

Lab3 会议组织与投稿

2020年 软件工程课程系列实验

注：本次 Lab 为小组实验。实验的截止日期是2020年4月13日 24:00

1. 实验目标

现在，你的系统已经实现了 用户注册/登录 的功能，并且有了一个简单的会议申请界面。

本次Lab中，我们的主要目标是对 会议申请与管理 相关功能进行完善。

2. 实验准备

在正式开始本次实验之前，你需要先完成一些准备工作。

2.1 完善Lab2

在实验过程中，你可能有过这些想法：

- 这里虽然能用，但是还不太完美，等我有时间了，再将它完善一把！
- 这里有个小bug，但是问题不大，我还是先赶进度，小bug就以后再说吧！
- 为什么别人的界面那么好看，我也要做好看的界面！
- 我这里的架构设计好像有点问题，但是没时间重构了，就这样吧！
-

你可能有过很多很多的想法，然而，你的身体却很诚实。距离 Lab2 结束已经好几天了，你是不是在截至日期之后，就没有再打开过项目呢？

所以正式开始 Lab3 之前，首先你需要对 Lab2 中未完成的功能进行完善，对还没有来得及修复的bug进行修复。否则放任不管的话，它们迟早有一天会再次找上门来。

2.2 取个名字吧

现在我们马上要开始新一轮迭代了，我们的项目当然不能再叫 Lab2 了（当然，叫 Lab3 也没有必要）。如果你比较细心，相信你有发现，我们助教给自己的项目取名为 `HardChair`，小组名称则为 `助教小分队`。

同样，你们也需要给自己的项目和小组取个名字了。在取好响亮的名称后，请完成以下操作：

- 组长在 DevCloud 中将自己组的项目名称修改为 `2020软件工程_<项目名称>_<小组名称>`，在项目描述中介绍组内成员。例如下图所示。

基本信息

项目名称：

2020软件工程_HardChair_助教小分队

✓

项目描述：

做最👏的学术会议投稿审稿系统。
小组成员：刘弋，严伟羽，赵成原，李博文

项目类型：

Scrum流程

创建时间：

2020/03/13 22:58:32 GMT+08:00

创建人：?

FudanSoft

▼

删除项目

归档

保存

- 所有同学在 DevCloud 中个人头像处点击 设置昵称，将昵称设置为自己的真实姓名。

另外，我们发现仍有少数同学没有设置 Git 中的姓名。再次强调，请利用 `git config` 相关命令设置自己的真实姓名，以体现个人贡献。

2.3 打标签

Git 可以给仓库历史中的某一个提交打上标签，以示重要。比较有代表性的是人们会使用这个功能来标记发布结点（`v1.0`、`v2.0` 等等）。

现在，你的 Lab2 已经完成了，项目已经有了雏形，这也是你们的一次标志性事件，那么，给它打一个tag吧。

在正式开始 Lab3 的开发之前，请寻找到前后端仓库中 **Lab2**截止日期前的最后一次commit，将其打上 lab2-finish 的标签。

具体应该如何操作，请 [RTFM](#) 吧！

在执行完打标签的命令后，可以利用 `git log` 查看标签是否成功打上，如下图所示。

```
libw5@DESKTOP-LBW MINGW64 ~/Develop/IdeaProjects/HardChair (master)
$ git log
commit ee7d2f22de9de30c4af8330931d25e8b34b26e7f (HEAD -> master, origin/master, origin/HEAD, huawei/master)
Author: LeBW <27091925@qq.com>
Date: Fri Mar 27 21:30:14 2020 +0800

    refactor project name

commit 5ece09de61b0b9e2f2866315584a494ba4ca95f0
Author: LeBW <27091925@qq.com>
Date: Wed Mar 25 11:43:54 2020 +0800

    add conference test

commit 5f59d9a8ce6a75fdffb84459a6d397309d3484ec (tag: lab2-finish)
Author: LeBW <27091925@qq.com>
Date: Wed Mar 18 12:29:08 2020 +0800

    add conference

commit 8ddcfbf4e47e6176b1fea54ea07742a0d5bc4763
Author: LeBW <27091925@qq.com>
Date: Sun Mar 15 01:27:16 2020 +0800

    delete SecurityConstant.java
    add Configproperties dependency

commit f02f73d522ec918a33603819389928cc7d4ad43f
Author: LeBW <27091925@qq.com>
Date: Sun Mar 15 01:02:01 2020 +0800

    add configProperties
```



