Lab2 初识Web开发

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3月24日 24:00

1.实验目标

- 了解团队协作开发的基本模式
- 了解Spring Boot + Vue 进行Java Web开发的基本流程
- 学习前后端分离的开发方式
- 初步体验工程化开发对代码质量和测试的要求

2. 实验内容

2.1 实验整体介绍

我们软件工程课程实验的最终目标是让同学们以小组合作的形式，完成一个基于Java Web的论文投稿系统。实验的一次次发布刚好对应着项目开发过程中的一次次迭代。让同学们在课程实验中学习到软件开发的具体流程，并且配合华为云平台，更能加深对于编译，构建，部署，测试等软件生命周期的理解。关于项目背景和论文投稿系统的具体介绍见任务书的附录。

2.2 项目创建

每组组长在华为云上新建Scrum项目，项目名称以：2020软件工程_Lab2_组长姓名_组长学号格式进行命名。

在项目中点击 更多->设置->通用设置，进行项目用户添加

同时添加FudanSoft和小组的其他成员，让他们加入到本次项目中来，同时将FudanSoft设置为项目经理。

用户名称	昵称	账号体系	企业用户
SE18302010066	--	华为云	FudanSoft
SE18302010067	--	华为云	FudanSoft
SE18302010068	--	华为云	FudanSoft
SE18302010069	--	华为云	FudanSoft
SE18302010070	--	华为云	FudanSoft
SE18302010071	--	华为云	FudanSoft
SE18302010072	--	华为云	FudanSoft
SE18302010073	--	华为云	FudanSoft
SE18302010074	--	华为云	FudanSoft
SE18302010075	--	华为云	FudanSoft

2.3 实验项目模板

为了让同学们更快入手并学习Spring Boot + Vue的前后端分离的项目开发，我们已经为大家准备了项目模板，并上传到了DevCloud的仓库中。在DevCloud中查看项目模板对应的仓库。其中前端模板仓库名称为HardChair-Frontend，后端模板仓库为HardChair-Backend。可以将模板仓库克隆到本地，在模板的基础上进行开发。当然，模板只是一个参考，同学们也可以自己搭建项目进行开发。

前端模板仓库地址：https://devcloud.huaweicloud.com/codehub/project/3e91d57e2bf3419a986b82fc8a959a87/codehub/648082/home

后端模板仓库地址：https://devcloud.huaweicloud.com/codehub/project/3e91d57e2bf3419a986b82fc8a959a87/codehub/648080/home

在将模板仓库克隆到本地后，利用IDEA导入Spring Boot的后端项目这个部分相信大家经过Lab1都已经掌握了，下面介绍一些如何在Spring Boot项目中导入我们的前端Vue模板。

对于克隆到本地的前端模板，大家同样可以选择自己的喜欢的IDE（如WebStorm）或者编辑器（如VS Code）打开。如果你想看vue项目运行起来是什么样子，可以进入到模板中的frontend文件夹下，利用指令npm install安装模板所需依赖，之后利用指令npm run dev，等待编译结束，即可通过你本地的80端口查看模板运行之后的结果，如果80端口被占用，请根据命令行的提示到对应端口查看。如果你不知道npm是什么，可以到这里了解更多：https://www.runoob.com/nodejs/nodejs-npm.html。

3 实验要求与评分细则

3.1 实验项目功能需求

- 实现用户注册功能，用户在注册过程中必须填写的信息为用户名，密码，邮箱（需要符合标准邮箱格式），所属单位（如“Fudan

University”），所在区域（如“China”）等，需要验证邮箱格式是否正确，用户名和密码是否符合规范，用户名是否与已注册的用户冲突，必填项不能为空等，如果出现注册信息填写不完整时需要将未填写项标红并在界面中给用户以提示，某项信息填写不符合要求的时候也要进行提示。

- 实现用户登录的功能，让用户输入自己的账号和密码进行登录操作，需要保证只有使用正确的账号信息才能登入系统（该账户已经在系统内进行过注册，并且账户名与密码相匹配），在用户登录失败时在登录界面给其必要的提示，使用拦截器进行登录检查，即只有在用户进行登录后才能使用系统的功能。
- 用户在系统中进行会议开设申请，需要填写会议的简称（如“ICSE 2020”）和全称（如“The 42nd International Conference on Software Engineering”）、举办时间（如“October 5-11, 2020”）、举办地点（如“ Seoul, South Korea”）、投稿截止日期、评审结果发布日期等。用户提交申请返回成功或失败的提示：如果信息不完整或有问题则提示用户进行修改，如果信息完整且没有问题则申请成果。注意，会议申请后需要管理员批准，该批准功能本次lab不要求实现。

