|  |
| --- |
|  |
| Web应用技术实验报告 |
| （软件工程&服务科学2021级适用） |
|  |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 班号 | 2111102 | 学号 | 2021211919 | 姓名 | 吕文宣 |
| 开始 | 2023.4.10 | 截止 | 2023.4.28。从4月29日开始逐班逐人检查 | | |

|  |  |  |  |
| --- | --- | --- | --- |
| No | 内容 | 完成情况 | 评价等级 |
| 1 | 开发环境搭建 | ☑完成 □基本完成 □未完成 |  |
| 2 | 用户注册，登录，退出 | ☑完成 □基本完成 □未完成 |
| 3 | 优化Servlet | ☑完成 □基本完成 □未完成 |
| 4 | 分类数据展示 | ☑完成 □基本完成 □未完成 |
| 5 | 旅游线路的分页展示 | ☑完成 □基本完成 □未完成 |

目录

1. 实验内容和要求 1

1.1. 大作业和实验的内容划分 1

1.2. 报告提交要求 1

1.3. 检查与评分 1

1.4. 资料 1

1.5. 技术选型 2

2. 任务1：Web Application开发环境搭建 3

2.1. Maven 安装及配置 3

2.2. 在IDEA中配置Maven 3

2.3. 在IDEA中建立Web application工程 4

2.4. 数据库及数据表构建 5

2.5. 修改druid.properties 6

3. 任务2：用户注册，登录，退出 7

3.1. 用户注册实现 7

3.2. 登录 8

3.3. index页面中用户姓名的提示信息功能 9

3.4. 退出 10

3.5. 成果展示 10

4. 任务3：优化Servlet 11

4.1. BaseServlet编写 11

4.2. UserServlet改写 11

4.3. 页面路径改写 11

5. 任务4：分类数据展示 13

5.1. 分析 13

5.2. 阅读前台代码 13

5.3. 后台代码CategoryDao实现 13

5.4. 后台代码CategoryDaoImpl实现 13

5.5. 结果展示 13

6. 任务5：旅游线路的分页展示 14

6.1. 根据id查询不同类别的旅游线路数据分析 14

6.2. 阅读前端代码 15

6.3. 后端-阅读PageBean类，该类用来记录每一页 15

6.4. 后端创建RouteServlet 15

6.5. 后端创建RouteService 15

6.6. 编写RouteDao接口及其实现RouteDaoImpl 15

6.7. 展示分页效果 15

# 实验内容和要求

为提高应用《Web应用技术》前后端相关知识的能力，特安排本综合实验（含大作业），简称综合实验。

## 大作业和实验的内容划分

《Web应用技术》是一门实际操作课，只有实践才能掌握内容。考虑到Web工程的工作量较大，本综合实验分解为5个任务，具体如下：

|  |  |  |  |
| --- | --- | --- | --- |
| No | 内容 | 工时估算 | 建议完成时间 |
| 1 | 开发环境搭建 | 2 | 课余 |
| 2 | 用户注册，登录，退出 | 3 | 课余 |
| 3 | 优化Servlet | 2 | 课余 |
| 4 | 分类数据展示 | 2 | 实验课 |
| 5 | 旅游线路的分页展示 | 3 | 实验课 |

## 报告提交要求

每位同学需要提交纸质版和电子版实验报告：

（1）纸质版报告：在教师检查作业时携带，并提交给教师。

（2）电子版报告：为方便教师在教务系统登记成绩，电子版报告的命名格式要求按照：“班号\_学号\_姓名.docx”的形式，例如：**2111101\_201110139\_张三\_实验报告.docx**。各位同学在报告截止时间之前，将电子版报告提交给本班学委。

学委收到各位各位同学的报告时，检查格式是否按要求命名。如果提交格式错误，提醒提交有错误的同学修改。检查完毕后，将本班提交的电子版报告以班号命名一个文件夹，打包后发给老师即可。

