

Lab2初识Web开发

1. 实验目标

2. 实验内容

2.1. 实验整体介绍

2.2 实验项目模板

3. 实验要求与评分细则

3.1. 实验项目功能需求

3.2. 实验过程要求

3.3. 测试

3.4. 提交项目以及实验报告

3.5. Lab2评分细则

4. 实验整体介绍

4.1. 项目背景

4.2. 系统的用户介绍以及用户权限

4.3. 系统的主要功能与工作流程

4.3.1. 图书管理

4.3.2. 用户管理

4.3.3. 用户借阅流程

5. 技术点介绍

5.1. SpringBoot

5.1.1 Spring Boot简介

5.1.2 Spring与依赖注入机制

5.1.3 SpringBoot的配置文件

5.1.4 一些其他有用的网址

5.2. 数据库

5.3. RESTful

5.4. Vue.js

5.5. Web架构与前后端分离

Lab2发布日期：2021年3月10日

Lab2截止日期：2021年3月28日23:59

注：本次 Lab为小组实验。

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的图书馆管理系统。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，更能加深对于需求、设计、实现、编译、构建、测试、部署等软件生命周期的理解。关于项目背景和图书馆管理系统的具体介绍见 4 实验整体介绍。

2.2 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中可以查看项目模板对应的仓库。其中前端模板仓库名称为HardLibrary-Frontend，后端模板仓库为HardLibrary-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.cn-north-4.huaweicloud.com/codehub/project/89edbcbf9b9a4b22bf48f61dcdd2b466/codehub/814935/home>

后端模板仓库地址：<https://devcloud.cn-north-4.huaweicloud.com/codehub/project/89edbcbf9b9a4b22bf48f61dcdd2b466/codehub/814917/home>

在将模板仓库克隆到本地后，大家可以自行学习如何利用IDEA导入Spring Boot的后端项目，下面介绍一下如何在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2048端口查看模板运行之后的结果，如果2048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账号、密码、邮箱（需要符合标准邮箱格式），需要验证邮箱格式是否正确，账号和密码是否符合规范，账号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。
- 用户在系统中上传新的书籍（具体图书的属性请看 4.3.1. 图书属性），用户上传后会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上传成功。用户可以在系统中看到所有已添加的书籍。
- 注意，理论上图书只有管理员可以进行上传，普通用户和游客不能上传，只能搜索查阅。本次lab不要求实现根据角色权限进行功能的限制与区分。

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6-32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需求工作规划功能进行项目开发的项目规划。
- 每个功能实现都要有与其对应的前端页面。
- 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 --global 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。如 `git config --global user.name xiaoming.`
- 在git中加入针对springboot的 `.gitignore` 文件。参考 <https://github.com/spring-projects/spring-boot/blob/master/.gitignore> 。避免仓库快速膨胀。

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账号冲突？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。

当你们对你们的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. 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。项目需要在本地可以运行，并有配套的前端页面。之后需要小组提交实验报告。

实验报告的要点如下

- 每个小组提交一份pdf版的实验报告，由每组的组长在华为云Devcloud的小组项目的文档中进行提交。



- pdf文件的命名为：小组号-Lab2实验报告
- DevCloud中对于Lab2进行项目需求规划分配任务的截图（截图对应新建项目为Scrum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

3.5. Lab2评分细则

评分点	分值
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
图书上新功能基本实现	15
完成图书展示搜索展示页面	15
注册信息验证，比如邮箱格式，账号是否冲突等	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

4. 实验整体介绍

4.1. 项目背景

书籍是人类宝贵的财富，也是人类进步的阶梯。图书馆是一个为大家免费提供书籍借阅的地方，但书籍种类之多，借阅读者数量之多，使得人工管理图书馆在这个信息爆炸的时代变得特别困难，因此就需要图书馆管理系统的辅助，管理书籍的馆藏状态、读者的借阅状态。我们本次的实验就需要大家开发一个完整的面对在校师生的图书馆管理系统。

相信大家对图书的借阅流程相当熟悉了，让我们一起回忆一下在复旦图书馆借阅的流程，复旦一共有四个图书馆，教师和学生可以跨校区预约借阅图书等，这个过程中涉及的环节就包含着我们的图书馆管理系统需要实现的功能。