类别	规则
账号规范	只能包含字母，数字或两种特殊字符（-_）且只能以字母或-开头
账号规范	长度为5-32个字符
密码规范	长度为6-32个字符
密码规范	字母，数字或者特殊字符（-_）至少包含两种
密码规范	不能包含账号
邮箱规范	使用雷·汤姆林森创立的标准E-mail格式，即用户标识符+ @ + 域名

3.2 实验过程要求

- 如果在DevCloud上新建项目为Scum项目，需要使用DevCloud需求规划功能进行项目开发的项目规划。如果在DevCloud上新建项目为看板项目，那么需要使用DevCloud管理看板功能进行任务工作分配。
- 每个功能实现都要有与其对应的前端页面，同时完成的系统也要和Lab1一样在每组的华为云服务器上进行部署，系统部署的具体端口号为8080。
- 使用DevCloud对于开发完成的项目进行代码检查并解决出现的代码问题。
- 在git里面，使用git config user.name 命令配置你的真实姓名的拼音，让我们可以知道你们每个人的提交记录，提交记录可以部分体现你对小组项目的贡献程度。原版本的任务书截图中命令格式存在问题，直接使用git config usr.name "xing ming"，不需要‘=’。

3.3 测试

这些功能模块的实现都有很多需要注意的要点，比如说用户注册的时候，需要检查用户信息表单填写是否完整，必填项是否都正确地填写了，新输入的账户名是否和数据库中已注册的用户名冲突？如果没有注意到的话，开发出来的系统就是存在缺陷的，无法在实际中进行应用。

那么如何来发现系统中存在的错误以及缺陷呢？那么就需要同学们对于自己开发的系统进行测试，自己来构造一些测试用例来对于系统进行测试，发现并解决系统中的错误或缺陷。

当我们对你们的Lab2实验进行评价的时候，测试用例的通过数也会影响到你们的实验成绩，所以，尽可能全面的考虑问题，并使自己构造测试用例的覆盖度尽可能高。

3.4 Lab2评分细则

评分点	分值
完成项目的部署，网站可以由公网IP的80端口访问的到	5
提交完整的实验报告	15
注册功能基本实现	15
登录功能基本实现	15
会议开设申请功能基本实现	15
完成会议开设申请页面	10
注册信息验证，比如邮箱格式，用户名是否冲突等	5
用户登录信息错误时是否给其提示，是否使用拦截器进行登录检查	5
使用DevCloud进行需求规划或者使用看板功能进行工作分配	5
使用DevCloud代码检查功能进行项目代码检查	5
使用Spring Boot框架	5

3.5 提交项目以及实验报告

在小组合作完成Lab2的Web项目之后，需要将Web项目上传到华为云的远程代码仓库中。并在DevCloud上完成构建部署，前端项目部署的端口号为80，后端项目部署的端口号为8080，保证项目是可以访问的，我们会在每个组服务器的80端口进行测试，同时需要小组组长提交实验报告。

实验报告的要点如下

- 每个小组提交一份pdf版的实验报告，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中对于Lab2进行项目需求规划或者使用看板分配任务的截图（两种截图分别对应新建项目为Scrum项目或者看板项目）
- DevCloud中对于Lab2的项目代码使用代码检查的检查结果截图
- 项目各个页面的截图
- 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

4. 附录

4.1 实验项目背景

大家知道在科研的过程中，科研的成果需要经过同行评审，来履行学术质量把关，只有经过了同行评审的科研成果才会被学术界认可。因此论文的投稿与审稿在科学研究中具有着重要的地位，论文的投稿与审稿离不开论文投稿系统，而我们这次的实验就需要大家开发一个完整的论文投稿与审稿系统。

大家可能对于论文的投稿并不熟悉，下面我以EasyChair为例，向大家介绍一下论文投稿与审稿系统。

4.1.1 系统的用户介绍以及用户权限

1. 系统中一共有四类角色，分别是会议主席（chair）,审稿人（program committee member）,管理员（administrator）和投稿人（author）。
2. 投稿与审稿网站用户包括普通用户（user）和管理员（administrator），他们从同样的入口登录，然后根据角色使用不同的功能。普通用户需要用邮箱注册获得登录账号，管理员使用预设的账号登录。普通用户包括chair，PC member和author。
3. 普通用户中三类角色区分是与会议有关的，例如一个人可以是a会议的author、b会议的chair、c会议的PC member。
4. 用户在每个学术会议中可以有三种不同角色：chair、PC member、author。其中，chair是申请开设某个会议并管理审稿过程的人，PC member是chair邀请来参加会议审稿的人，author是向会议投稿的人。