## 检查与评分

教师会逐班逐人检查任务完成情况。需要强调的是，纸质报告和可运行的工程代码二者缺一不可。根据教务处《教师教学督导条例》及教学评估相关要求，无纸质报告，教师给出分数属于违规；而没有工程代码，报告的书写就毫无根据。因此教师需要对实验报告和工程代码综合评价，最后根据5项任务完成的质量给出评分等级或成绩。

## 资料

为降低开发难度，本综合实验已提供相关资料，具体如下：

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 说明 |
| 01 | 实验报告 | 对任务进行说明，并提出实验内容和要求 |
| 02 | 静态页面 | 实验必要的前端页面 |
| 03 | 配置文件 | 数据库datasource配置文件 |
| 04 | 工具类 | JDBCutils，UuidUtils |
| 05 | 数据库脚本 | travel.sql |
| 06 | 实体类 | 系统需要的实体类，与数据库的表对应，用于后端数据传输 |
| 07 | 其他常用类 | 编码过滤器，消除乱码 |
| 08 | Pom文件 | Maven项目管理的配置文件 |
| 09 | 部分代码 | 降低难度，给出部分代码 |
| 10 | 参考视频 | 编写有困难的同学，可以参考往年任务的视频教程 |

## 技术选型

所有任务均基于同一个 Web Application 工程。该工程采用三层架构，DBMS采用MySQL。各层的参考技术选型如下：

|  |  |  |
| --- | --- | --- |
| Web层（web包） | BBL层（Service包） | 数据访问层（dao包） |
| HTML：视图结构 | Java类与接口 | Druid：数据库连接池 |
| CSS：视图样式 | JdbcTemplate：jdbc工具 |
| JS：视图动作 | JDBCUtils：工具类 |
| JQuery：js的封装 |  |
| Ajax：异步传输 |  |
| Servlet：前端控制器 |  |
| BeanUtils：数据封装 |  |
| Jackson：json序列化工具 |  |

# 任务1：Web Application开发环境搭建

本实验所用基础环境清单如下：

|  |  |  |
| --- | --- | --- |
| 序号 | 内容 | 备注 |
| 01 | JDK 1.8及以上 | 需要配置JAVA\_HOME环境变量，并设置path |
| 02 | IDEA | 2022.1或者2021.1版本均可。但不建议用最新版本 |
| 03 | Tomcat | 7.0版本及以上均可 |
| 04 | Maven 3.6.2及以上 | 项目管理及自动构建工具 |
| 05 | Mysql 5.7及以上 | 注意驱动程序与DBMS匹配，否则易出现各种奇妙bug |
| 06 | Navicat 15.0 | Mysql的图形化管理工具。 |
| 07 | Word 2013及以上 | 不建议使用Wps编写。打印前建议转为pdf再打印 |

## Maven 安装及配置

根据课程内容知识，自行下载项目构建工具Maven。要求在Maven的配置文件中添加阿里镜像库，并建立以自己名字拼音全称命名的本地代码库。完成后请在如下的表格中，粘贴以上配置的截图。

