



广东工业大学

软件体系结构实验报告

通讯录系统

学 院 _____ 计算机 _____

专 业 _____ 软件工程 _____

年级班别 _____ 14 级软件工程 3 班 _____

学 号 _____ 3214006278 _____

学生姓名 _____ 何炜婷 _____

指导教师 _____ 郝彦军 _____

2017 年 5 月

目录

一、SSM 框架 (Spring+SpringMVC+MyBatis+MySQL)	1
1.1 Spring 概念	1
1.2、Spring MVC 概念	1
1.3、MyBatis 概念	2
二、eclipse 配置 git 的过程	3
三、pring mvc 环境搭建	9
四、建立数据库	14
五、部分 mvc 架构代码解释	15
5.1 Spring+SpringMVC+MyBatis+MySQL 设计思想	15
5.2 代码例子解释 mvc	15
六、运用到的其他技术	19
6.1 md5 加密	19
6.2 excel 导出通讯录	19
七、测试	21
7.1 注册测试	21
7.2 登录测试	22
7.3 添加联系人测试	23
7.4 编辑联系人测试	24
7.5 删除联系人测试	25
7.6 查找联系人测试	26
7.7 excel 导出联系人测试	27
7.8 md5 密码加密测试	27
八、实验总结	28

一、SSM 框架 (Spring+SpringMVC+MyBatis+MySQL)

本次实验使用的是 SSM 框架。SSM 框架在项目开发中经常使用到,相比于 SSH 框架,它在仅几年的开发中运用的更加广泛。

Spring 作为一个轻量级的框架,有很多的拓展功能,最主要的我们一般项目使用的就是 IOC 和 AOP。

SpringMVC 是 Spring 实现的一个 Web 层,相当于 Struts 的框架,但是比 Struts 更加灵活和强大!

Mybatis 是一个持久层的框架,在使用上相比 Hibernate 更加灵活,可以控制 sql 的编写,使用 XML 或注解进行相关的配置。

1.1 Spring 概念

Spring 是一个开源框架, Spring 是于 2003 年兴起的一个轻量级的 Java 开发框架,由 Rod Johnson 在其著作 Expert One-On-One J2EE Development and Design 中阐述的部分理念和原型衍生而来。它是为了解决企业应用开发的复杂性而创建的。Spring 使用基本的 JavaBean 来完成以前只可能由 EJB 完成的事情。然而, Spring 的用途不仅限于服务器端的开发。从简单性、可测试性和松耦合的角度而言,任何 Java 应用都可以从 Spring 中受益。简单来说, Spring 是一个轻量级的控制反转 (IoC) 和面向切面 (AOP) 的容器框架。

1.2、Spring MVC 概念

Spring MVC 属于 SpringFrameWork 的后续产品,已经融合在 Spring Web Flow 里面。Spring MVC 分离了控制器、模型对象、分派器以及处理程序对象的角色,这种分离让它们更容易进行定制。

1.3、MyBatis 概念

MyBatis 本是 apache 的一个开源项目 iBatis, 2010 年这个项目由 apache software foundation 迁移到了 google code, 并且改名为 MyBatis。MyBatis 是一个基于 Java 的持久层框架。iBatis 提供的持久层框架包括 SQL Maps 和 Data Access Objects (DAO) MyBatis 消除了几乎所有的 JDBC 代码和参数的手工设置以及结果集的检索。MyBatis 使用简单的 XML 或注解用于配置和原始映射, 将接口和 Java 的 POJOs (Plain Old Java Objects, 普通的 Java 对象) 映射成数据库中的记录。

二、eclipse 配置 git 的过程

①调出 eclipse 中的配置 git 的窗口

在菜单的 window 中的 show view 选中 other，然后在弹框中输入 git，然后选中 git repositories，即可显示出 git 项目的窗口框。

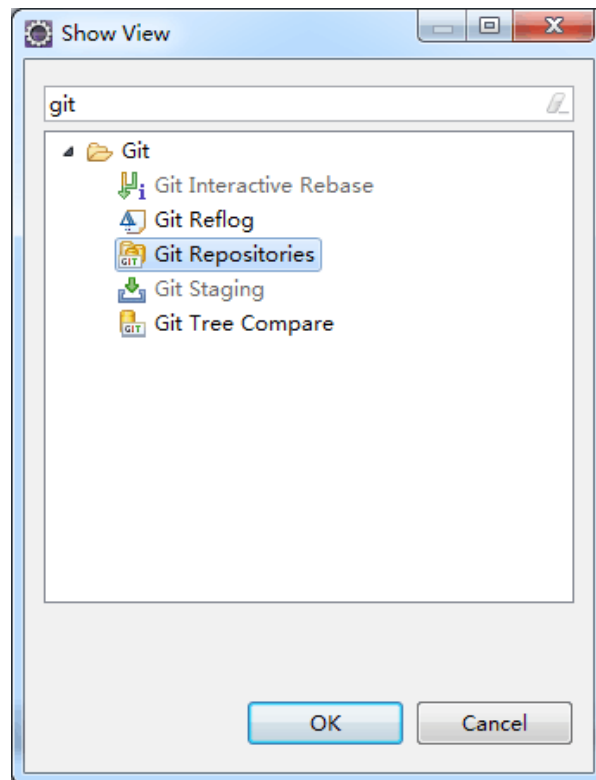


图 2.1 show view 调出 git repositories 窗口图

②导入项目目录，并进行初始化。

在 git repositories 窗口中点击 clone a git repository 导入 git 项目，选择所要导入的 git 项目的路径 url 即可将导入 git 项目。