4.2. 系统的用户介绍以及用户权限

1. 系统中一共有三大类角色，分别是图书管理员（librarian）、用户（user）和游客（visitor）。
2. 图书管理员对图书的管理和用户的借阅有着最高权限，包括对图书的数量、编号、类型、名称等进行添加、修改、删除管理。同时，图书管理员可对用户信息进行添加、修改和删除，并对其借阅、归还图书等相关功能拥有执行权限，还可以添加删除其他的管理员。
3. 用户包括教师（teacher）、学生（student）。教师和学生从同样的入口登录，然后根据角色有不同的权限，都可以查询、借阅、续借图书，并对图书进行收藏、评论、打分等操作。此外，教师对图书的预约较学生具有更高的优先级。教师和学生在使用学校邮箱注册后，需要等待图书管理员通过注册申请，依据注册邮箱发送通过或被拒绝的通知。
4. 图书管理员有一个系统预设的账号叫做admin，用于管理图书和用户，还可以添加删除其他的图书管理员，管理员工作账号和个人用户借阅账号功能需要分离。
5. 游客不需要登陆，只可以查询浏览图书相关信息，其他任何操作都不允许。

4.3. 系统的主要功能与工作流程

4.3.1. 图书管理

- **图书属性：**名称，作者，简介，ISBN，出版时间，封面图片，本书馆藏有几本、每本图书馆藏在哪里（一共有四个图书馆：邯郸，张江，江湾，枫林）。每本书可以有多个副本，每个副本都应该有唯一标识；每本书也应该有唯一编号。
- 图书管理员负责对新入库书籍的信息录入，若该书籍是某些用户申请借阅的，会有邮件通知对应用户。
- 图书的数量，编号、类型、名称都由图书管理员负责修改和删除管理。
- 图书管理员对图书的借阅状态进行修改，包括可借阅（available）、已被借阅（borrowed）、已预约（reserved）和已损坏（damaged）。
- 图书管理员可以对图书所在校区进行编辑。

4.3.2. 用户管理

- 用户具有借阅书籍的功能，同一时段内最大借阅量为五本图书。
- 用户可申请跨校区借阅图书，由图书管理员转移书籍后，邮件通知该用户取书借阅。
- 用户可预约已被借阅的图书，借书者则不再具备续借的功能，且教师用户的预约优先级高于学生，同一类型的用户按照预约的先后顺序进行借阅，当可借阅时将收到邮件通知。
- 用户在最后还书期限内，可在线最多两次续借该书，若有他人预约该书，则无法续借。
- 用户对于未查询到的书籍或者已损坏无法借阅的书籍，可提出图书馆购买该书的申请，申请成功后将收到邮件通知。
- 每位用户都有借书信用分，初始状态为满分100分，预约后未在规定时间内借阅、逾期未归还、损坏图书等都将扣除借书信用分。当借书信用分低于80分时，无法再进行书籍的借阅，需要向图书管理员支付对应的金额，补偿缺失的借书信用分。

4.3.3. 用户借阅流程

1. 用户登入账号，搜索所需图书。若图书状态显示可借阅，则用户可直接去图书馆进行借阅，若跨校区则点击跨校区借阅，等书籍到达本校区后将收到邮件通知；若显示已被借阅，则需选择预约功能等待邮件通知后，方可在指定时间后取书借阅；若显示已损坏或无法找到该图书，用户可选择申请图书馆购买该书，等书籍到货后将收到邮件通知。
2. 用户申请跨校区调书、预约图书、申请购买图书功能，在收到邮件后三日内不去借阅，将取消该书的借阅资格，并降低借书信用。
3. 用户若在最后还书期限内未完成阅读，可在线进行图书续借，每本书最多两次续借机会。
4. 用户需在还书处还书，图书管理员将检查书籍是否被损坏并修改图书状态，若有损坏将降低用户借书信用。
5. 用户逾期未还图书，则降低借书信用，且在信用未提升前，无法继续借阅其他书籍。

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发布。

5. 技术点介绍

5.1. SpringBoot

5.1.1 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。

5.1.2 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

5.1.3 SpringBoot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@CongurationProperties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@CongurationProperties；

@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

5.1.4 一些其他有用的网址

Spring官网：<https://spring.io/projects/spring-boot/>

相关博客1：<http://tengj.top/2017/02/26/springboot1/>

相关博客2：<https://www.cnblogs.com/ityouknow/p/5662753.html>

maven 入门教程：<https://cloud.tencent.com/developer/article/1381622>

maven环境配置：<https://www.runoob.com/maven/maven-setup.html>

maven基础知识：<https://www.imoooc.com/wiki/mavenlesson/mavenintroduction.html>

5.2. 数据库

推荐使用MySQL数据库。

5.3. RESTful

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

什么是REST？

REST是设计风格而不是标准。REST通常基于HTTP、URI、XML以及HTML这些现有的广泛流行的协议和标准。

- 资源是由URI来指定。
- 对资源的操作包括获取、创建、修改和删除，这些操作正好对应HTTP协议提供的GET、POST、PUT和DELETE方法。
- 通过操作资源的表现形式来操作资源。

- 资源的表现形式则是XML或者HTML，取决于读者是机器还是人、是消费Web服务的客户软件还是Web浏览器。当然也可以的任何其他的格式，例如JSON。

什么是RESTful API?