4.1.2 系统的主要功能与工作流程

1. 投稿与审稿网站上可以有多个学术会议，每个会议可以处于不同的阶段：准备中、投稿中、审稿中、终评中、审稿结束。
2. 用户申请开设某个会议并填写基本信息，管理员审批通过后自动进入准备阶段，会议申请人自动成为chair。
3. chair可以设置会议的投稿开始和结束时间，并邀请PC member加入这个会议。
4. 用户收到某个会议PC member邀请并接受后自动成为这个会议的PC member。
5. 用户可以看到目前所有开设的学术会议，如果会议处于投稿状态那么可以填写论文作者、题目、摘要等信息并提供PDF版本论文进行投稿，成为这个会议的author。
6. chair在投稿结束后将每篇论文发给一个或多个PC member评审，并在终评阶段根据评分情况决定论文录用结果（accept或者reject）。
7. 会议的author可以查看论文状态和最终结果。

上面的介绍只是为了让大家对于系统有一个初步的了解，更加详细具体的要求我们会在后续的实验中进行发布。

4.2 Spring Boot

4.2.1 Spring Boot简介

> *简化Spring应用开发的一个框架；*
>
> *整个Spring技术栈的一个大整合；*
>
> *J2EE开发的一站式解决方案；*

Spring Boot来简化Spring应用开发，约定大于配置， 去繁从简，just run就能创建一个独立的，产品级别的应用。

4.2.2 Spring Boot程序文件结构

- resources文件夹中目录结构
    - static：保存所有的静态资源； js css images；
    - templates：保存所有的模板页面；（Spring Boot默认jar包使用嵌入式的Tomcat，默认不支持JSP页面）；可以使用模板引擎（freemarker、thymeleaf）；
    - application.properties：Spring Boot应用的配置文件；可以修改一些默认设置；

4.2.3 Spring与依赖注入机制

对于 Spring 程序, Spring 框架为我们提供一个 IoC 容器, 该容器负责创建对象和维护对象之间的依赖关系。而对于普通程序, 我们是通过对象本身来创建和解决自己的依赖问题。

ApplicationContext 即是 Spring 程序的 IoC 容器, 该容器负责创建 Bean, 并将功能类 Bean 注入到你需要的 Bean 中。有三种方式告诉Spring程序我们有哪些Bean类以及这些类之间的依赖关系是什么：

- XML配置方式
- 注解配置方式
- Java配置方式

4.2.4 Spring Boot的配置文件

Spring Boot使用一个全局的配置文件，配置文件名是固定的：

application.properties或application.yml

@Value获取值和@ConfigurationProperties获取值比较

- 如果说，我们只是在某个业务逻辑中需要获取一下配置文件中的某项值，使用@Value；
- 如果说，我们专门编写了一个javaBean来和配置文件进行映射，我们就直接使用@ConfigurationProperties；

@PropertySource&@ImportResource&@Bean

- **@PropertySource**：加载指定的配置文件；
- **@ImportResource**：导入Spring的配置文件，让配置文件里面的内容生效；
- 使用**@Bean**给容器中添加组件

```
@Configuration
public class MyAppConfig {

//将方法的返回值添加到容器中；容器中这个组件默认的id就是方法名
@Bean
public HelloService helloService(){
    System.out.println("配置类@Bean给容器中添加组件了...");
    return new HelloService();
   }
}
```

参考学习资料

Spring Boot官方网址

4.3 Vue.js

Vue是一套用于构建用户界面的渐进式JavaScript框架，它让你通过简单而灵活的 API 创建由数据驱动的 UI 组件。与类似React相比，更轻量级、更容易上手。通过Vue中的“单文件组件”特性，更灵活的定义组件，不仅使代码结构更清晰，而且能与任何其他组件进行随意组合，更具复用性。

为什么要使用前端框架？

在过去十年内，因为JavaScript我们的网页变得更动态化和更加强大，我们将很多原来的服务器端代码放到浏览器中，这样就产生了成千上万行连接HTML和CSS的JavaScript，但是其缺乏正规的组织形式。所以我们需要前端框架来将他们合理并有效的组织起来。