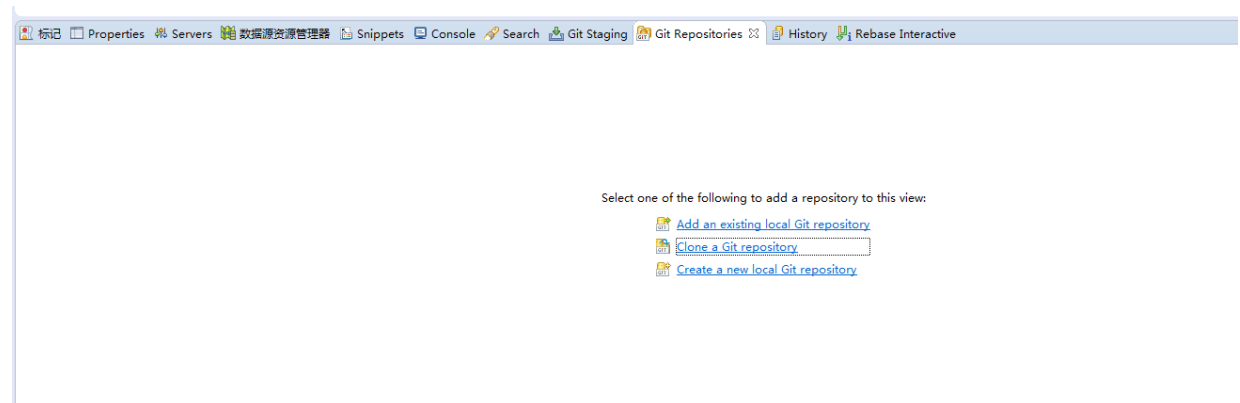


图 2.2 git 项目窗口

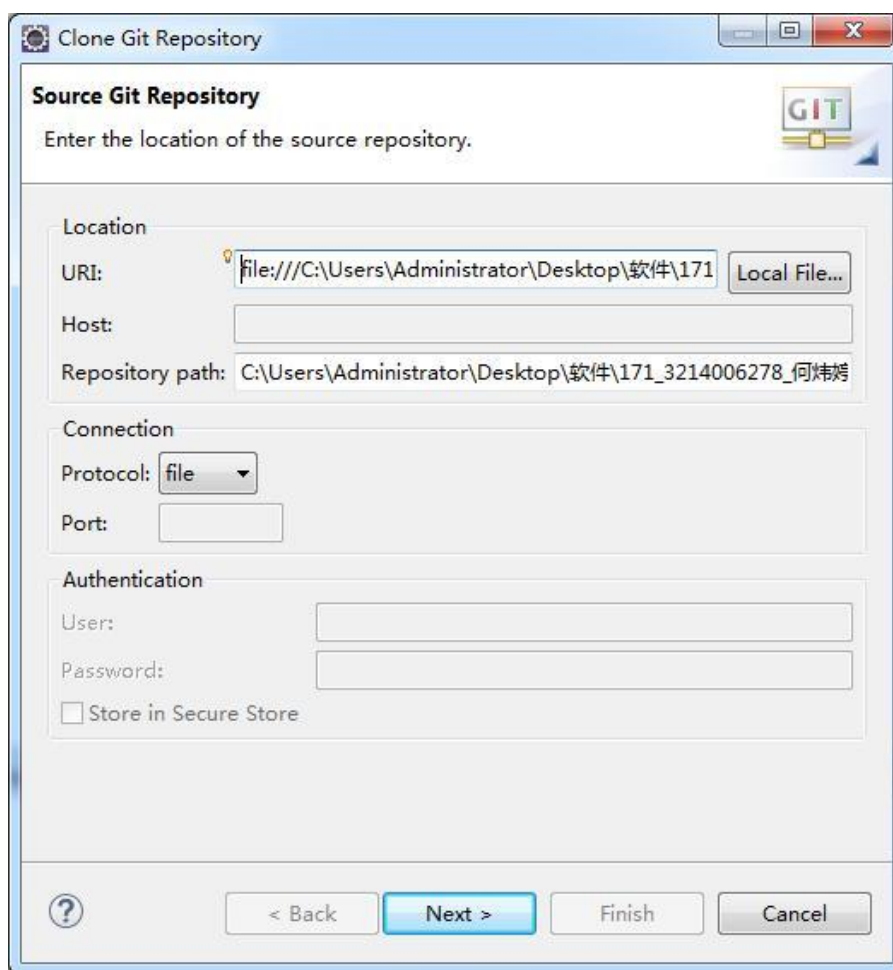


图 2.3 选择所要导入的 git 项目的路径



图 2.4 成功导入 git 项目后的窗口显示结果

在 git repositories 窗口中选中所要导入进 Eclipse 的项目，右键点击，选择导入项目这一项，然后默认确认，即可把 git 项目导入进 eclipse 的项目 project 中。

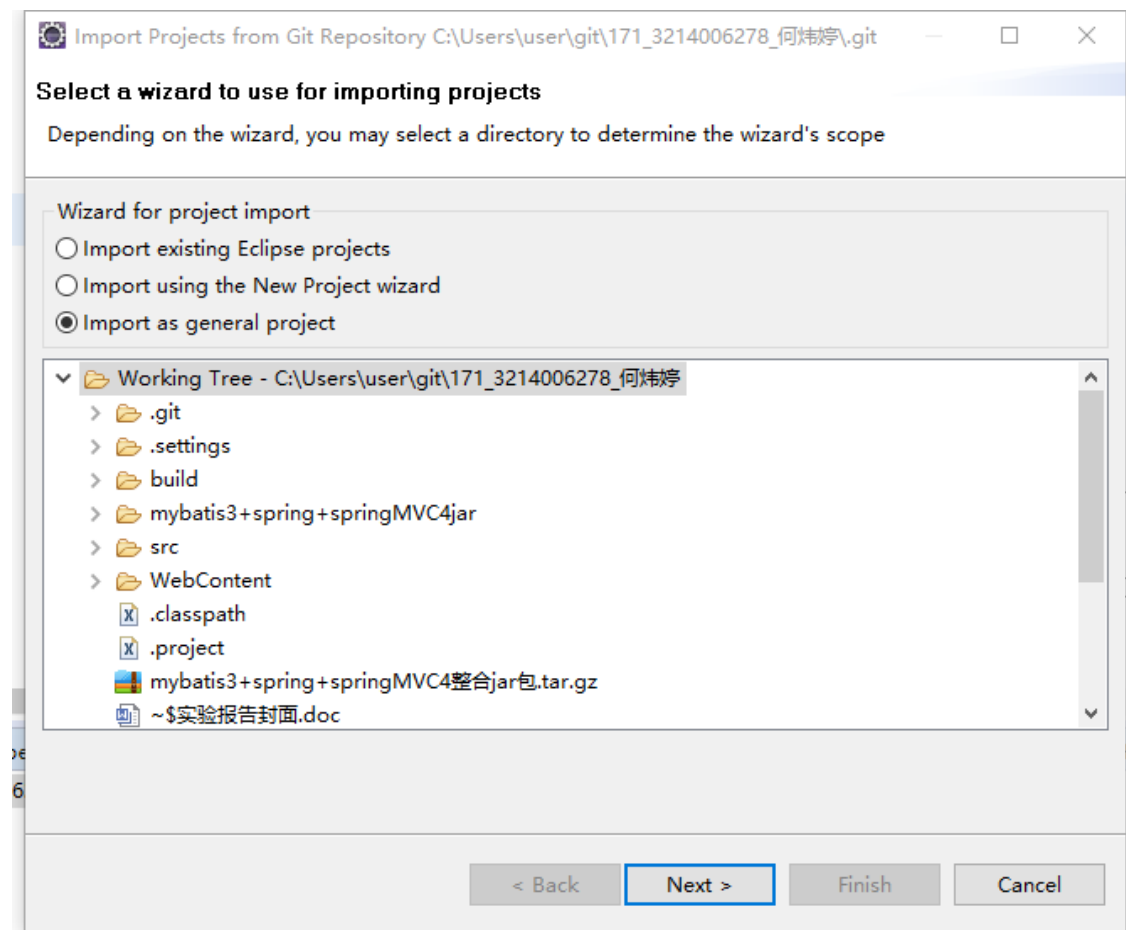


图 2.5 导入 eclipse 项目

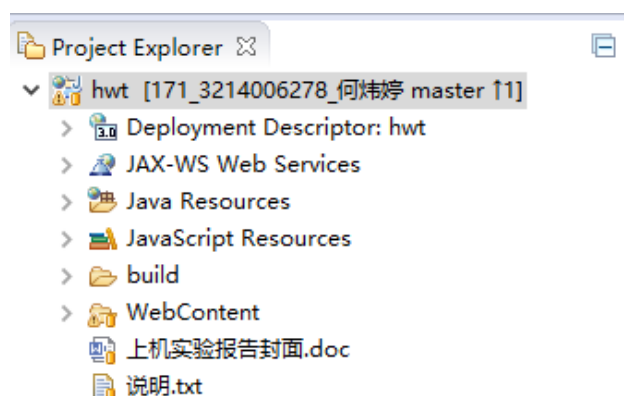


图 2.6 成功导入 eclipse 项目结果

③设置好用户信息

在菜单栏 window 中，选择 preference，弹出选项框，然后选项 team 的 git 设置选项，再选中 configuration，在 user settings 用户名设置中添加/移除用户信息，按照图 1.7 的用户信息设置好。