成功打上标签后，你需要将其push至华为云仓库中。这一步完成后，你需要能够在华为云仓库中看到相应标签，即代表成功。

HardChair_Backend				
创建者: FudanSoft 创建时间: 2020-03-13 23:49:07 最近更新: 2020-03-29 11:47:35				
文件 分支 标签 合并请求 成员 关联工作项 仓库统计 仓库网络 设置				
标签列表 标签可以标记已经部署的版本				
请输入关键字，按Enter键搜索				
标签名	关联提交	更新时间	下载	操作
lab2-finish	5f59d9a8ce6a75fdffb84459a6d397309d3484ec	10天前	ZIP TAR GZ	
lab2	5c2f12107f1911a02	15天前	ZIP TAR GZ	

Tag 是 Git 提供的一个强大的功能。你会发现，在实验提交章节中，我们会再次要求你进行与标签有关的操作。

3. 实验内容

下面开始正式介绍本次实验的主要内容了。本次实验的主要目标是对会议申请与管理相关功能进行完善。

如果你对于本系统用户类型还不太熟悉，可以参考Lab2的附录。

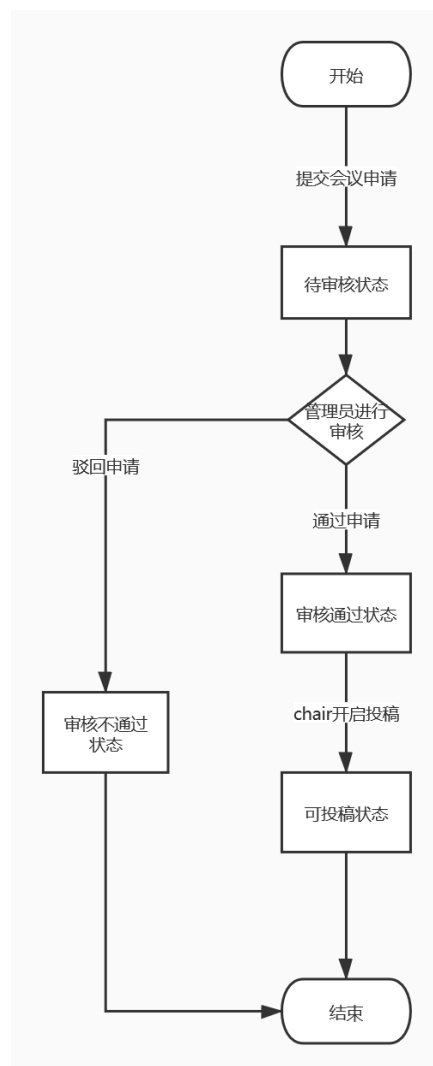
3.1 需要实现的功能

功能列表	详细功能描述
用户注销	用户在登录系统之后，可以进行用户的注销，在用户点击注销按钮的时候，需要让用户对其注销操作进行确认（是否确定注销用户）。在用户成功进行账号的注销之后系统自动跳转到登录页面。
会议审核	管理员对于会议申请进行审核，可以通过会议申请或者驳回会议申请，当会议申请通过之后，申请人自动成为该会议的chair。申请者可以在系统内查看自己提交的申请的状态，比如说审核中，已通过和未通过。此次实验中，管理员只需要实现这一个功能。
会议及角色选择	在用户登录系统之后，如果想要对会议事务进行处理，就首先要确定自己所要处理的会议以及以什么身份进入该会议。首先系统应该列出该用户参与的所有会议，在用户选择完会议之后，系统应列出在该用