|  |
| --- |
|  |

## 在IDEA中配置Maven

在IDEA中配置Maven（不要用IDEA默认的Maven）。请在如下表格中，粘贴IDEA中Maven的配置信息截图。

|  |
| --- |
| IMG_256 |

## 在IDEA中建立Web application工程

建立以**你的名字汉语拼音**为项目名的Web application工程。并按照如下规则建立包。

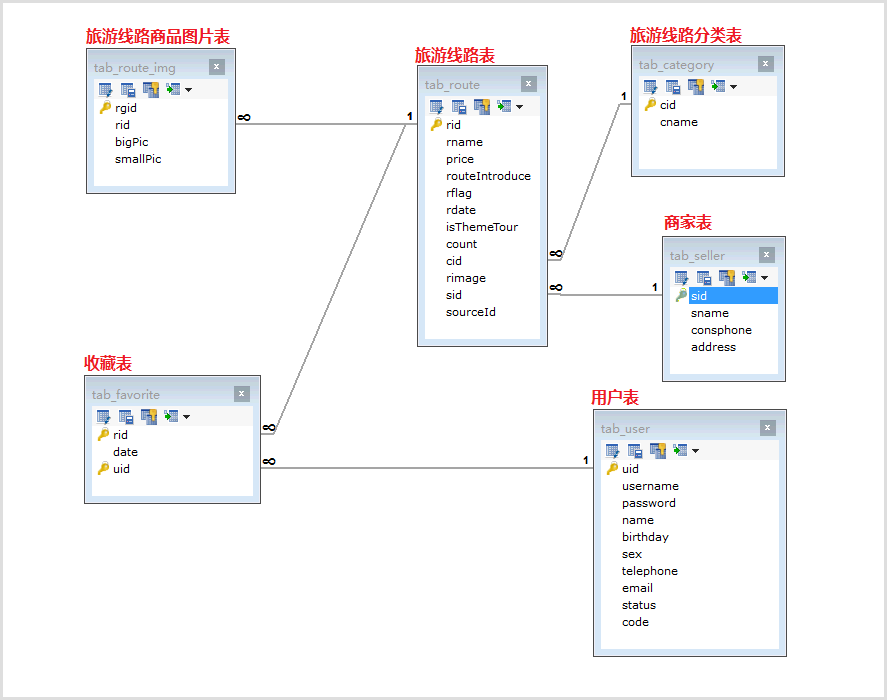
|  |  |  |
| --- | --- | --- |
| 相关层的Java包命名规则 | 对应的层 | 作用 |
| edu.hit.项目名.dao | 数据访问层 | 存放数据访问层接口 |
| edu.hit.项目名.dao.impl | 存放数据访问层实现类 |
| edu.hit.项目名.service | 业务逻辑层 | 存放业务逻辑层接口 |
| edu.hit.项目名.service.impl | 存放业务逻辑层接口实现类 |
| edu.hit.项目名.web | 表示层 | 存放web层接口类 |
| edu.hit.项目名.util | -- | 存放相关工具类 |
| edu.hit.项目名.domain | -- | 存放实体类 |

完成包的创建后，根据本实验提供的资料，拷贝代码到IDEA中Web application工程的对应位置。在IDEA中，将工程目录中代表三层架构的目录或者包展开，截图如下：

|  |
| --- |
|  |

## 数据库及数据表构建

本项目的实体关系图（即ER图）如下：



请自行安装 MySQL 及 Navicat ，以**你的姓名的汉语拼音命名**数据库，并在数据库中运行相关资料中的 travel.sql 语句，以便建立上述ER图对应的数据表。在 Navicat 中左侧菜单中，展开数据库，体现出新建立的数据表，并截图如下：