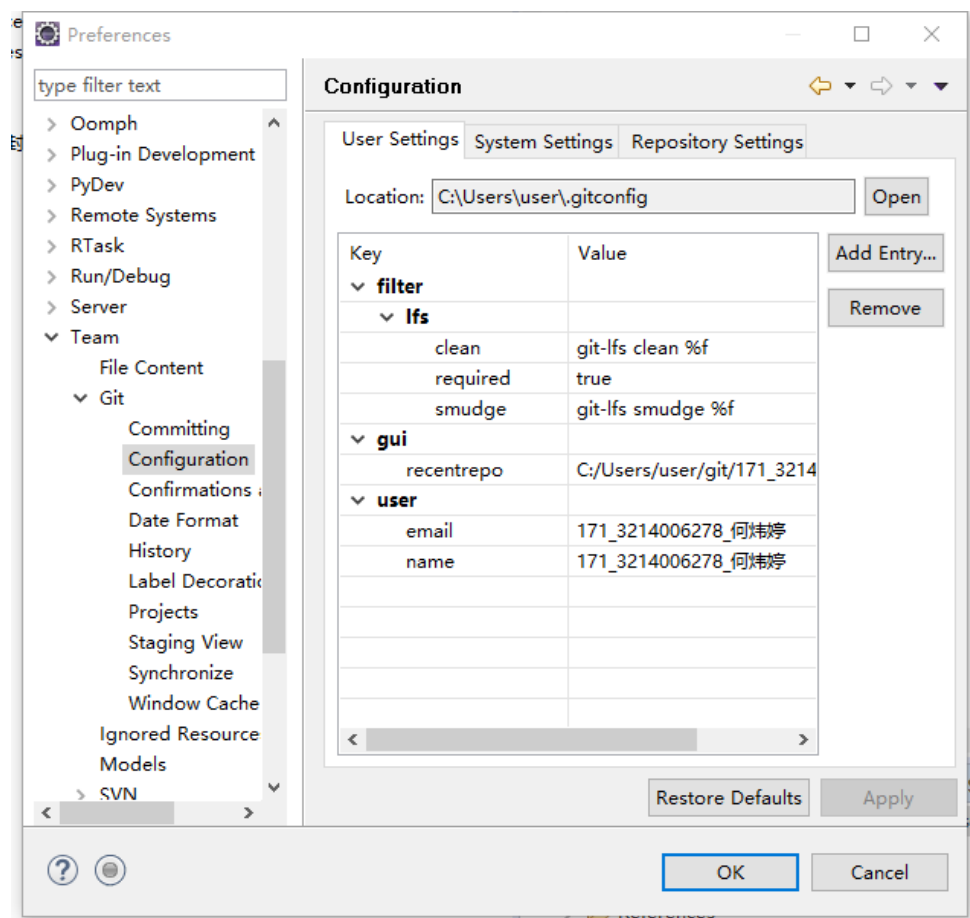


图 2.7 设置用户信息

④commit 项目

每次更新项目时，在 git repositories 窗口中选中项目点击右键，然后选择点击 commit 选项，弹出 commit changes 弹框，填写 commit 项目的相关内容信息。然后点击 commit 按钮，即可本地保存此时项目的版本。

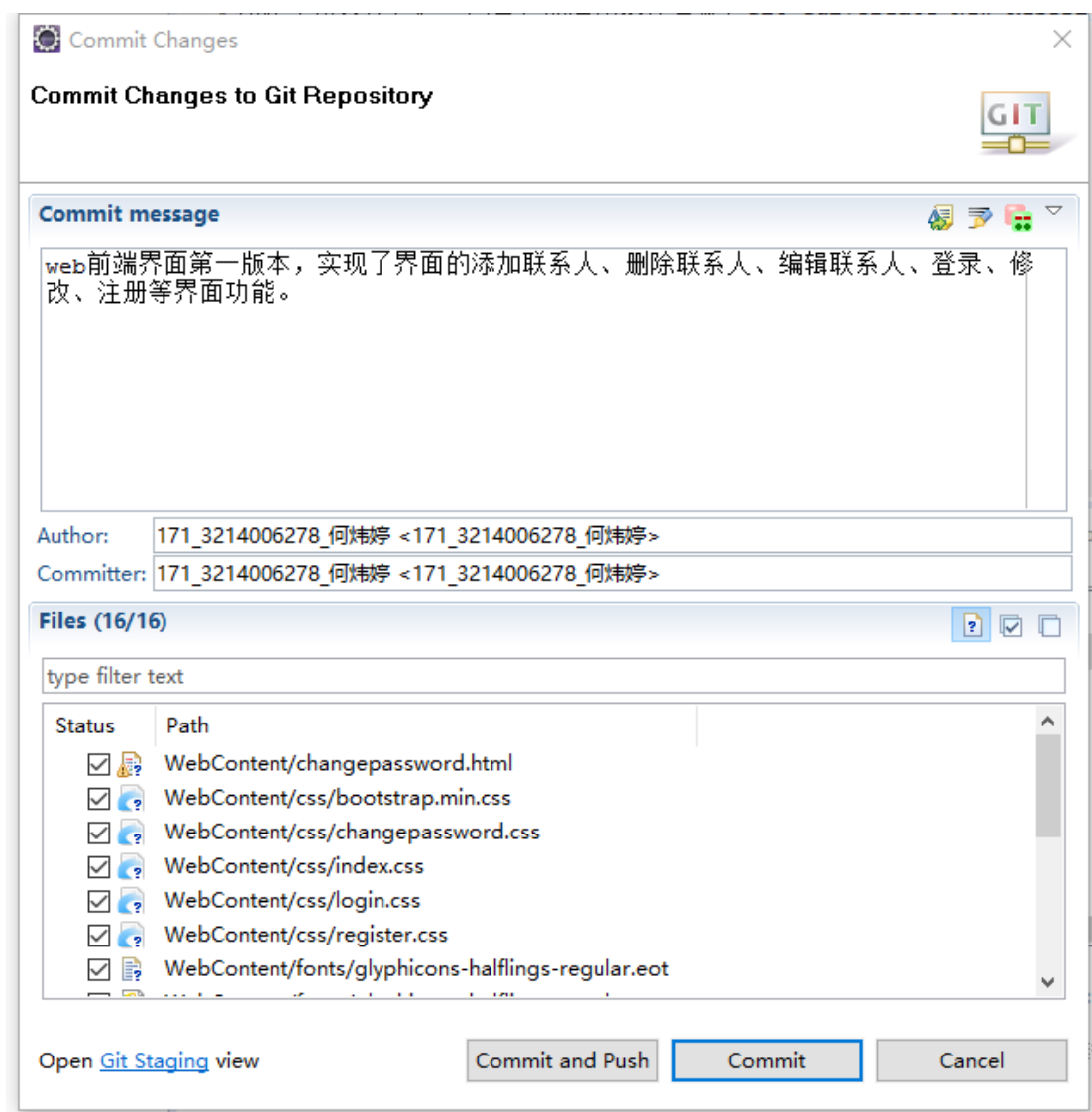


图 2.8 commit 保存项目版本

如果想要远程上传和下载项目，也是在 git repositories 窗口中右键点击项目，然后点击 push/pull 选项，填入相关的远程地址、用户名、密码等信息进行相关操作即可。

⑤查看 commit 历史记录

成功保存项目的版本之后，如果想查看 commit 的历史记录或者历史版本信息,则在git repositories窗口中右键点击项目,然后点击 show in选择history,即可显示出所有项目历史版本的信息。



图 2.9 查看 commit 历史记录

三、spring mvc 环境搭建

①创建包目录与修改包名

依据网上的 demo 修改而来，demo 来源为传智播客 springmvc+mybatis 教学资源，根据 demo 修改包名，修改配置文件中的配置项。

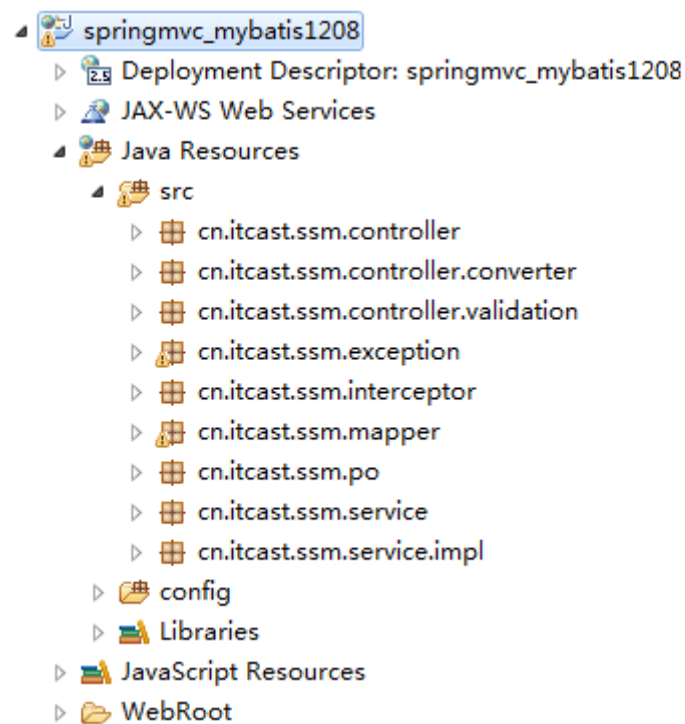


图 3.1 修改包名后的包

②导入 mybatis3+spring+springMVC4 整合 jar 包

用文件夹导入的方式，在项目中，把下载的 mybatis3+spring+springMVC4 整合 jar 包文件夹拷贝到项目中，选中需要添加到项目中的 jar 包，右键选 Build Path 中的 Add to Build Path，这样 Jar 包就成功导入到项目中了。