户在该会议中的所有角色。只有这两个选项都选择完成之后，才能对其角色相关的事务进行处理。

邀请会议的PC member	当用户以chair的身份进入某个会议之后，就可以邀请系统用户成为该会议的PC member。chair按照用户姓名进行查询，查出所有相关的用户（有可能有重名的用户）以及用户的详细信息，如邮箱，区域等。chair勾选一个或多个用户发送邀请。chair还可以对自己发送邀请的状态进行查询，邀请状态可以是待确认，已同意或者已拒绝。
接受邀请并成为PC member	用户可以在会议邀请通知界面查看所有未处理的PC member邀请，系统显示的邀请的具体信息包括chair的姓名，会议的名称等。对于每个邀请用户可以同意或者拒绝，如果同意邀请，则会立即成为该会议的PC member。
开启投稿	当用户以chair身份进入某个会议，并且该会议的状态是审核通过时，用户可以开启该会议的投稿。当该会议投稿开启之后，其他系统用户才能在该会议中进行投稿。
论文投稿	在用户登录进入系统之后，除了可以以某一具体的角色进入到某个具体的会议处理事务之外，还可以直接查看当前所有的会议列表（列表中显示的会议一定是通过审核的）。所有的会议还需要显示其状态（审核通过和投稿中）。选中一个会议，如果该用户不是该会议的chair，则可以直接进行投稿，投稿的时候需要填写论文的标题（长度限制50个字符），摘要（长度限制800个字符），还需要进行论文的文件上传（只要pdf文件），当前上传者就是论文作者。

此次lab会议的状态图如下所示：



3.2 注意事项

- 在进行用户注册的时候，需要加上一个必填字段为用户姓名。
- 一个会议的chair不可以向该会议投稿。
- 对于管理员来说，其登录系统后只能进行会议审核和用户注销，不能实现其他的功能。

4. 实验要求

除了功能需求之外，我们还从三个方面，对本次实验提出了具体要求。

4.1 实验过程要求

4.1.1 需求规划

在实验过程中，使用 DevCloud 中提供的需求规划功能或者看板功能进行需求规划。

4.1.2 Git协同开发

在实验过程中，使用 Git 进行小组协同开发。良好的 Git 使用习惯包括且不限于：

- 及时commit，以及规范的commit信息。
- 合适的分支管理策略。

4.2 测试要求

在上次的实验中，我们对大家测试这部分没有严格的要求。只是提醒大家按照各自的方法，编写测试用例，对系统的功能等方面尽可能进行完备的测试。有的同学等到系统全部做完了，再来测试，结果发现一大堆问题。不是这里有问题，就是那里有问题。所以这次，我们要求大家学会“单元测试”，并保证测试覆盖率不低于80%。（注：主要是对后端进行单元测试，前端可自行功能测试）

什么是单元测试呢？单元测试就是针对最小的功能单元编写测试代码。Java程序最小的功能单元是method，因此，对Java程序进行单元测试就是针对单个Java方法的测试。但是有很多同学会说，这种单元测试真麻烦，每实现一个功能方法，都要进行测试。但这种测试驱动开发（Test-Driven Development, TDD）的方式是敏捷开发中的一项核心实践和技术，是对系统功能和代码质量重要保证。

上次实验中，同学们大多数采用在`main()`函数中找几个测试用例把对应的method跑一遍的方式进行测试。这是大家很自然想到的方法，但是使用`main()`函数测试有很多缺点：