符合REST风格的Web API称为**RESTful API**。它从以下三个方面资源进行定义：

- 直观简短的资源地址：URI，比如：`http://example.com/resources`。
- 传输的资源：Web服务接受与返回的互联网媒体类型，比如：JSON，XML，YAML等。
- 对资源的操作：Web服务在该资源上所支持的一系列请求方法（比如：POST，GET，PUT或删除DELETE）。

HTTP请求方法在RESTful API中的典型应用

资源	GET	PUT	POST	DELETE
一组资源的URI， 比如 <code>https://example.com/resources</code>	列出URI，以及该资源组中每个资源的详细信息（后者可选）。	使用给定的一组资源替换当前整组资源。	在本组资源中创建/追加一个新的资源。该操作往往返回新资源的URL。	删除整组资源。
单个资源的URI， 比如 <code>https://example.com/resources/142</code>	获取指定的资源的详细信息，格式可以自选一个合适的网络媒体类型（比如：XML、JSON等）	替换/创建指定的资源。并将其追加到相应的资源组中。	把指定的资源当做一个资源组，并在其下创建/追加一个新的元素，使其隶属于当前资源。	删除指定的元素。

RESTful API举例

例如，一个简单的网络商店应用：

1. 列举所有商品：

```
1 GET http://www.store.com/products
```

2. 呈现某一件商品：

```
1 GET http://www.store.com/products/12345
```

3. 下单购买（创建订单）：

```
1 POST http://www.store.com/orders
2 <purchase-order>
3   <item> ... </item>
4 </purchase-order>
```

参考学习资料

1. 如何开发RESTful风格的API? : [RESTful API 设计指南](#)
2. RESTful是什么意思? : [理解RESTful架构](#)
3. REST维基百科: [Representational state transfer, From Wikipedia](#)

5.4. 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

Why Vue.js? : [Why Vue](#)

Vue官方学习文档: [Vue官方文档](#)

5.5. Web架构与前后端分离

Web开发简要历程

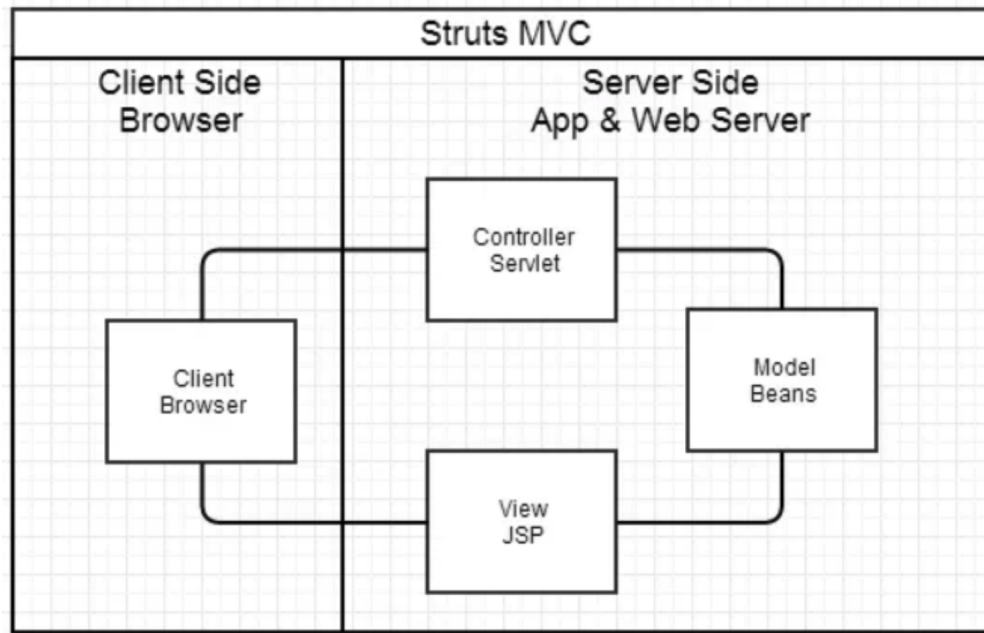
- 第一阶段：HTML + CSS + JS，直接通过原生的javascript操作Dom树；
- 第二阶段：JQuery诞生，配合前端MVC为代表的Backbone.js，让我们可以优雅而简单的操作Dom树；
- 第三阶段：后端架构升级为MVC，前后端分工更清晰，前端工程化，ECMAScript规范开始崭露头角；
- 第四阶段：后端架构进入了微服务时代，前端架构不仅升级为MVVM，ES6更是成为目前事实上的标准。

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。

MVVM架构

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

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。

参考学习资料

如何理解MVC与MVVM：[从项目工程化的角度，谈一下 MVC 与 MVVM](#)