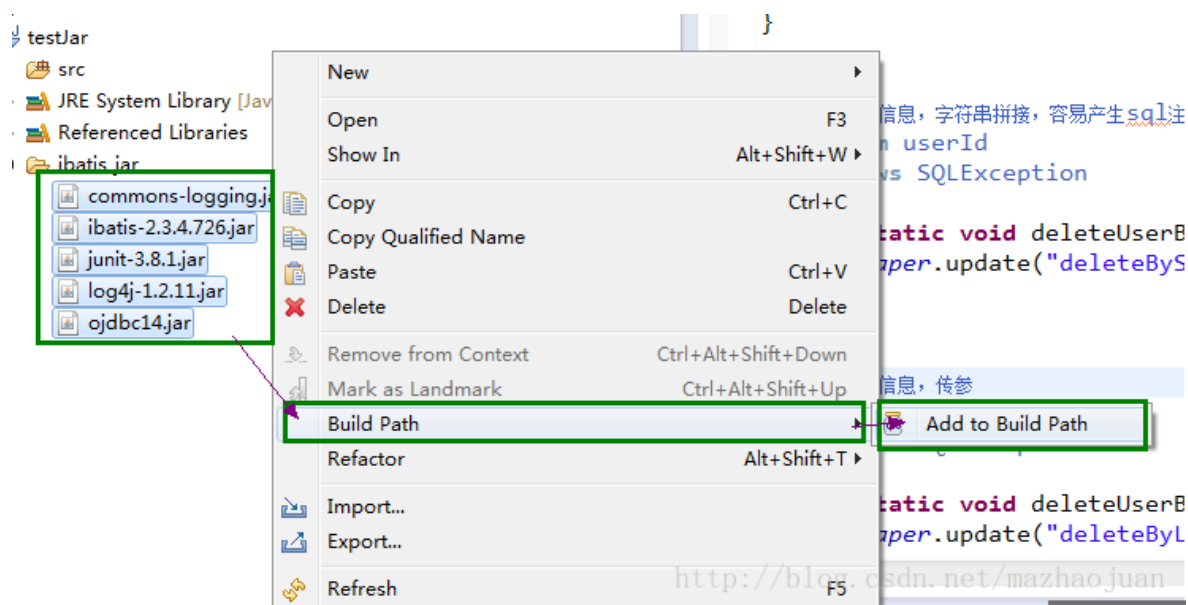


图 3.2 用文件夹导入的方式导入 jar 包

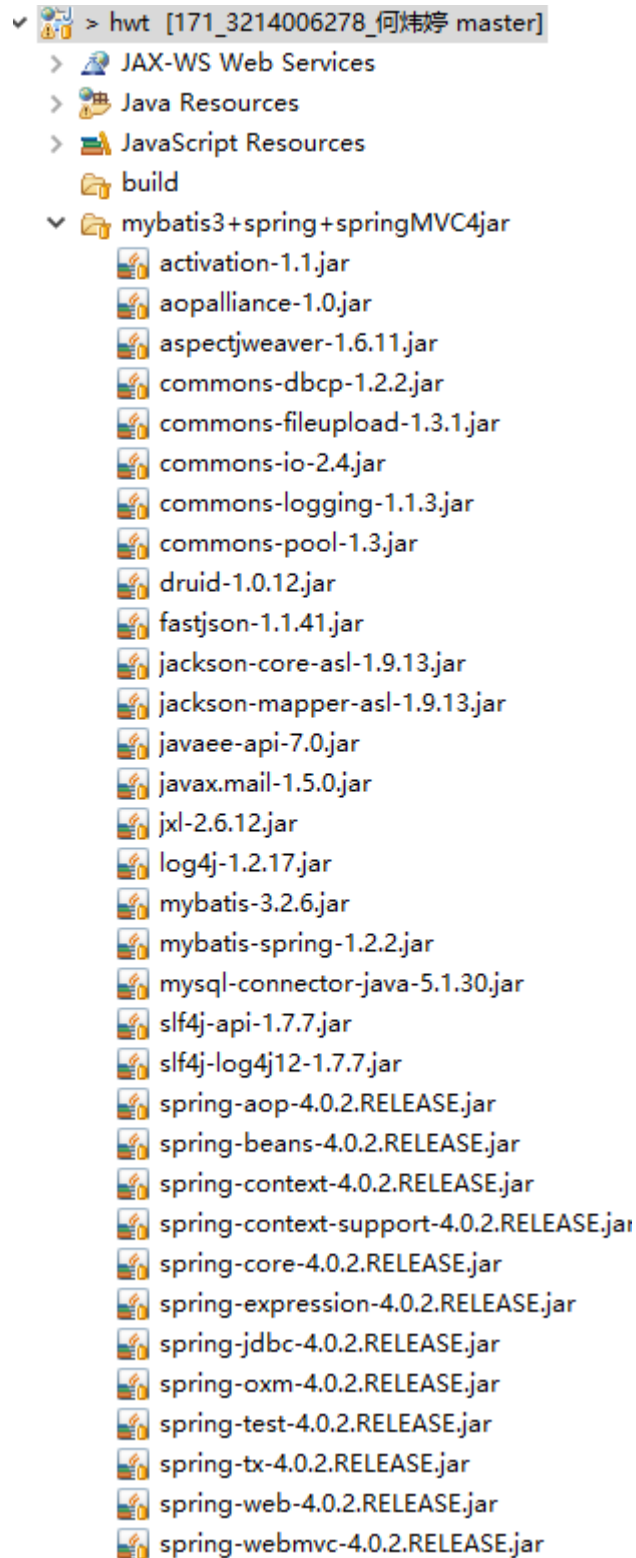


图 3.3 mybatis3+spring+springMVC4 整合 jar 包目录

③修改后台框架配置文件

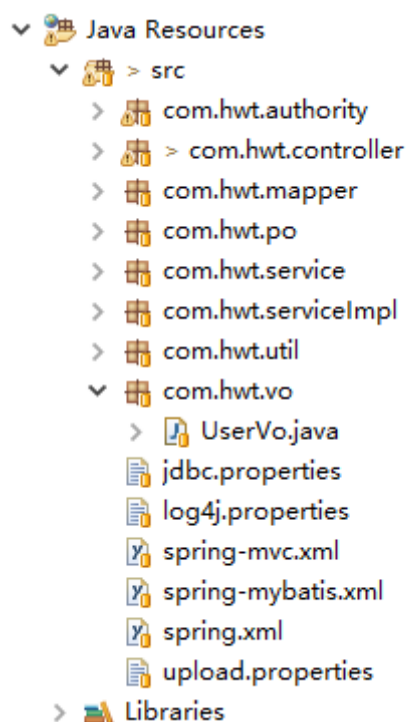


图 3.4 需要设置配置的文件

- (1) jdbc.properties 是数据库连接配置文件, 是 java 连接数据库的一个类。jdbc_url 是数据库连接指向地址, username 与 password 是数据库用户名与密码

```
1 jdbc_driver = com.mysql.jdbc.Driver
2 jdbc_url = jdbc:mysql://localhost:3306/test2?
3 jdbc_username = root
4 jdbc_password = 88888888
```

图 3.5 数据库连接配置代码

- (2) log4j.properties 是 Log 日记, 正常不改用默认配置。

- (3) spring-mvc.xml 是前端控制器, 需要改一些相关的包名

```
<mvc:annotation-driven/>
<!-- 自动扫描controller包下的所有类, 使其认为spring mvc的控制器 -->
<context:component-scan base-package="com.hwt.controller" />
```

