多问题是可以在上线之前就应该发现以及修复的。

而且开发人员写代码也比较随性，没有统一标准，我们需要制定一些基本的标准，并对代码进行持续的静态代码扫描，一旦发现问题立即在迭代内修复。这样才能保证我们上线的应用相对可靠。

在本实验中，您将了解如何使用 DevCloud 提供的“代码检查”服务完成针对不同技术栈的代码静态扫描，问题收集与修复。

必做打卡任务

● 打卡任务要求

1. 为示例项目配置代码检查任务并执行

● 准备工作

1. 完成前四日实践打卡任务

● 实践步骤

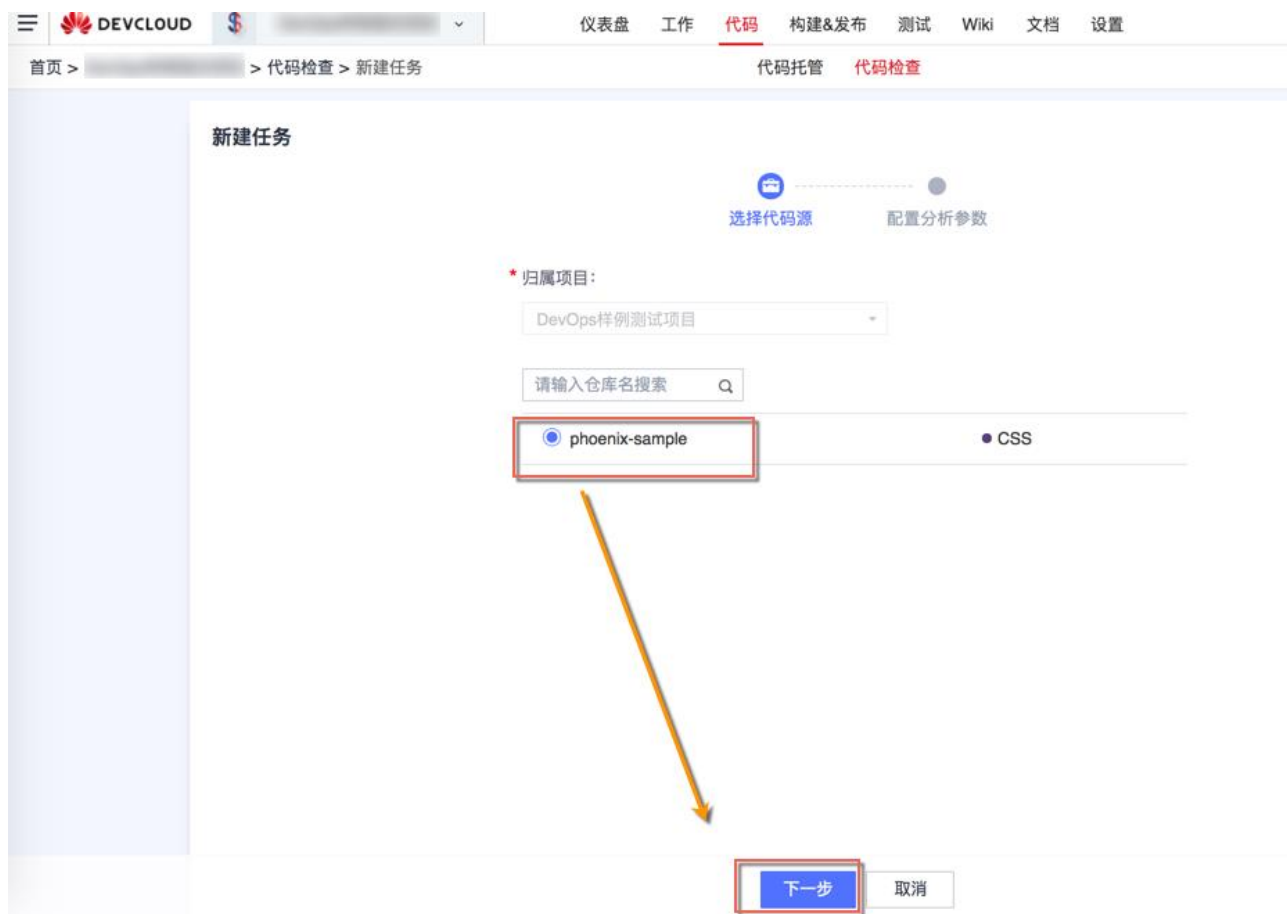
在本练习中，您将学习如何创建代码检查任务，并执行代码检查任务，完成应用的静态代码检查。