一是，只能有一个`main()`方法，不能把测试代码分离；

二是，不能同时打印出测试结果和期望结果，例如，`expected: 3628800, but actual: 123456;`

三是，很难编写一组通用的测试代码。

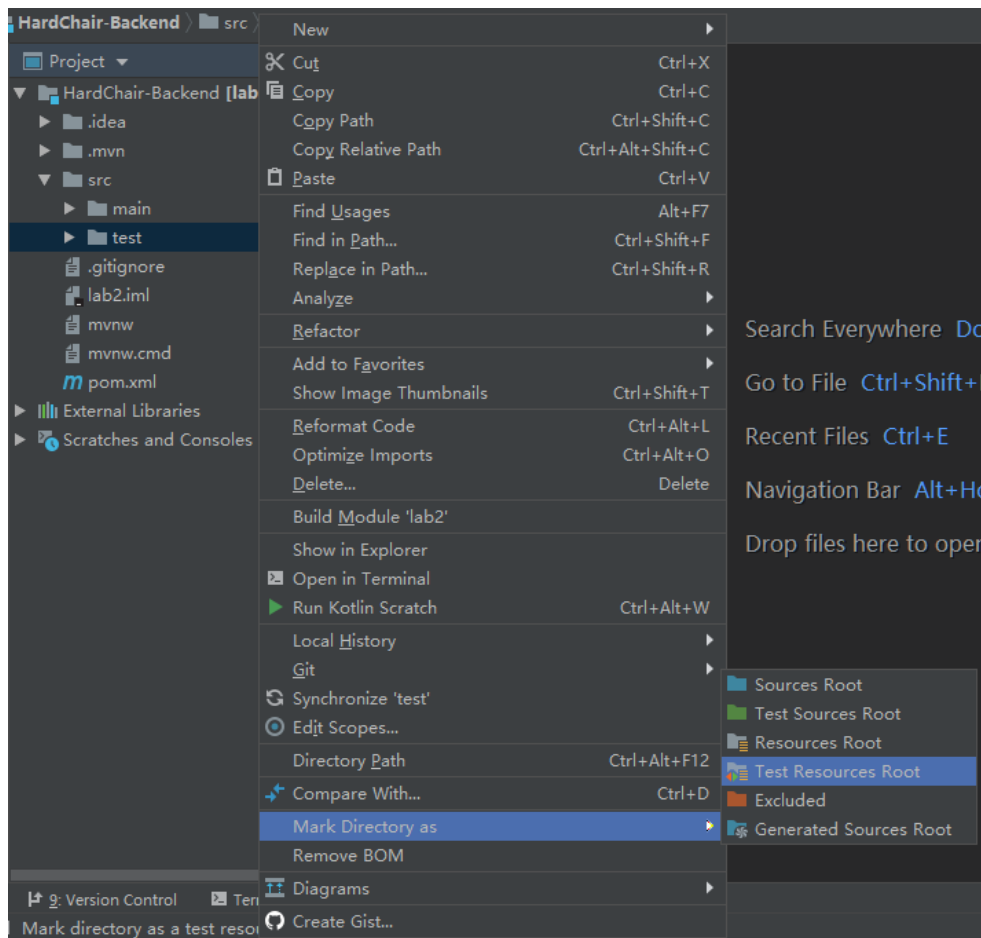
因此，我们需要一种测试框架，帮助我们编写测试。

没错，可能你们已经猜到了，就是**JUnit**。它是一个开源的Java语言的单元测试框架，专门针对Java设计，使用最广泛。JUnit是事实上的单元测试的标准框架，任何Java开发者都应当学习并使用JUnit编写单元测试。使用JUnit编写单元测试的好处在于，我们可以非常简单地组织测试代码，并随时运行它们，JUnit就会给出成功的测试和失败的测试，还可以生成测试报告，不仅包含测试的成功率，还可以统计测试的代码覆盖率，即被测试的代码本身有多少经过了测试。对于高质量的代码来说，测试覆盖率应该在80%以上。此外，几乎所有的IDE工具都集成了JUnit，这样我们就可以直接在IDE中编写并运行JUnit测试。