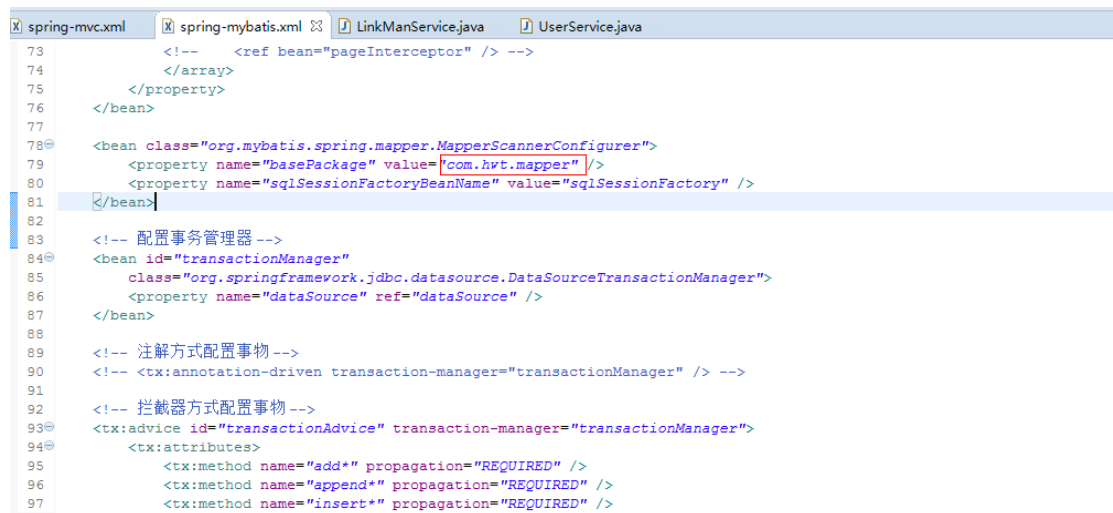
图 3.6 spring.xml 需要修改的包名

(4) spring-mybatis.xml 是一个 dao 层的工具，直接连接数据库的模块。需要修改的地方如下图所示。



```
100 <tx:method name="modify*" propagation="REQUIRED" />
101 <tx:method name="edit*" propagation="REQUIRED" />
102 <tx:method name="delete*" propagation="REQUIRED" />
103 <tx:method name="remove*" propagation="REQUIRED" />
104 <tx:method name="repair" propagation="REQUIRED" />
105 <tx:method name="delAndRepair" propagation="REQUIRED" />
106
107 <tx:method name="get*" propagation="SUPPORTS" />
108 <tx:method name="find*" propagation="SUPPORTS" />
109 <tx:method name="load*" propagation="SUPPORTS" />
110 <tx:method name="search*" propagation="SUPPORTS" />
111 <tx:method name="datagrid*" propagation="SUPPORTS" />
112
113 <tx:method name="*" propagation="SUPPORTS" />
114 </tx:attributes>
115 </tx:advice>
116 <aop:config>
117 <aop:pointcut id="transactionPointcut"
118 expression="execution(* com.hwt.service.**Impl.*(..))" />
119 <aop:advisor pointcut-ref="transactionPointcut"
120 advice-ref="transactionAdvice" />
121 </aop:config>
122
123 <!-- 配置druid监控spring jdbc -->
124
```

图 3.7 声明式事务管理搜索的包配置



```
73 <!-- <ref bean="pageInterceptor" /> -->
74 </array>
75 </property>
76 </bean>
77
78 <bean class="org.mybatis.spring.mapper.MapperScannerConfigurer">
79 <property name="basePackage" value="com.hwt.mapper" />
80 <property name="sqlSessionFactoryBeanName" value="sqlSessionFactory" />
81 </bean>
82
83 <!-- 配置事务管理器 -->
84 <bean id="transactionManager"
85 class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
86 <property name="dataSource" ref="dataSource" />
87 </bean>
88
89 <!-- 注解方式配置事物 -->
90 <!-- <tx:annotation-driven transaction-manager="transactionManager" /> -->
91
92 <!-- 拦截器方式配置事物 -->
93 <tx:advice id="transactionAdvice" transaction-manager="transactionManager">
94 <tx:attributes>
95 <tx:method name="add*" propagation="REQUIRED" />
96 <tx:method name="append*" propagation="REQUIRED" />
97 <tx:method name="insert*" propagation="REQUIRED" />
```

图 3.8 mybatis 中 mapper 包所在位置配置

(5) spring.xml 是自动装配。需要把包名修改为自己的项目相关的包名。

```
<!-- 引入属性文件 -->
<context:property-placeholder location="classpath:jdbc.properties" />

<context:annotation-config />
<!-- 自动扫描(自动注入) -->
<context:component-scan base-package="com.hwt.serviceImpl" />
```

图 3.9 spring.xml 中需要修改的代码

(6) upload.properties 是上传下载目录的配置等

四、建立数据库

根据系统的需求设计数据库，建立数据库与建表。如下图所示：

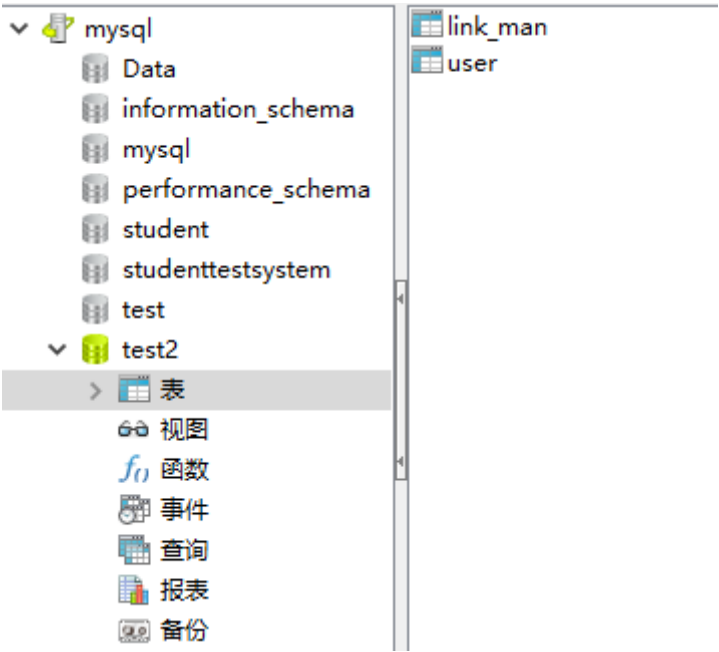


图 4.1 建立数据库 test2

A screenshot of a database management tool's table structure view for the 'link_man' table. The window title is 'link_man @test2 (mysql) ...'. The menu bar includes '文件', '编辑', '查看', '窗口', and '帮助'. The toolbar includes '导入向导', '导出向导', '筛选向导', '网格查看', and '表单查看'. The table structure is shown in a grid with columns: 'id', 'account', 'tel', and 'userId'. The first row shows all columns as '(Null)'.

id	account	tel	userId
(Null)	(Null)	(Null)	(Null)

图 4.1 建 link_man 通讯录信息表

A screenshot of a database management tool's table structure view for the 'user' table. The window title is 'user @test2 (mysql) - 表'. The menu bar includes '文件', '编辑', '查看', '窗口', and '帮助'. The toolbar includes '导入向导', '导出向导', '筛选向导', '网格查看', '表单查看', '备注', and a plus icon. The table structure is shown in a grid with columns: 'id', 'username', and 'password'. The first row shows data: '4', '何炜婷', and '726a579dd9de03976c949cfc4c98d093'.

id	username	password
4	何炜婷	726a579dd9de03976c949cfc4c98d093

图 4.1 建 user 用户信息表

五、部分 mvc 架构代码解释

5.1 Spring+SpringMVC+MyBatis+MySQL 设计思想

- ①controller 层调用 service 层，service 层调用 dao 层。
- ②controller 是跟前端接触的第一层。
- ③service 是处理复杂数据的一层，一般处理 dao 层获取的数据。
- ④dao 层是与数据库交互的直接层，就是 dao 层获取数据库的数据。
- ⑤在本系统中就是 mapper 包下的文件，xml 中是其 sql 实现。.java 文件是其函数方法声明。

5.2 代码例子解释 mvc

很多功能的编写代码的方式都是类似的，都是基于 mvc 架构来编写，所以就简单地举例一个功能的代码来演示 spring mvc 的设计模式编写代码的思想。

添加联系人功能：

- ①视图中 js 的 ajax 请求向 controller 控制器层请求。

```
$.ajax({  
    async : true,  
    dataType : 'json',  
    type : "POST",  
    url : 'user/AddLinkMan.do', //url 为在 controller 层中的地址  
    data : {  
        Account :AddAccount,  
        Tel : AddTel,  
    },  
    error : function(request) {  
    },  
    success : function(data) {  
    }  
});
```

②controller 层调用 service 层。

```
@Auth(role = "admin,user", description = "AddLinkMan")//标明权限
@ResponseBody()//设置返回对象为 json
@RequestMapping("AddLinkMan") //对应着视图中 ajax 请求的 url 地址
public Json AddLinkMan(String Account, String Tel,
                        HttpSession session) {
    Json json = new Json();
    User user=(User)session.getAttribute("userSession");//获取用户
    的登录信息
    if(user==null){
        json.success = false;
        json.message = "登录已经失效，请重新登录";
        json.href = "login.html";
        json.data="-1";
        return json;
    }
    List<LinkMan> linkManList=
linkManService.selectLinkManByTel(Tel,user.getId());//先查看电话号码
是否已存在
    if(linkManList.size()!=0){
        json.success = false;
        json.message = "电话存在重复，请确认后重新添加";
        json.data="0";
        return json;
    }else{
        LinkMan linkMan=new LinkMan();
        linkMan.setAccount(Account);
        linkMan.setTel(Tel);
```

```

        linkMan.setUserId(user.getId());

        linkManService.add(linkMan); //添加输入的联系人及其电话

        json.success = true;

        json.message = "添加成功";

        return json;
    }
}