- 创建代码检查任务

1. 选择 “代码”>“代码检查”>“新建任务”，如下图所示：



2. 输入任务名称 “phoenix-sample-code-check-java”，设置代码分支为 master，选择检查语言 Java & Python，点击确定，如下图所示：





3. 配置完成后，如下图所示：

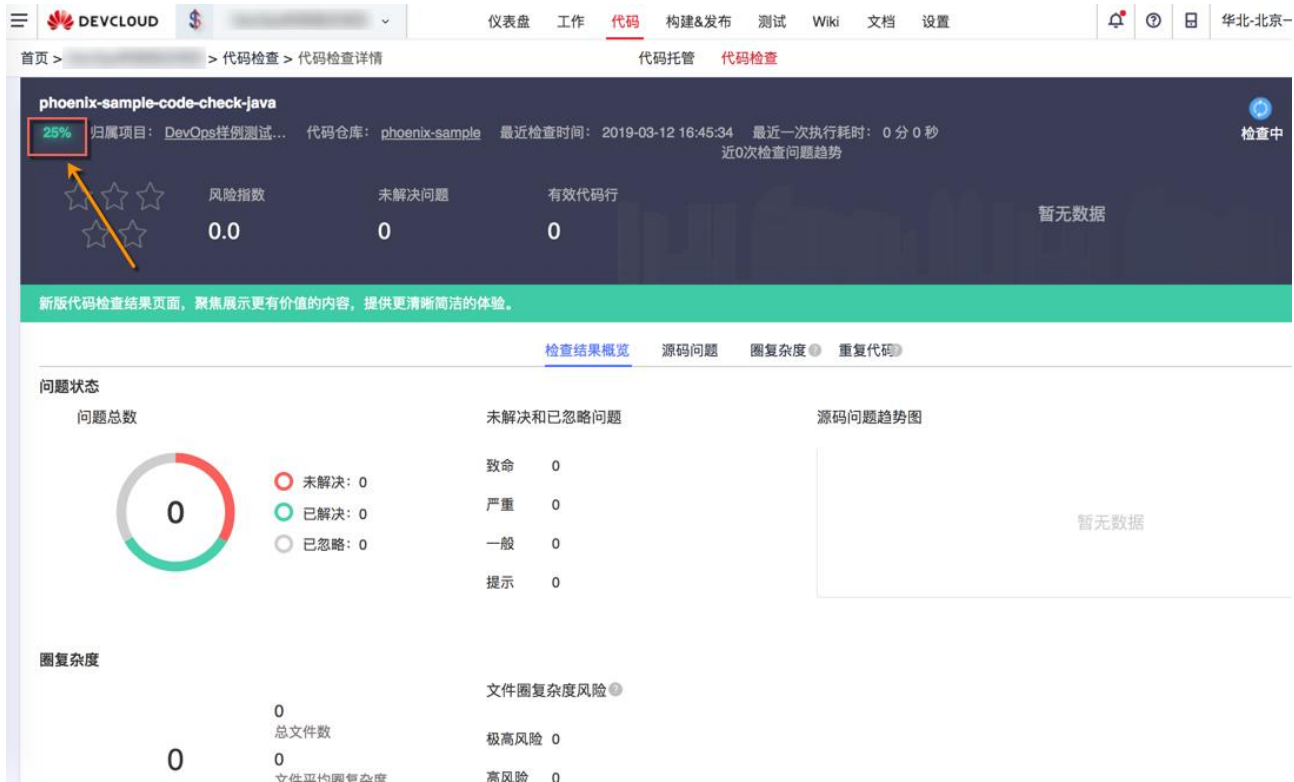


- 启动代码检查任务

1. 点击启动任务，启动 Java 代码静态检查，如下图所示：



2. 代码扫描中，会显示代码的进度百分比，如下图所示：



3. 扫描完成后，如下图所示：



至此 day05 必做的实践打卡任务已完成。

打卡任务截图

点击“代码”>“代码检查”并截图，截图中需包含项目名称、以及对应的代码检查任务

● 截图示例：



● 截图验证点：

1. 项目名称
2. 用户名
3. 代码检查任务执行情况

选做打卡任务-分析代码检查结果并修复问题

在本练习中，您将学习如何根据代码检查结果分析代码中存在的问题，并解决代码问题。

- 分析代码检查报表

1. 如下图所示，当前扫出了 9 个代码问题



2. 点击“源码问题”，切换到代码问题详情界面，如下图所示：

当前界面列出了问题的严重级别、状态、所属文件、以及检查规则等

phoenix-sample-code-check-java

成功 归属项目: DevOps样例测试... 代码仓库: phoenix-sample 最近检查时间: 2019-03-12 16:45:34 最近一次执行耗时: 3 分 44 秒

风险指数 65.0 未解决问题 9 有效代码行 84

新版代码检查结果页面, 聚焦展示更有价值的内容, 提供更清晰简洁的体验。

检查结果概览 **源码问题** 圈复杂度 重复代码

问题描述	问题级别	问题状态	问题分类	检查规则	语言	所在文件	操作
<input type="checkbox"/> Add an end condition to this loop.	致命	未解决	编码问题	循环不应该是...	Java	worker/src/main/java/worker/Worker.java	
<input type="checkbox"/> Add password protection to this database.	致命	未解决	编码问题	数据库应该受...	Java	worker/src/main/java/worker/Worker.java	
<input type="checkbox"/> Use try-with-resources or close this 'PreparedSt...	致命	未解决	编码问题	资源关闭	Java	worker/src/main/java/worker/Worker.java	