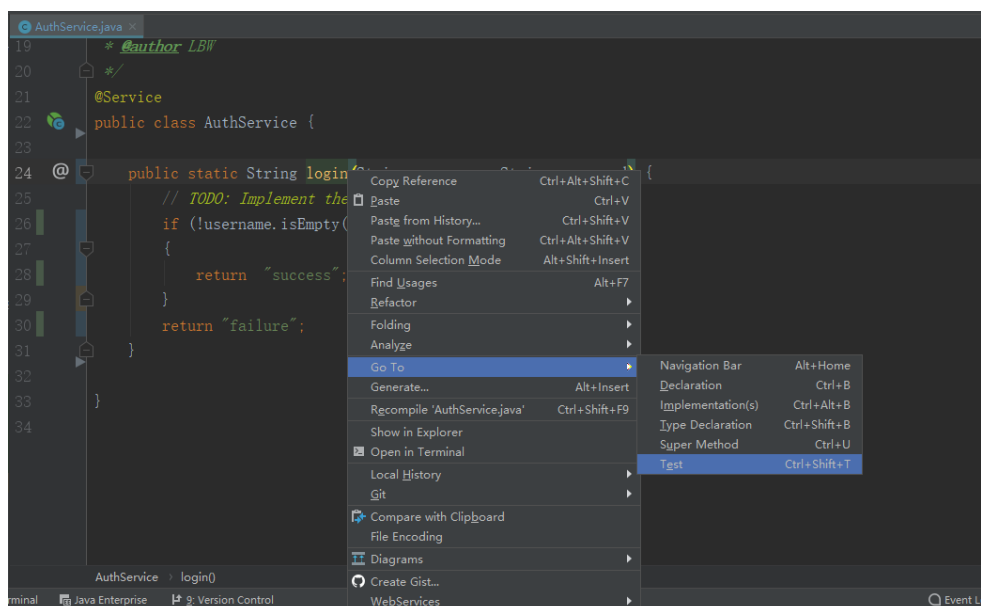
接下来，我们将教大家如何在IntelliJ IDEA中使用JUnit的基础操作。

- 首先，我们需要在src同级目录下创建一个test目录，并将其设置为

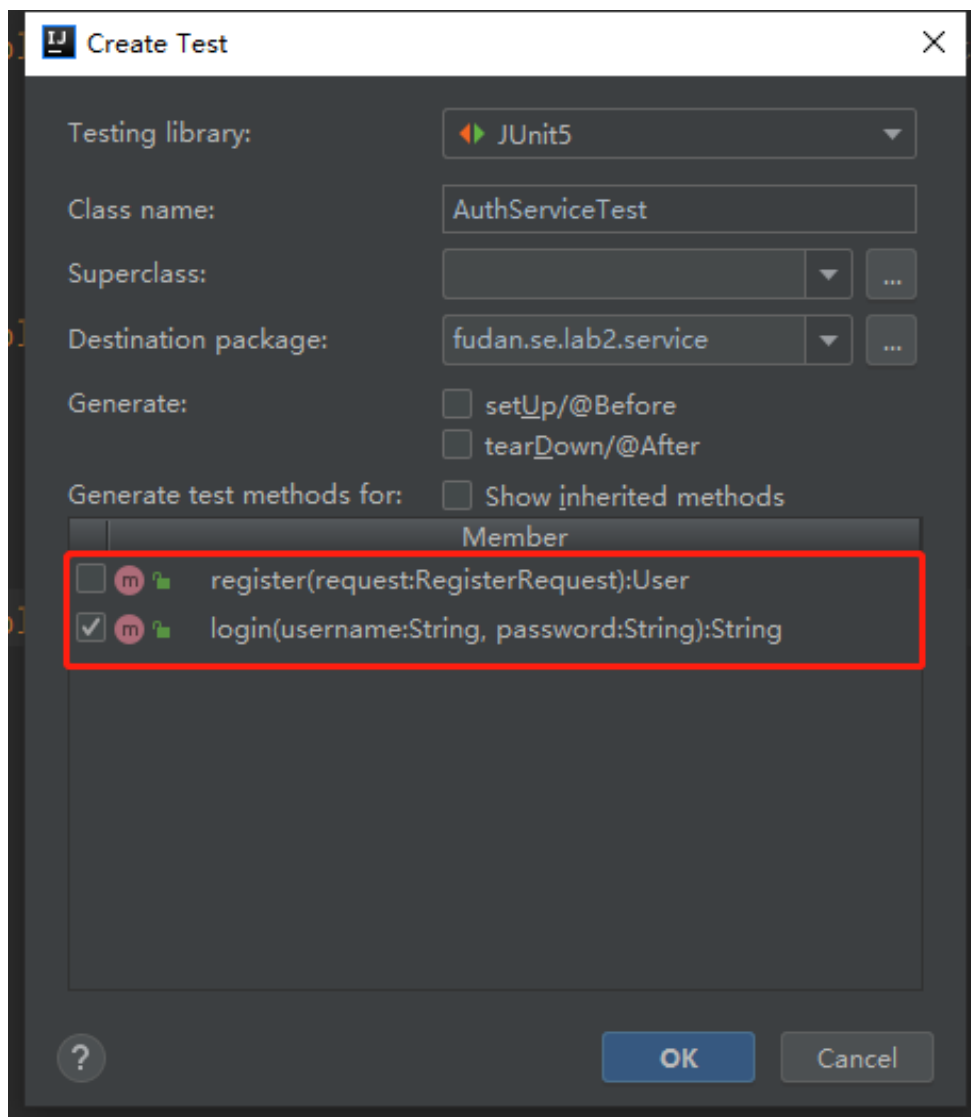
测试专用文件夹，然后在下面创建一个java目录（当然，上次实验中我们已经为大家做好了）