```

③service 处理 dao 层获取的数据

```

public interface LinkManService extends BaseService<LinkMan, String> {

    List<LinkMan> selectLinkMan(String userId, String account, String
tel);

    List<LinkMan> selectLinkManByTel(String tel, String userId);

}

public interface BaseService<T, ID extends Serializable> {

    int add(T t);

    int deleteById(ID id);

    int update(T t);

    T getById(ID id);

    List<T> list(T t);

}

```

④dao 层获取数据库的数据，与数据库交互的直接层

```
<select id="selectLinkManByTel" resultMap="LinkMan">
  select * from `link_man`
  where userId=#{userId}
    and tel= #{tel}
</select>
```

```
<insert id="insert" parameterType="LinkMan">
  insert into `link_man`
    (`account`,`tel`,`userId`)
  values
    (#{account},#{tel},#{userId})
</insert>
```

六、运用到的其他技术

6.1 md5 加密

MD5 即 Message-Digest Algorithm 5（信息-摘要算法 5），用于确保信息传输完整一致。是计算机广泛使用的杂凑算法之一（又译摘要算法、哈希算法），主流编程语言普遍已有 MD5 实现。将数据（如汉字）运算为另一固定长度值，是杂凑算法的基础原理。

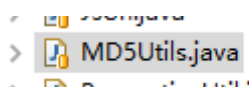


图 6.1 md5 加密的 java 类文件

在网上下载一个 md5 的 java 文件，然后放在项目文件中，在一个 java 文件中导入该类包 `import com.hwt.util.MD5Utils;`，然后直接在该文件代码中调用其方法即可，例：`user.setPassword(MD5Utils.getMD5(newPassword));`

6.2 excel 导出通讯录

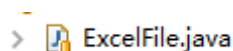


图 6.1 excel 导出的 java 类文件

在网上下载一个导出 excel 的 java 文件，然后放在项目文件中，在一个 java 文件中导入该类包 `import com.hwt.util.ExcelFile;` 然后直接在该文件代码中调用其方法即可，例：

```
String title[] = {"通讯录"};
List<List<String>> header=new ArrayList<List<String>>();
//表头数据
List<String> header0 =new ArrayList<String>();
header0.add("姓名/Name");
header0.add("号码/Tel");
//表头
header.add(header0);
//查数据
List<List<List<String>>> data = new
ArrayList<List<List<String>>>();
List<List<String>> data0 =new ArrayList<List<String>>();
//个人信息表
List<LinkMan> linkManList=
linkManService.selectLinkMan(user.getId(),null,null);
for (int i = 0; i < linkManList.size(); ++i) {
    List<String> aList =new ArrayList<String>();
    aList.add(linkManList.get(i).getAccount());
```

```

        aList.add(linkManList.get(i).getTel());
        data0.add(aList);
    }
    data.add(data0);
    ExcelFile ef = new ExcelFile(title, header, data, 1);
    try {
        ef.saveMultisheet(request, response, user.getUsername() + "
的通讯录");
    } catch (IOException e) {
        e.printStackTrace();
    }
    json.success = true;
    json.message = "导出成功";
    return json;