|  |
| --- |
|  |

## 修改druid.properties

根据以上数据库的设置，修改druid.properties 配置文件，并截图如下：

|  |
| --- |
|  |

# 任务2：用户注册，登录，退出

该任务是系统的基本功能。具体可以划分为三个子任务，即：用户的注册，登录和退出。其中注册和登录在课程进行过程中，已经训练多次，可以借鉴课程中和代码完成。

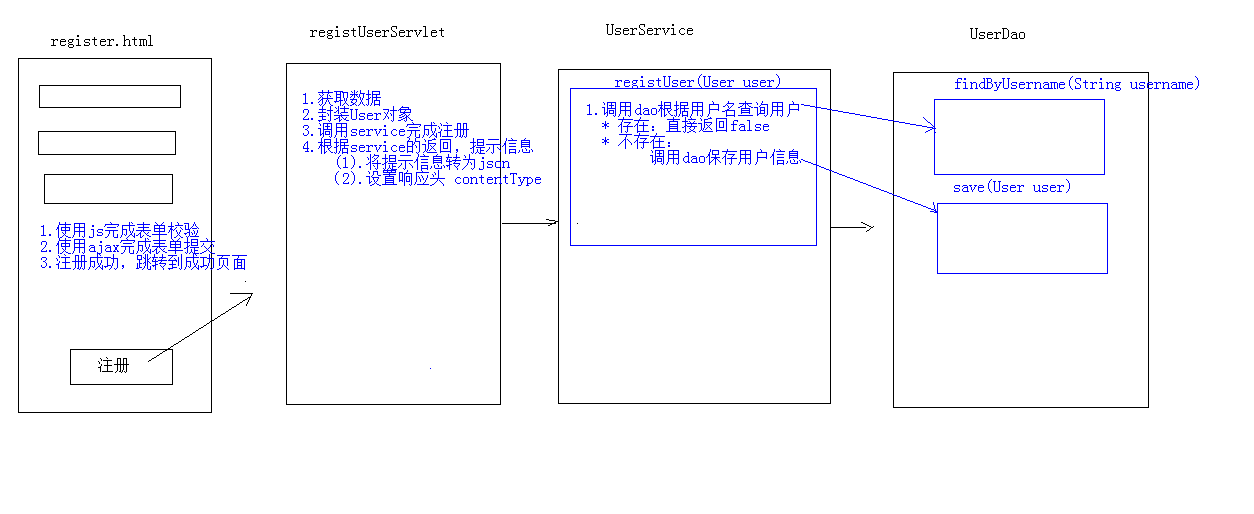
## 用户注册实现

用户注册的页面效果如下。其中

* 用户名：单词字符，长度6-12位
* 密码：单词字符，长度6-12位
* Email：邮件格式
* 姓名：非空（不做要求）
* 手机号：手机号格式
* 出生日期：非空（不做要求）



### Step1：功能的实现分析



### Step2：阅读前台异步 (ajax) 提交表单

校验完毕后，使用异步提交表单进行提交。采用异步表单提交是为了获取服务器响应的数据。因为我们前台使用的是 html 作为视图层，不能够直接从servlet相关的域对象获取值，只能通过ajax获取响应数据



### Step4：编写后端RegistUserServlet

为降低难度，在《资料09-部分代码》中已经给出该程序。

### Step5：编写后端UserService以及UserServiceImpl

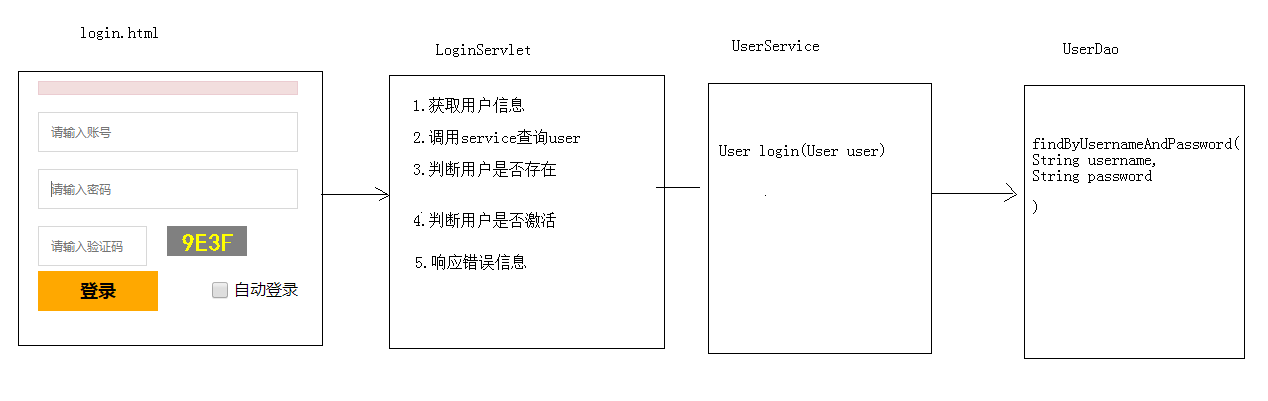
自行完成

### Step6：编写后端UserDao以及UserDaoImpl

自行完成

## 登录

### Step1：功能的实现分析



### Step2：阅读前台代码

在login.html中已给出如下ajax与后端交互的代码：



### Step3：编写后端LoginServlet

参考已经给出的RegistUserServlet，完成LoginServlet

### Step4：在UserService和UserDao添加登录相关代码

在UserService和UserDao中，参考注册功能的代码，完成接口和实现类的编写。

## index页面中用户姓名的提示信息功能

效果：



### Step1：阅读header.html的异步处理代码



### Step2：后端编写FindLoginUserServlet

思路提示：

（1）从session中获取登录用户

Object user = request.getSession().getAttribute(**"user"**);