<input type="checkbox"/> Use try-with-resources or close this 'PreparedSt...	致命	未解决	编码问题	资源关闭	Java	worker/src/main/java/worker/Worker.java	
<input type="checkbox"/> Use try-with-resources or close this 'Connection...	致命	未解决	编码问题	资源关闭	Java	worker/src/main/java/worker/Worker.java	
<input type="checkbox"/> Use try-with-resources or close this 'PreparedSt...	致命	未解决	编码问题	资源关闭	Java	worker/src/main/java/worker/Worker.java	
<input type="checkbox"/> Either re-interrupt this method or rethrow the 'Int...	严重	未解决	编码问题	'InterruptedException...	Java	worker/src/main/java/worker/Worker.java	
<input type="checkbox"/> Use a logger to log this exception.	一般	未解决	编码问题	不要调用Throw...	Java	worker/src/main/java/worker/Worker.java	
<input type="checkbox"/> Use a logger to log this exception.	一般	未解决	编码问题	不要调用Throw...	Java	worker/src/main/java/worker/Worker.java	

3. 切换到 Python 扫描结果, 鼠标移至最后一个问题, 查看此问题

通过查看问题描述我们可以看到此问题是一个编码风格的问题, fieldStorage 这个变量的命名规则不符合相应的正则表达式, 但是在凤凰团队内部, 其实并没有对此有严格的要求, 我们可以在此忽略此问题

phoenix-sample-code-check-python

成功 归属项目: DevOps样例测试... 代码仓库: phoenix-sample 最近检查时间: 2019-03-12 16:52:42 最近一次执行耗时: 1 分 1 秒