```


七、测试

7.1 注册测试

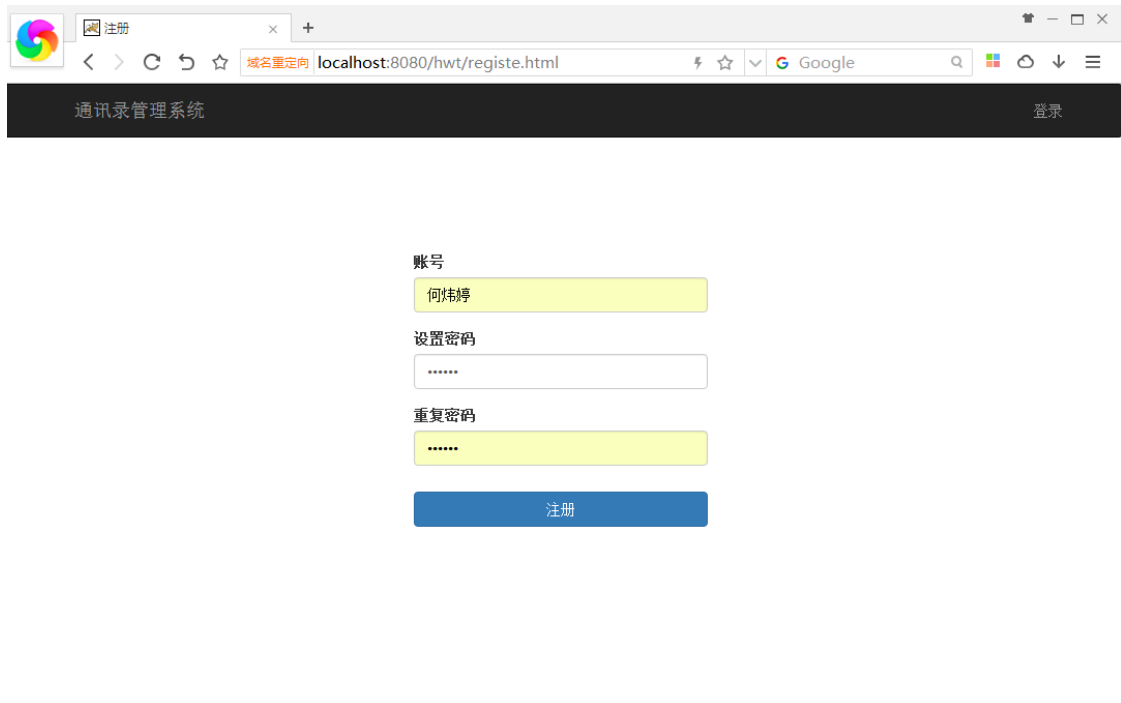


图 7.1 注册测试

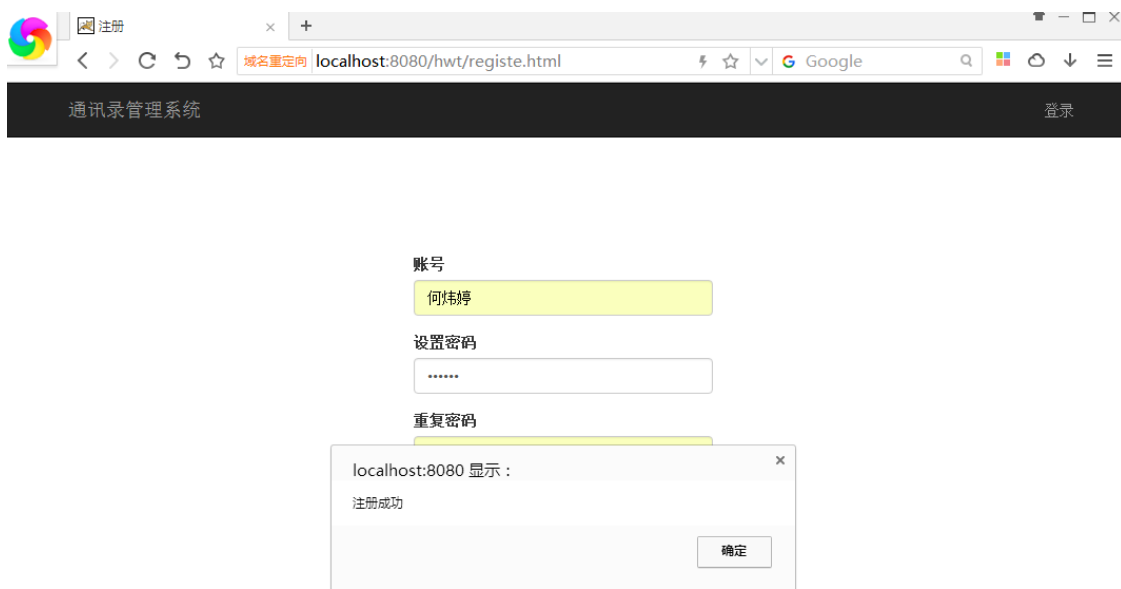


图 7.2 注册成功

7.2 登录测试

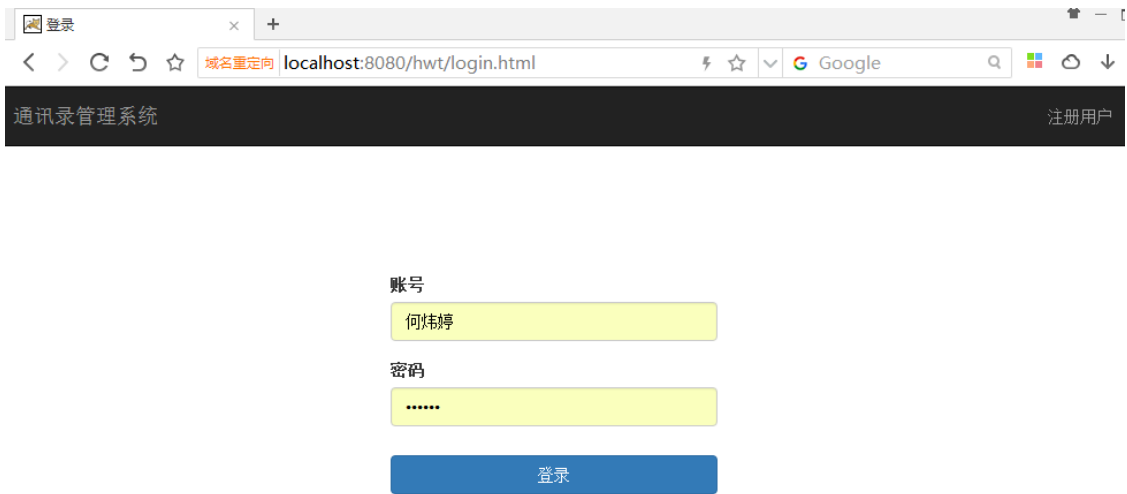


图 7.3 登录测试



图 7.4 登录成功

7.3 添加联系人测试

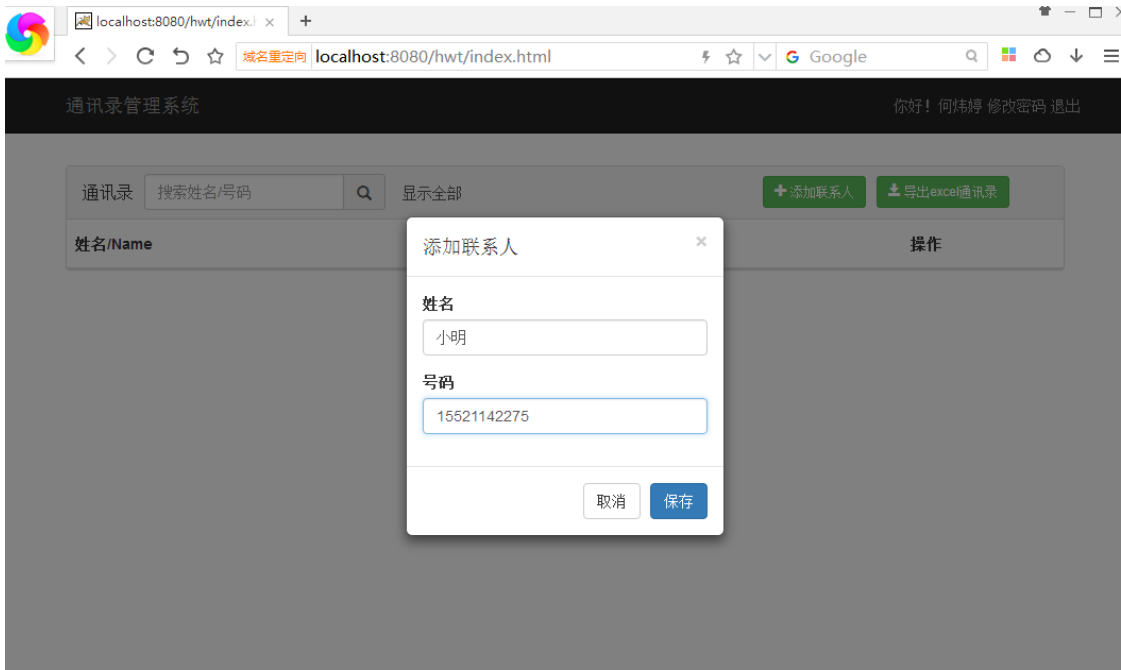


图 7.5 添加联系人



图 7.6 成功添加联系人

7.4 编辑联系人测试

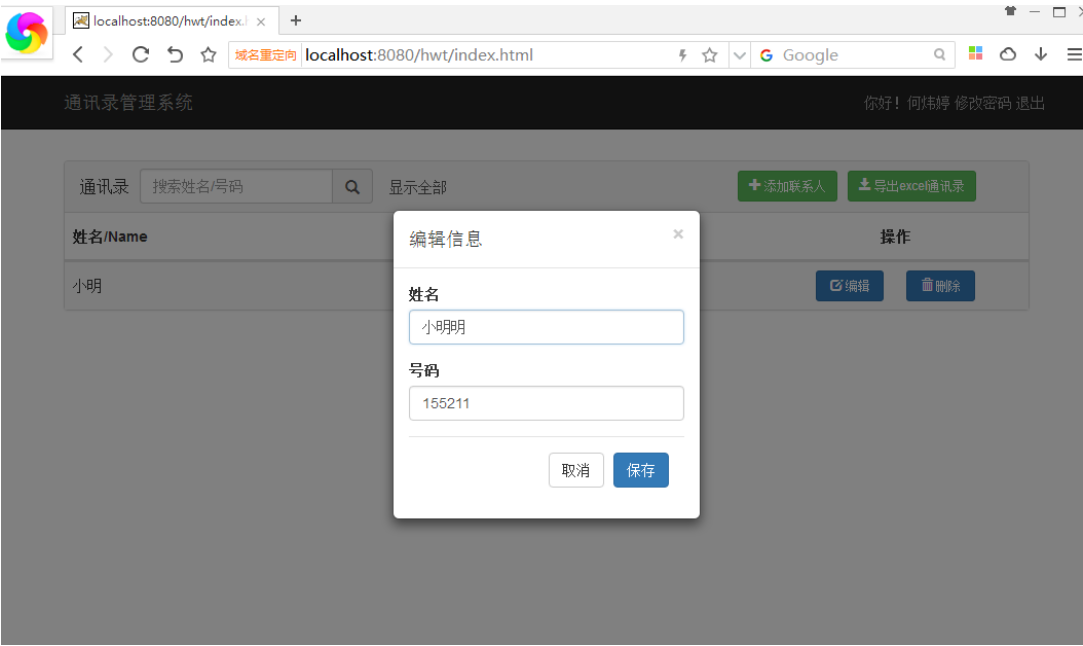


图 7.7 编辑联系人



图 7.8 成功编辑联系人

7.5 删除联系人测试

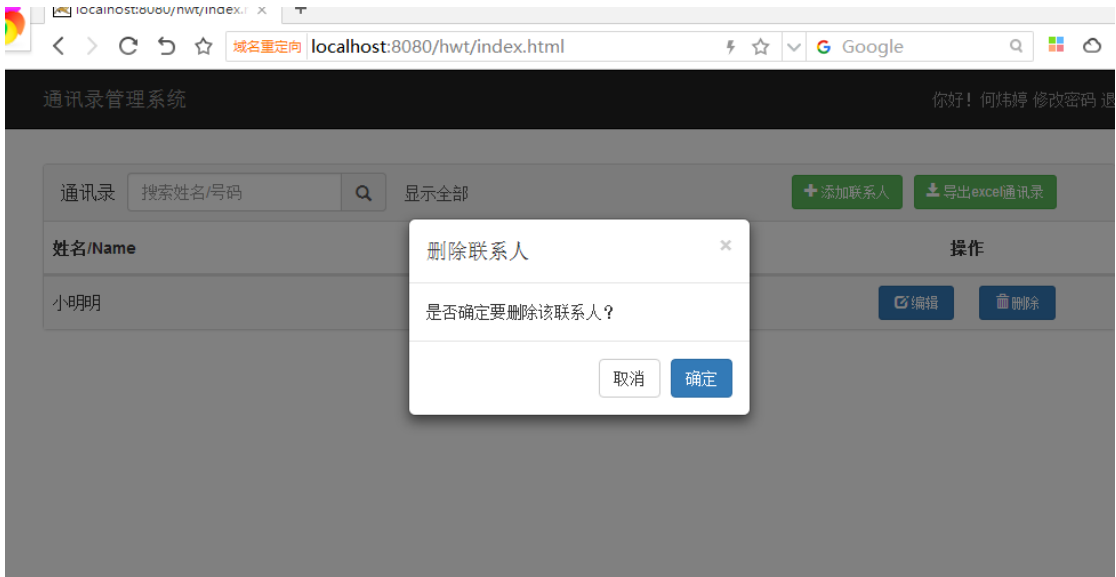


图 7.9 删除联系人



图 7.10 成功删除联系人

7.6 查找联系人测试

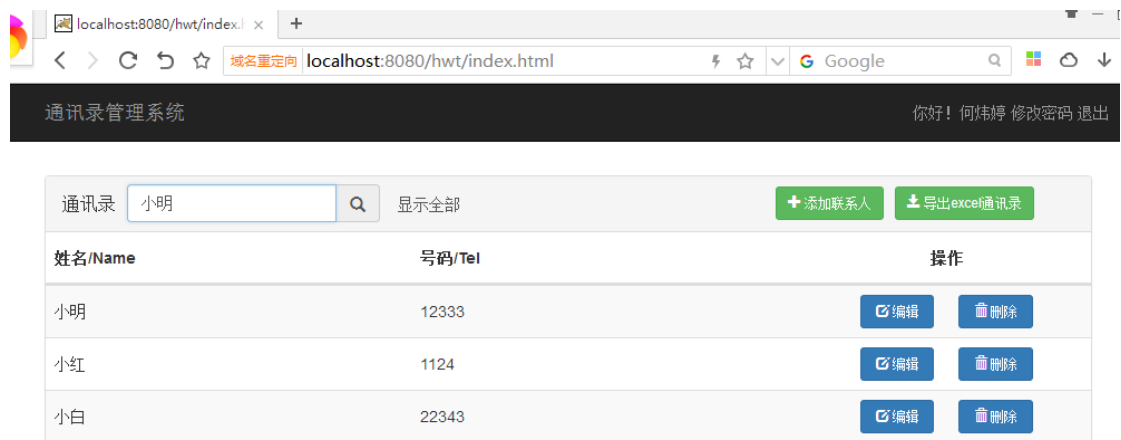


图 7.9 查找联系人



图 7.10 成功查找联系人