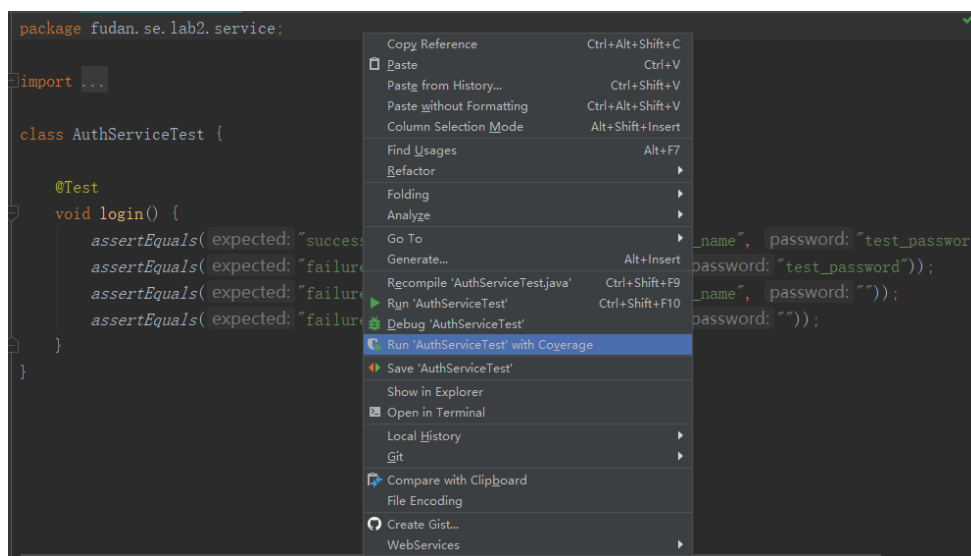
- 在你想要测试的method或class上右键创建对应的测试文件，或者使用快捷键“Ctrl+shift+T”



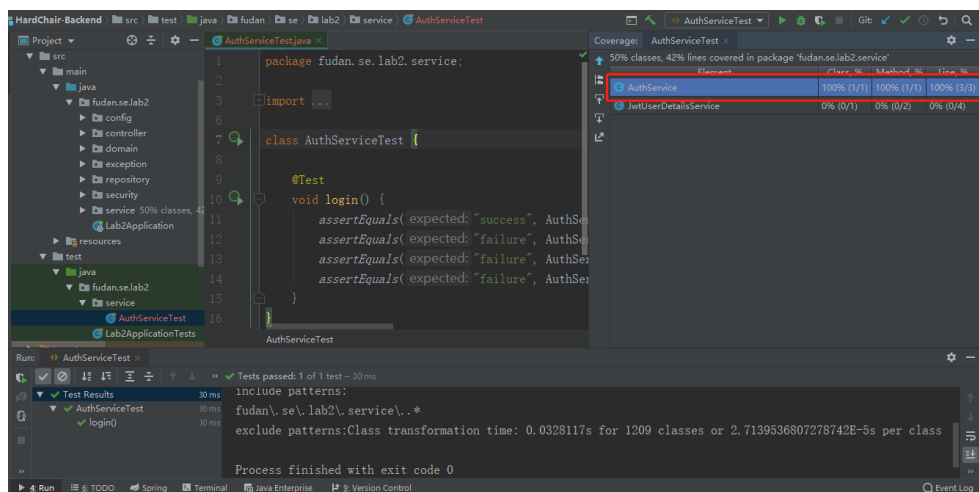
- 选择你要进行测试的method，点击“OK”创建，测试文件会自动创建在test目录中