参考学习资料

Vue官方文档

4.4 Web架构与前后端分离

4.4.1 Web开发的简要历程

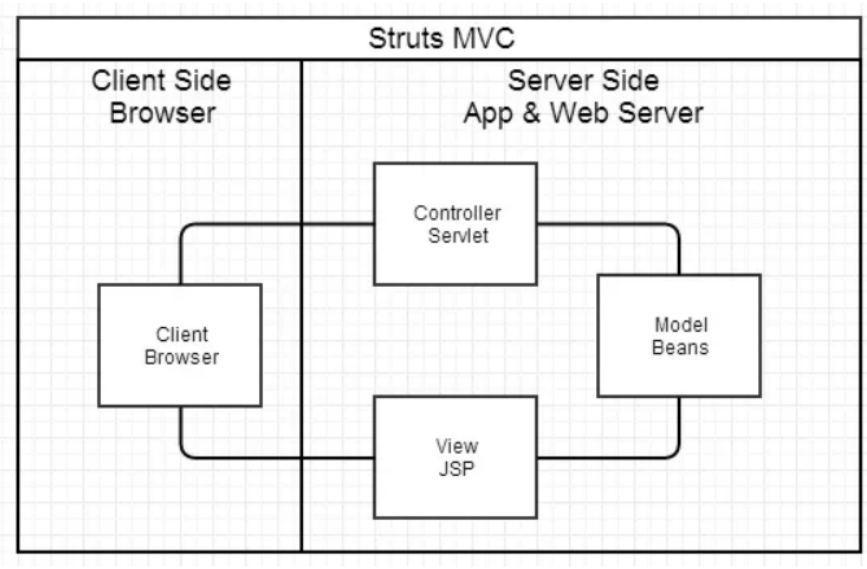
- 第一阶段: HTML + CSS + JS，直接通过原生的javascript操作Dom树;
- 第二阶段: JQuery诞生，配合前端MVC为代表的Backbone.js, 让我们可以优雅而简单的操作Dom树;
- 第三阶段: 后端架构升级为MVC，前后端分工更清晰，前端工程化，ECMAScript规范开始崭露头角;
- 第四阶段: 后端架构进入了微服务时代，前端架构不仅升级为MVVM，ES6更是成为目前事实上的标准;

4.4.2 MVC架构

在介绍MVVM架构之前，我们需要先了解一下MVC架构，MVC即Model-View-Controller

- View：进行数据显示。
- Model：用于封装与应用程序的业务逻辑相关的数据以及对数据的处理方法。
- Controller：处理用户交互，负责转发请求，并对请求进行处理（向模型请求数据或发送数据）

其架构如下图所示：

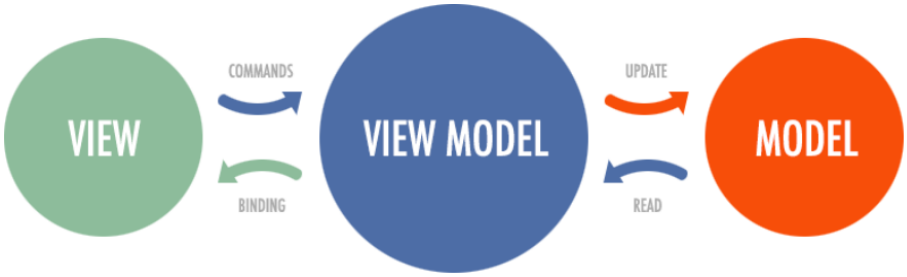


MVC相对于最早期的Web架构来说，代码维护性较好，但是随着不同终端的出现，前端的工作量变大，前后端职责纠缠不清。

4.4.3 MVVM

在MVVM中，使用View-Model来代替MVC中的Controller

- Model
- View
- View-Model：简化的 Controller，为 View 提供处理好的数据，不含其他逻辑。



MVVM很好的解决了前端开发中遇到的三个问题

- 开发者在开发过程中需要调用大量的DOM API，造成了很严重的代码冗余
- 大量的DOM操作使得页面的渲染性能降低，加载速度变慢并且影响用户体验
- 当Model频繁发生变化的时候，开发者需要主动更新到View，操作繁琐并且很难维护复杂多变的状态