（2）将user写回客户端

ObjectMapper mapper = new ObjectMapper();

response.setContentType("application/json;charset=utf-8");

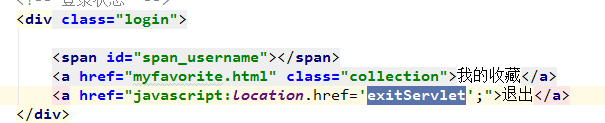
mapper.writeValue(response.getOutputStream(),user);

## 退出

什么叫做登录了？--🡪 session中有user对象。

### Step1: 阅读前端代码

Header.html中的代码



### Step2: 后端代码

ExitServlet的编写思路：

（1）将session销毁

（2）跳转到登录页面

## 成果展示

以你自己的名字注册、登录后，将index页面中用户姓名的提示信息功能展示出来。如下图所示：



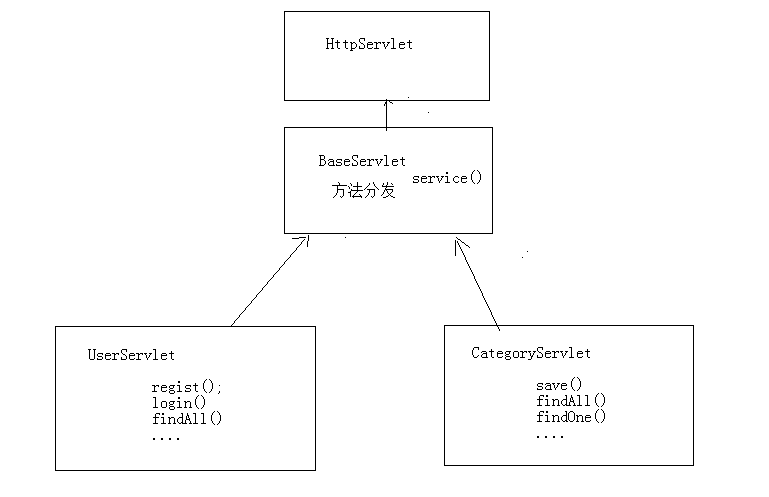
在下方表格中粘贴你的截图：

|  |
| --- |
| X6J6$ETOOV9V(NKNA8R@[)K |

# 任务3：优化Servlet

以上实现任务的过程中，一个功能做成了一个Servlet。当业务量增加时，就会出现数目繁多的servlet，不易维护。因此需要对servlet进行优化。

核心思想是：将其优化为一个模块一个Servlet，相当于在数据库中一张表对应一个Servlet，在Servlet中提供不同的方法，完成用户的请求。



## BaseServlet编写

该代码已经在《资料09-部分代码》中给出。

## UserServlet改写

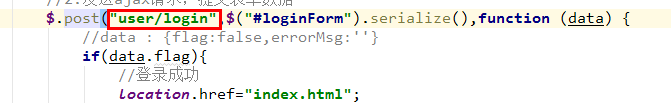
自行研读BaseServlet相关代码后，将之前的Servlet实现的功能，抽取到UserServlet中的不同方法中实现，并且将UserService创建抽取到成员变量位置。

