

项目重构报告

行为定义

在计算机编程和软件设计中，代码重构是重组现有计算机代码的过程——改变过程，而不改变其外部行为。重构的目的是改善软件的设计、结构和/或实现（其非功能属性），同时保留其功能。重构的潜在优势可能包括提高代码的可读性和降低复杂性；这些可以提高源代码的可维护性，并创建一个更简单、更干净、或更有表现力的内部架构或对象模型，以提高可扩展性。重构的另一个潜在目标是提高性能；软件工程师面临着一个持续的挑战，即编写性能更快或者使用更少内存的程序。

重构动力

本项目为前后端分离项目。

前端或后端在开发的过程中，对于数据接口、数据展示在不同的时间阶段有不同的要求、预期。当项目进入进入新时期时，过去的代码尽管能够表现正确的外在行为，但其内部实现于高可读性、低复杂性、高可维护性、高扩展性上表现不佳。

重构的动机通常是注意到代码的异样。例如，某一方法可能非常长，或者它可能与附近的另一个方法几乎重复。一旦发现，这些问题就可以通过重构源代码来解决，或者将其转化为一种新的形式，其行为与以前一样，但不再有异样了。

对于一个长的过程，可以提取一个或多个较小的子过程；或者对于重复的过程，可以删除重复的部分，用一个共享函数代替。不进行重构会导致技术债务的积累；另一方面，重构是偿还技术债务的主要手段之一。

常用手段

本项目的开发中使用了許多业界领先的重构技术。基于成熟的重构技术进行重构，可以大大缩短团队的开发时间并保证开发质量。

- 引入变量

将光标高亮的代码部分抽离出原本的语句（通常是链式调用）作为单独的变量（或常量），开发者可以对引入的变量进行额外的查看、操作。

```

@GetMapping("/test")
public PageResult<ProjectOutline> getProjects(
    @RequestParam("userId") Long userId,
    @RequestParam("userRole") Role userRole,
    @RequestParam("page") Integer page,
    @RequestParam("pageSize") Integer pageSize
) {
    return projectService.findProjectOutlines(page, pageSize, userId, userRole);
}

@GetMapping("/test")
public PageResult<ProjectOutline> getProjects(
    @RequestParam("userId") Long userId,
    @RequestParam("userRole") Role userRole,
    @RequestParam("page") Integer page,
    @RequestParam("pageSize") Integer pageSize
) {
    PageResult<ProjectOutline> projectOutlines = projectService.findProjectOutlines(page, pageSize, userId, userRole);
    return projectOutlines;
}

```

- 提取方法

将光标高亮的代码部分抽离出原本的方法作为单独的、依实际情况可能有参数、依实际情况可能有返回值的新的方法。开发者将多处重复的代码段提取为新方法并在重复处替换为新方法。

```

@Override
public ProjectFormIds getProjectFormIds(String projectId) {
    Project project = projectDAO.findProjectById(projectId);
    if (project == null) {
        throw new ProjectNotFoundException("该测试项目不存在");
    }
    return project.getProjectFormIds();
}

@Override
public ProjectFormIds getProjectFormIds(String projectId) {
    Project project = projectDAO.findProjectById(projectId);
    extracted (project);
    return project.getProjectFormIds();
}

private void extracted(Project project) {
    if (project == null) {
        throw new ProjectNotFoundException("该测试项目不存在");
    }
}

```

后端重构记录

后端共进行了 3 次重构。

第 1 次重构

重构动机：现有的委托服务、合同服务的接口与实际业务流程稍有不同，与实际业务流程的结合的紧密程度不高；团队期望通过重构，提升代码的可读性、可维护性。

参与人员：孙逸扬

时间起止：2022.05.19 - 2022.05.21

重构内容：重构委托服务、合同服务，相应地调整用户服务、网关服务

提交记录：

```
from 6f61722b660115de2554dacbadb277b1f11d92aa to  
7967bf22ef571fd8e6a00ddffc6aa3fa77dc4f5a
```

```
119      files changed  
4,832    additions  
2,850    deletions
```

第 2 次重构

重构动机：测试服务的状态原设计由各子数据表自行维护，在实际前端展示时不够自然合理；团队期望通过抽取各子数据表的状态、结合业务流程，将项目的总体状态抽象并集中于测试项目数据结构中。重构后，前后端无需再维护各子数据表的状态，只需根据测试项目的状态即可把握流程。

参与人员：杨茂琛、王岳、姜宁、戴显灏

时间起止：2022.06.05 - 2022.06.22

重构内容：重构测试服务的状态相关内容、数据表之间交互相关内容

提交记录：

```
from 9e2697eb6bf484f1c3e4bc0502eff5242b385d46 to  
2214fec3686989cc6dcc881f509828447d83acf2
```

```
83      files changed  
1,152    additions  
868      deletions
```

第 3 次重构

重构动机：通过使用静态分析工具 codeQL 扫描代码，团队发现了潜在的安全威胁；团队期望通过如加强用户输入校验、过滤用户输入等方式消除已知的安全威胁。

参与人员：杨茂琛、张世茂

时间起止：2022.06.29 - 2022.06.29

重构内容：重构文档服务中前端传入的内容作为文档路径或方法参数的相关内容

提交记录：

from f37535626241b02001bb6339479e1db25fecae38 to
9c8a16a26084f3add39e5d4da855ee6adb06ecf6

21 files changed
257 additions
78 deletions