- 在测试类中编写测试代码，右键选择‘Run with Coverage’运行。



- 最后所有测试用例通过，并且保证被测试的method或class的测试覆盖率不低于80%即可。



核心测试方法 `login()` 加上了 `@Test` 注解，这是JUnit要求的，它会把带有 `@Test` 的方法识别为测试方法。在测试方法内部，我们用 `assertEquals("success", AuthService.login("test_name", "test_password"))` 表示，期望 `AuthService.login("test_name", "test_password")` 返回 "success"。`assertEquals(expected, actual)` 是最常用的测试方法，它在 `Assertion` 类中定义。`Assertion` 还定义了其他断言方法，例如：

- `assertTrue()`：期待结果为 true
- `assertFalse()`：期待结果为 false
- `assertNotNull()`：期待结果为非 null
-

4.3 代码质量要求

上次的实验，我们已经要求大家学会使用华为云的代码检查工具，对前后端的代码进行质量检查，但并没有强制要求大家根据质量评估报告，对代码中的质量问题进行处理，修改相应的代码。

所以这次，大家继续使用代码检查工具，并根据质量评估结果，对代码问题，圈复杂度，代码重复率都需要进行修改。每个代码问题华为云都会有相应的修改意见，大家可以直接进行修改，最后要保证代码问题数为‘0’全部解决，质量门禁显示‘passed’，而圈复杂度和代码重复率尽可能低。（注：如果前端开源组件等有质量问题无法处理，可以在质量检查的时候，设置忽略该组件的相关代码文件。其他可能出现的质量问题如果合理，可在实验报告中说明情况）

我们会检查大家在解决代码质量问题时代码的修改情况。合格的代码质量检查结果如下图：

任务	质量门禁	问题	最近一次检查	操作
Lab2 frontend official	Passed	0 未解决 10 已解决	手动触发 检查于 2020-03-23 14:10:54	...
Lab2 backend	Passed	0 未解决 4 已解决	手动触发 检查于 2020-03-23 01:20:01	...

5. 实验提交

5.1 打标签

在完成本次Lab的所有内容后，首先你们需要使用Git为项目打上相应的标签。具体来说，你们需要分别选择前后端截止日期前的一次 `commit`，将其打上 `lab3-finish` 的标签，并上传至远程仓库中。

注意，我们将使用此标签所对应的 `commit` 版本进行评分。

5.2 编译构建+部署

在 Lab2 中，我们并没有要求使用华为云进行编译构建和部署，尤其是前端，因为前端的部署需要一定的Linux知识以及Nginx知识，这是你们需要自己连上服务器，去钻研去了解的知识。

在经历了 Lab2 中前端的编译构建和部署后，相信你对NPM，Nginx等知识有了更进一步的认识。其实前端的编译构建与部署也是可以在华为云上完成的，在设置好相应的步骤后，这可以大大减少你们以后的工作量。

那么这次，我们就开始要求使用华为云分别进行前后端的编译构建和部署了。这里我们要求对 `lab3-finish` 标签所对应的 `commit` 版本进行 编译构建和部署。

编译构建

华为云支持使用特定的Tag版本进行编译构建。具体来说，在新建编译配置的步骤里面，你需要：

1. 点击高级设置。
2. 选择使用 指定Tag构建，填写合适的克隆深度。

新建编译构建任务

基本信息 选择代码源 选择构建模板

* 源码源:

DevCloud GitHub 通用Git 码云 来自流水线

* 源码仓库:

HardChair-Frontend

* 分支:

master

收起高级设置

使用指定代码仓库Tag或CommitID构建:

☐ 不指定 ☒ 指定Tag构建 ☐ 指定CommitID构建

克隆深度:

10

这样，在执行编译构建的时候，你需要输入相应Tag的名称，它就会选择相应的版本，进行编译构建了。

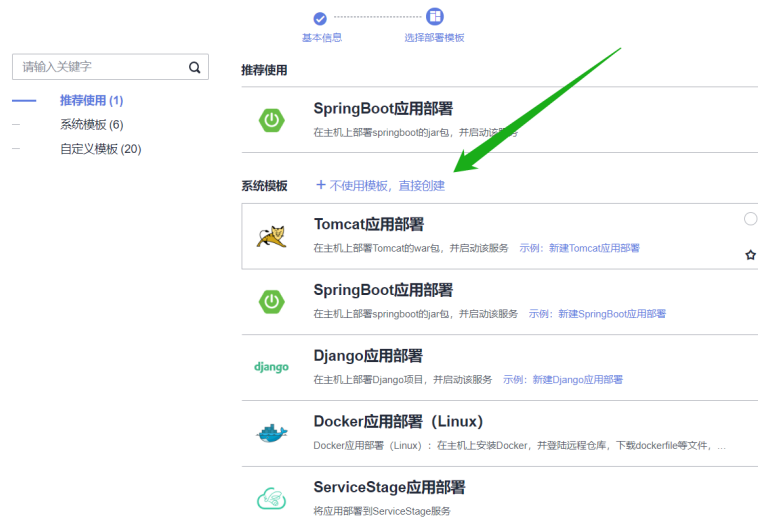
注意：在实验截止后，你仍可以任意地往仓库中提交代码，但是请保证**在80端口部署** `**lab3-finish**` 所对应的项目版本，以供助教检查。

后端的编译构建与 Lab1 无异，前端则推荐使用华为云提供的 npm 模板。需要注意的是，华为云的上传软件包步骤不支持文件夹操作，所以你需要利用 shell 命令，将构建结果进行打包后再进行上传。

具体是什么命令呢？RTFM 吧！

部署

后端的部署与 Lab1 无异。前端则需要稍作研究。你会发现，在华为云的部署中，并没有提供一个开箱即用的 Nginx 部署模板，因此你需要选择 **不使用模板，直接创建**，然后自行添加所需要的步骤，形成一个自定义的前端部署模板。



你可能会想，那我到底需要添加哪些步骤呢？原理上来说，使用华为云进行部署与手动登上服务器进行部署是没有任何区别的。仔细想一想，当你手动去服务器上部署时，经历了哪些步骤呢？想通以后，你只需要将其转换成华为云中相应的步骤，就完成了这一自动化的过程。那么以后在任何时候你想要对前端进行部署，都不需要手工登上服务器进行繁琐的操作了，仅需敲一个按钮即可。

与 Lab2 类似，你需要保证项目是可以访问的，我们会在每组服务器的80端口进行测试。

5.3 实验报告提交

除此之外，小组组长需要在Classroom中提交实验报告。

实验报告部分的部分要点如下：

- 每个小组提交一份pdf版本的实验报告，由每组的组长在Classroom中进行提交
- pdf文件的命名为：组长姓名-组长学号-Lab2实验报告
- DevCloud中对Lab3进行项目需求规划（Scrum项目）或者使用看板分配任务（看板项目）的截图
- DevCloud中对Lab3的项目代码使用代码检查的检查结果截图
- 利用Git对Lab2截止日期前的最后一次commit打tag的截图
- 利用Git给Lab3截止日期前的最后一次commit打tag的截图
- 项目各个页面的截图以及使用说明。
- 每个组员的任务分配情况
- 小组的实验过程记录，遇到的问题以及解决方案
- 每个小组成员单独的实验总结

附录

对于整个系统来说，有两种大粒度的身份划分，分别是系统的管理员和用户。

管理员	审核会议 审核结果有2种--通过/驳回
用户	

对于某个会议而言，系统中的用户可以有以下三种身份。

AUTHOR	默认所有人都可以看到已发布的会议，给未截稿的会议投稿
PC member	chair通过姓名在系统中邀请PC member，PC member也可以投稿
chair	申请会议成功后自动成为该会议chair