## 页面路径改写

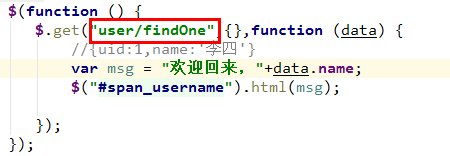
register.html



login.html



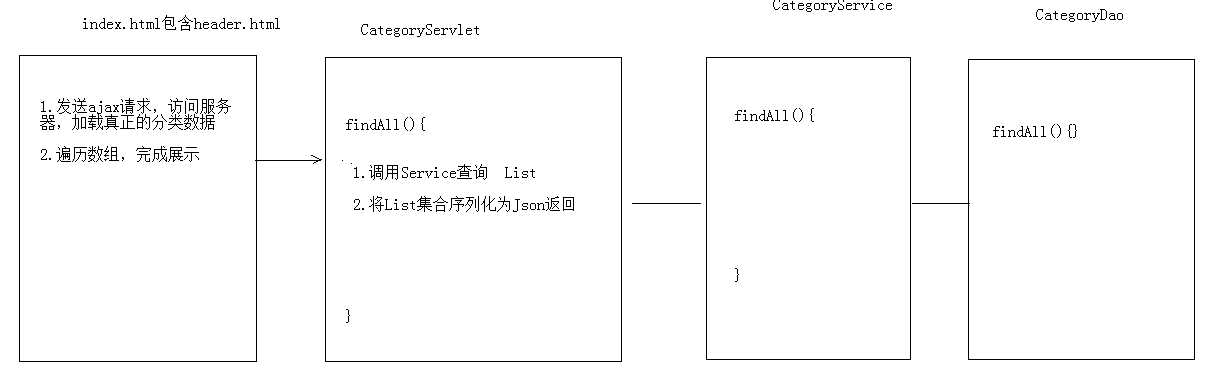
header.html



# 任务4：分类数据展示



## 分析



## 阅读前台代码

hader.html加载后，发送ajax请求，请求category/findAll



## 后台代码CategoryDao实现

新建CategoryDao接口，在接口中定义findAll()方法。

## 后台代码CategoryDaoImpl实现

新建CategoryDaoImpl类，实现接口CategoryDao。

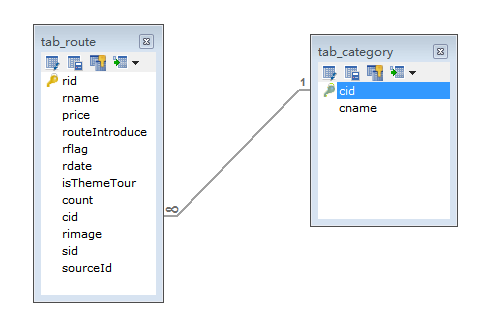
## 结果展示

在下面的表格中展示你的分类数据展示

|  |
| --- |
| IMG_256 |

# 任务5：旅游线路的分页展示

点击了不同的分类后，将来看到的旅游线路不一样的。通过分析数据库表结构，发现，旅游线路表和分类表时一个多对一的关系



查询不同分类的旅游线路sql

Select \* from tab\_route where cid = ?;

* 在html页面通过拼接字符串形式以href传递cid

header.html传递cid

**var** li = **'<li><a href="route\_list.html?cid='**+data[i].cid+**'">'**+data[i].cname+**'</a></li>'**;

* 获取cid

**$**(**function** () {

**var** search = ***location***.search;

//alert(search);//?id=5

// 切割字符串，拿到第二个值

**var** cid = search.split(**"="**)[1];

});

## 根据id查询不同类别的旅游线路数据分析



## 阅读前端代码

## 后端-阅读PageBean类，该类用来记录每一页

属性如下：

**private int totalCount**;//总记录数

**private int totalPage**;//总页数

**private int currentPage**;//当前页码

**private int pageSize**;//每页显示的条数

**private** List<T> **list**;//每页显示的数据集合

## 后端创建RouteServlet

## 后端创建RouteService

## 编写RouteDao接口及其实现RouteDaoImpl

## 展示分页效果

点击国内线路的第5页，截图如下：

|  |
| --- |
| SY64Q29F6DF~%O(`LOHSFQG |