

迭代一后端总结文档

TagMaker

2018 年 3 月 24 日

目 录	2
-----	---

目录

1 后端开发历程简述	3
2 后端设计中的重要决策	3
3 后端技术简介	4
4 后端有待完善的地方	5
5 后端迭代二中的目标	5

1 后端开发历程简述

在迭代一的过程中，后端开发整体较为顺利，基本能够按计划完成。开始开发时为后端分配了三位同学，后来根据实际任务量调整至两位同学。两位同学先是分成了 developer 和 QA 两个角色，分别学习各自的技术，QA 同学搭建好持续集成、质量检测、测试环境，developer 同学构建基本的 springboot 项目，做了简单的设计之后两位同学分层开发，分别负责逻辑层和数据层。开发完毕后由 QA 同学主要负责测试，developer 同学主要负责修复，在各种工具的帮助下很快将单元测试覆盖率达到 60% 以上。之后后端又对一些细节做了加强，比如动态地上传本地文件，这样可以在运行时加载图片。比如对代码质量进行了一定的提升，改善了一些编码风格。

2 后端设计中的重要决策

REST

软件使用前后端分离的 restful 接口进行开发，因为这样可以前后端并行开发，适应迭代一快速而简明的任务要求。同时 REST 接口将前后端的联系限制在数据本身，降低了耦合度，有利于之后实现功能的扩充。

分层

后端的逻辑层和数据层使用分层开发的模型进行构建。因为迭代三有极大可能会使用数据库来增加对数据的处理能力，因此分层有利于实现这个变更。

3 后端技术简介

Springboot

Springboot 是后端代码构造过程中使用的主要工具。我们主要使用了 Springboot 的 IoC 技术来控制类与类之间的协作，用 AOP 技术来对异常进行了捕获和处理，用 Springboot 自带的多线程 (@Scheduled) 进行了方便快捷的多线程编程。

Gradle

gradle 主要负责依赖管理、测试、生成测试报告

Jacoco

jacoco 和 gradle 配合，用来生成覆盖率报告

TestNG

测试工具

Jenkins

持续继承工具，每一次提交到 git 上之后，Jenkins 会自动进行构建和测试并生成晴雨表。除此之外，每天的固定时间会执行一次构建。

Sonar

进行代码质量检测，并给出修复建议

RestClient

用于方便地进行 Rest 接口的测试，并用于生成 Rest 规格文档

4 后端有待完善的地方

- 由于复杂度较低，没有使用测试驱动开发
- 回归测试不够充分，测试阶段注入 bug 较多
- 新技术较多，有的利用不充分
- 没有自动部署

5 后端迭代二中的目标

- 充分利用自动部署工具
- 随着需求的复杂化，重构面向前端的 `restController` 的组织形式
- 增加一些统计/计算相关的模块，增加较为复杂的算法
- 增加单元测试力度，使测试更充分，并设立回归测试用例组进行回归测试