7.7 excel 导出联系人测试

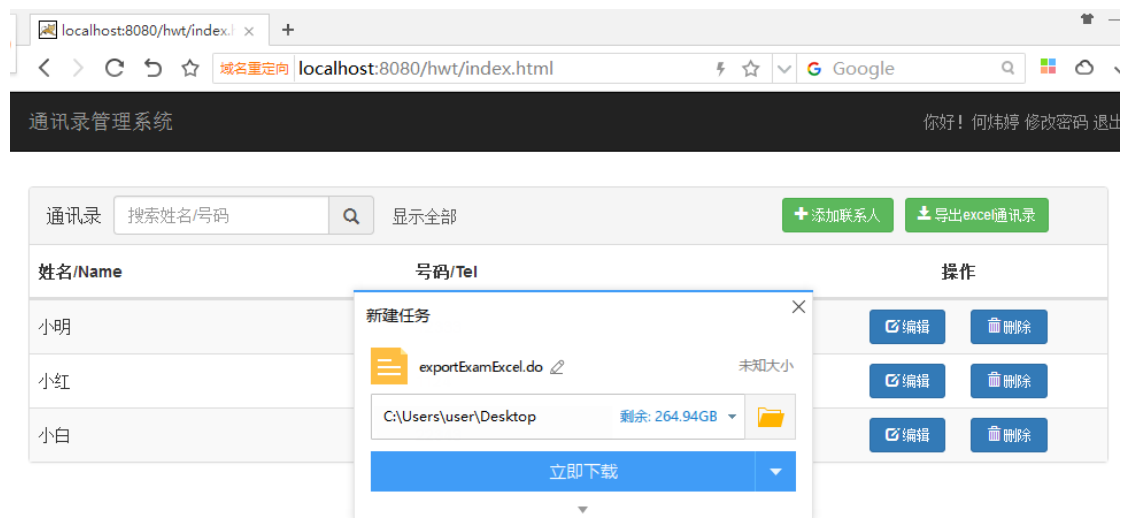


图 7.11 excel 导出联系人

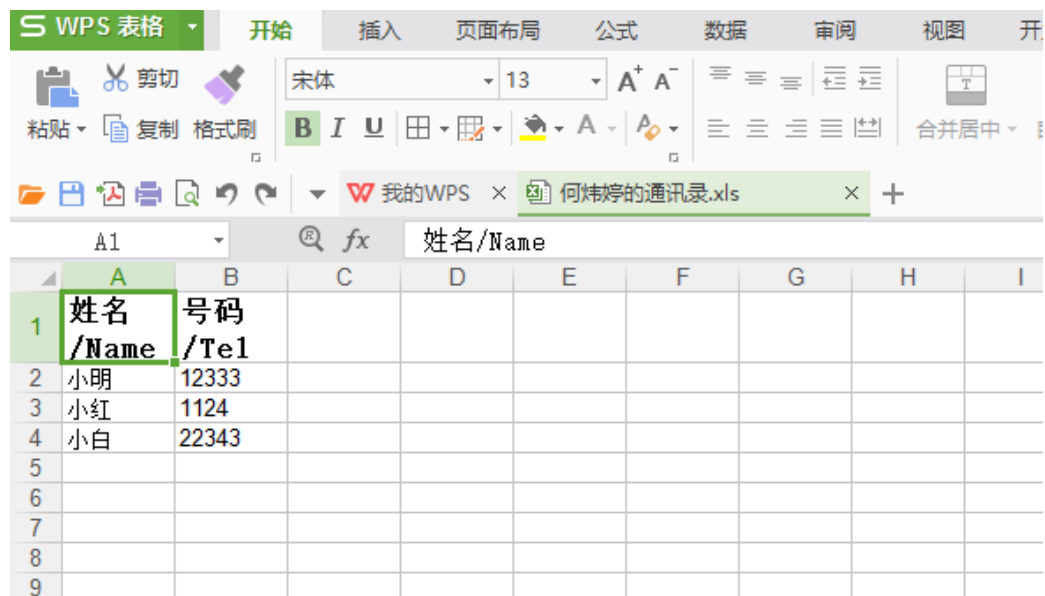


图 7.12 成功导出联系人 excel 表

7.8 md5 密码加密测试

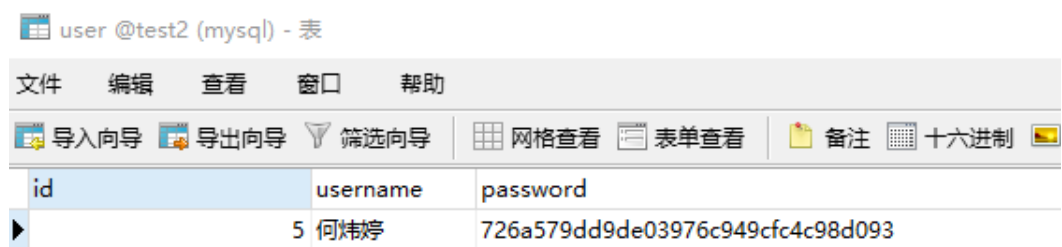


图 7.13 数据库中查看密码已被 md5 加密

八、实验总结

此次实验,我学会了怎样搭建 spring mvc 的后台框架,并且了解到了 spring mvc 的设计思想和工作原理,并且在项目中也使用了 spring mvc 来搭建项目的后台部分。

在此次实验中,我还学会了如何搭建一个 git 项目,并且如何配置 git 项目,怎么使用 git 来控制项目的历史版本,不止学会了怎么使用本地库也学会了使用远程库。

在这次实验中,我还尝试了 md5 加密技术与在网页中用 excel 导出信息的技术。

在此次实验中,对于我来说最大的挑战是,我以前接触得最多的是前端部分的代码,后台部分的代码很少接触,所以这是我这次实验的最大挑战,也让我学到了很多后台部分的代码知识与技术。