我们前面所讲到的Vue.js就是一种基于MVVM的前端设计框架

4.4.4 前后端分离

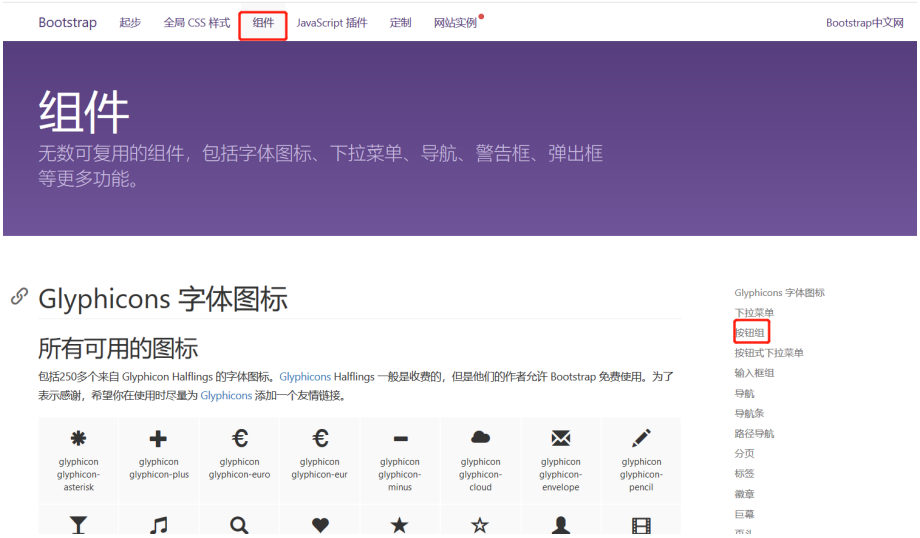
- Web服务器不再负责业务的处理，而且是将收到的请求发送给相关的后端服务器，而后端服务器将处理过后的业务数据填入HTML模板，并发送给浏览器。
- 前后端之间可以自由选取通信手段，AJAX或者RPC等。
- 对于我们的实验设置来说，后端采用Spring Boot框架，前端采用基于MVVM架构的Vue框架，使得前后端分离开发成为可能。
- 只要接口约定好之后，前端工程师不需要关心业务处理的具体实现，后端工程师也不需要关心前端页面的设计与布局，他们只需要做好自己的职责，这也使得前后端职责纠缠问题得到了很好的解决。

4.5 使用Bootstrap实现快速的前端开发

Bootstrap 让前端开发更快速、简单，所有开发者都能快速上手。前端新手也可以很快学会Bootstrap， 并利用其开发出简洁易用的前端网页，而不需要编写大量的HTML和CSS代码。

在我们按照官方教程安装好Bootstrap之后，如果我们想在页面中实现一个按钮组的UI组件的时候，我们只需要访问Bootstrap的官方文档，在导航栏的组件部分找到按钮组，将其给出的代码加入到我们的项目中，一个美观的按钮组就添加到我们的页面中啦。

当然，Bootstrap还可以实现更多令人惊艳的功能，例如响应式布局和移动端适配，更多的功能就由你来探索啦！



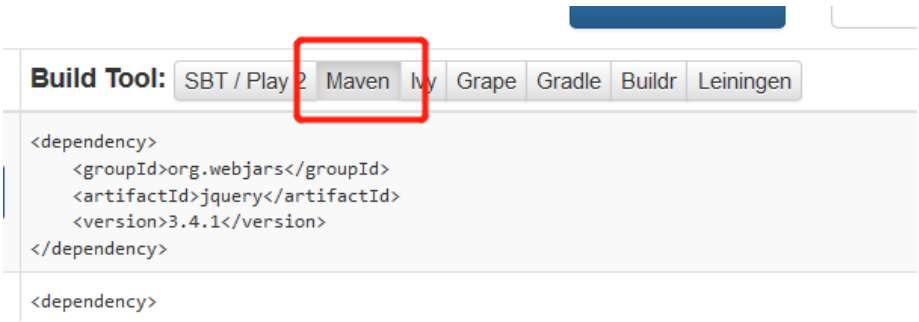
参考学习资料

Bootstrap官网

4.6 Webjars

使用Webjars进行前端资源管理。Webjars是将这些通用的Web前端资源打包成Java的Jar包，然后借助Maven工具对其管理，保证这些Web资源版本唯一性，升级也比较容易。我们可以到webjars官网上找到自己需要的资源，在自己的工程中添加入maven依赖，即可直接使用这些资源了。

例如我们如果要在项目中使用Bootstrap的时候，我们先登录https://www.webjars.org/，并在Build Tool里面选择Maven



在Popular WebJars里面找到Bootstrap，并选择我们需要的版本，网站就会自动为我们生成Bootstrap资源的Maven依赖。

静态资源引入

```
<link rel='stylesheet'
href='webjars/bootstrap/3.3.6/css/bootstrap.min.css'>

<script type='text/javascript'
src='webjars/bootstrap/3.3.6/js/bootstrap.min.js'>
</script>
```

参考学习资料

webjars官方网站

4.7 使用REST

推荐使用REST的定位资源的URL风格和REST的资源操作规则

例如

http://114.116.XXX.XXX:8080/login 用来登录

http://114.116.XXX.XXX:8080/register 用来注册