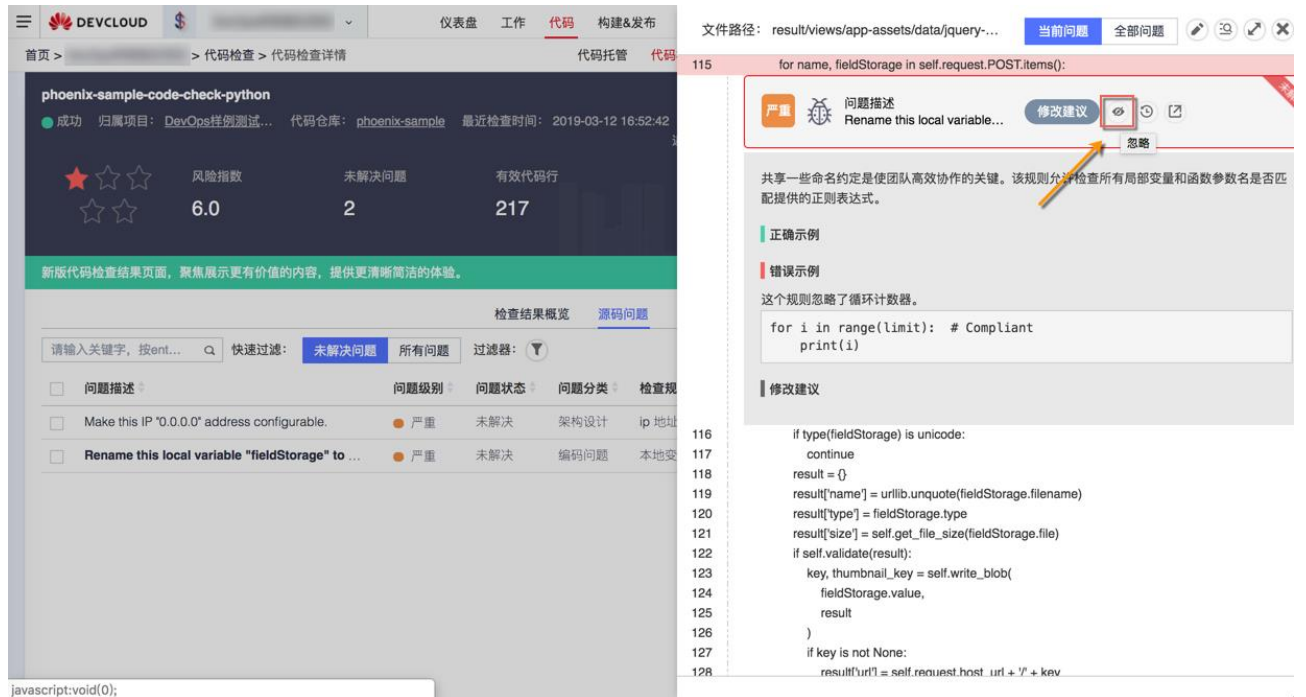
风险指数 6.0 未解决问题 2 有效代码行 217

新版代码检查结果页面, 聚焦展示更有价值的内容, 提供更清晰简洁的体验。

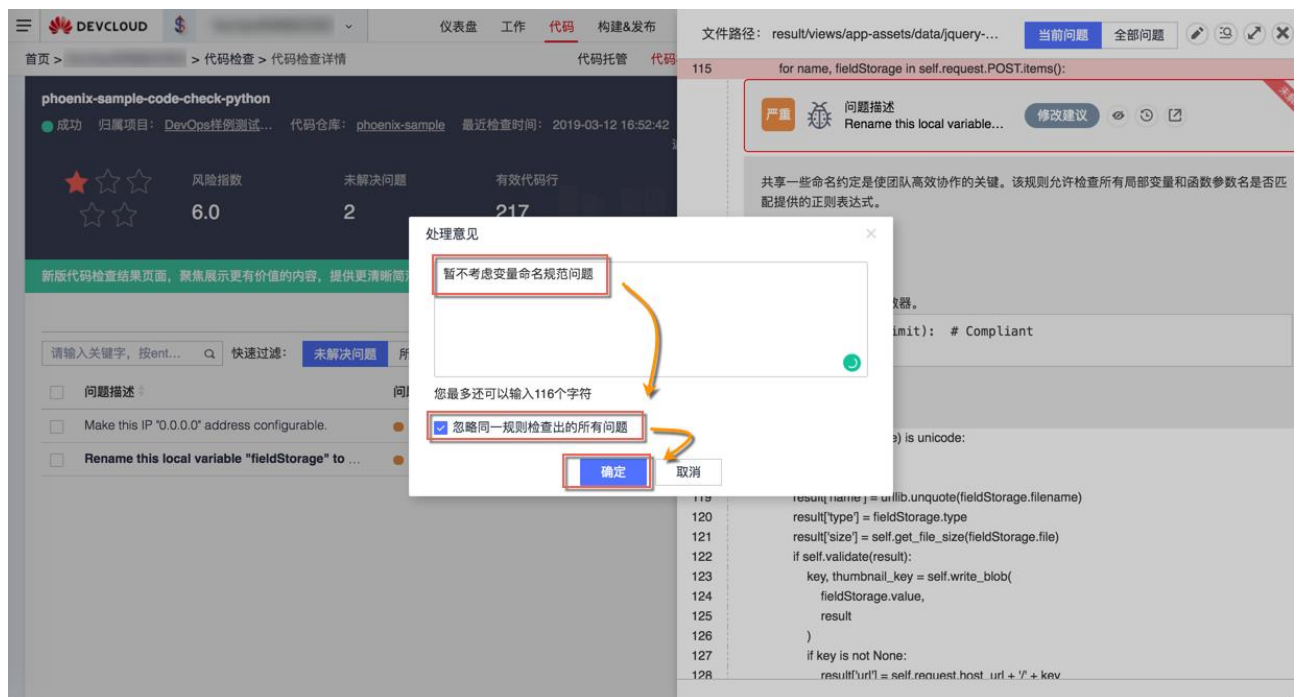
检查结果概览 **源码问题** 圈复杂度 重复代码

问题描述	问题级别	问题状态	问题分类	检查规则	语言	所在文件	操作
<input type="checkbox"/> Make this IP '0.0.0.0' address configurable.	严重	未解决	架构设计	ip 地址不应该...	Python	vote/app.py	
<input type="checkbox"/> Rename this local variable 'fieldStorage' to matc...	严重	未解决	编码问题	本地变量和函...	Python	result/views/app-assets/data/query-file-u...	

4. 点击具体错误, 进入到错误详情界面, 可以通过点击忽略按钮, 忽略此问题, 如下图所示:



5.输入“处理意见”以及勾选“忽略统一规则检查出的错误”，如下图所示：

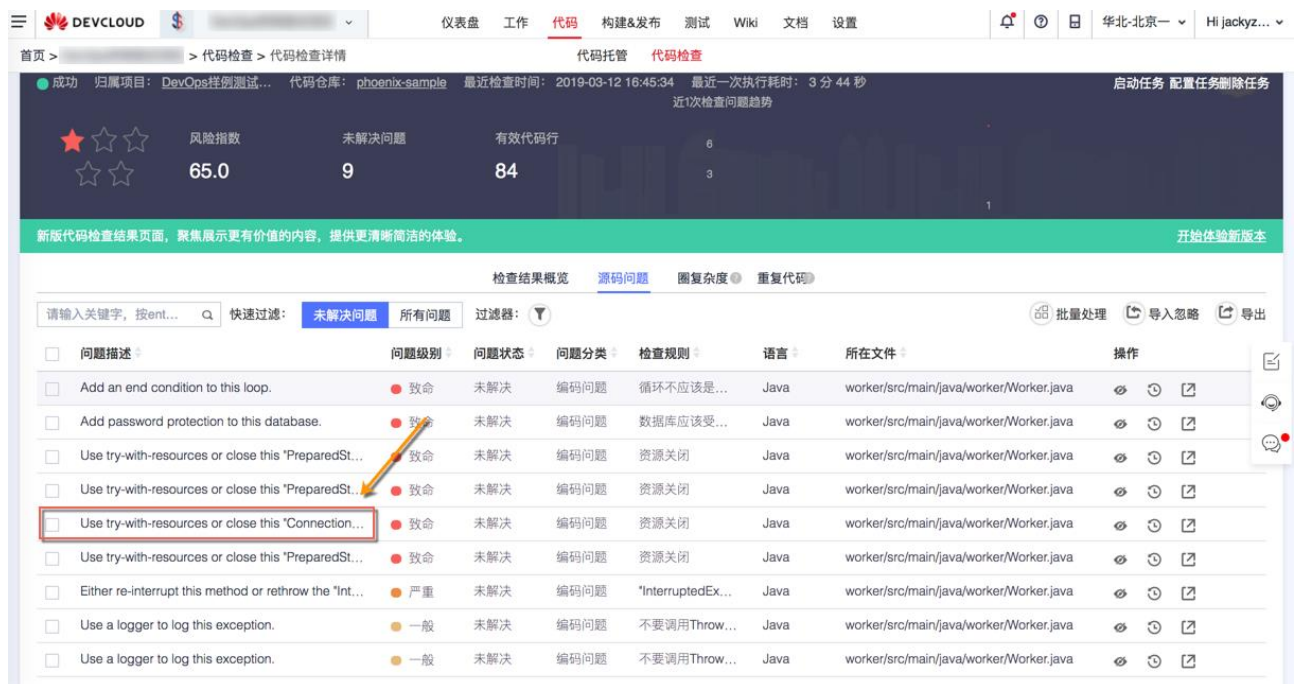


6.回到“检查结果概述”，查看问题状态汇总已经发生了变化，如下图所示：



7.继续查看代码问题，查看第五个问题，如下所示：

此问题是在代码里 Connection 没有关闭可能会引起的资源泄漏问题 属于非常严重的问题，建议快速修复掉



8.可以参考修改建议对代码问题进行修复，如下图所示：

DEVCLLOUD

仪表盘 工作 代码 构建&发布

首页 > > 代码检查 > 代码检查详情

成功 归属项目: DevOps样例测试... 代码仓库: phoenix-sample 最近检查时间: 2019-03-12 16:45:34

风险指数

65.0

未解决问题

9

有效代码行

84

新版代码检查结果页面，聚焦展示更有价值的内容，提供更清晰简洁的体验。

检查结果概览

源码问题

请输入关键字，按ent...

快速过滤: 未解决问题 所有问题

过滤器: Y

问题描述	问题级别	问题状态	问题分类	检查规则
<input type="checkbox"/> Add an end condition to this loop.	致命	未解决	编码问题	循环不
<input type="checkbox"/> Add password protection to this database.	致命	未解决	编码问题	数据库
<input type="checkbox"/> Use try-with-resources or close this "PreparedSt...	致命	未解决	编码问题	资源关
<input type="checkbox"/> Use try-with-resources or close this "PreparedSt...	致命	未解决	编码问题	资源关
<input type="checkbox"/> Use try-with-resources or close this "Connect...	致命	未解决	编码问题	资源关
<input type="checkbox"/> Use try-with-resources or close this "PreparedSt...	致命	未解决	编码问题	资源关
<input type="checkbox"/> Either re-interrupt this method or rethrow the "Int...	严重	未解决	编码问题	"Intern
<input type="checkbox"/> Use a logger to log this exception.	一般	未解决	编码问题	不要调
<input type="checkbox"/> Use a logger to log this exception.	一般	未解决	编码问题	不要调

文件路径: worker/src/main/java/worker/Worker.j... 当前问题 全部问题

75 conn = DriverManager.getConnection(url, "postgres", "");

致命

问题描述

Use try-with-resources or c...

修改建议

不能依靠java垃圾回收去清理所有资源。尤其是connections、streams、files以及其他一些实现了 **Closeable** 接口或者其super-interface、**AutoCloseable** 接口的类,它们必须在使用后手动关闭。不这样做的话会导致应用程序资源泄漏。

例外:

Java7的新特性 the try-with-resources 语句块可以毫无保留关闭 **Closeables**, 所有在此语句块中打开的资源都会忽视此规则。

```
try (BufferedReader br = new BufferedReader(new FileReader(fileName))
//...
}
catch ( ... ) {
//...
}
```

正确示例

```
OutputStream stream = null;
try{
    stream = new FileOutputStream("myfile.txt");
    for (String property : propertyList) {
        // ...
    }
}catch(Exception e){
    // ...
}finally{
    stream.close();
}
```