**尚硅谷毕业设计之学生管理系统**

## 前言

学生管理系统：主要是以年级、班级为单位，进行老师和学生信息记录和统计功能。项目前端采用JSP+JSTL+EasyUI来实现页面效果展示，后端服务采用SpringMVC+Spring+Mybatis框架。存储层使用高性能的MySQL，服务使用的是Tomcat服务器，使用Maven来管理jar包和项目构建。

## 环境准备：

### 1.1、开发工具:

建议使用IntelliJ IDEA 2017.2 x64及以上版本

IntelliJ IDEA详细使用教程请参考尚硅谷官方免费教程

<http://www.atguigu.com/download_detail.shtml?v=11>

### 1.2、Jdk:

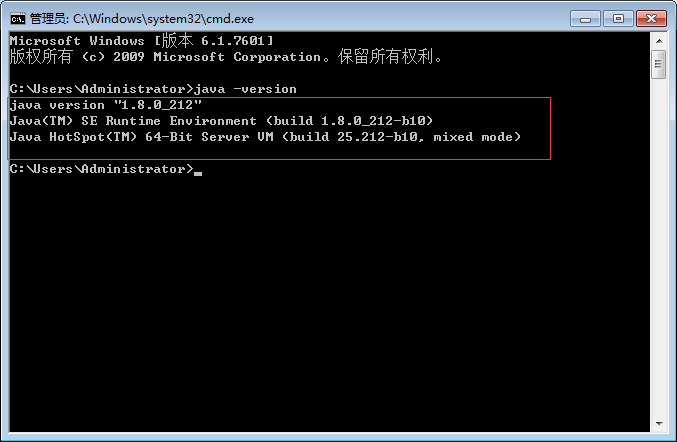
版本：java1.8

验证安装是否成功

1、打开cmd窗口

2、输入java –version

如图所示：



3、出现上图中红线部分即安装jdk成功

JDK详细教程请参考尚硅谷官方免费教程

<http://www.atguigu.com/download_detail.shtml?v=129>

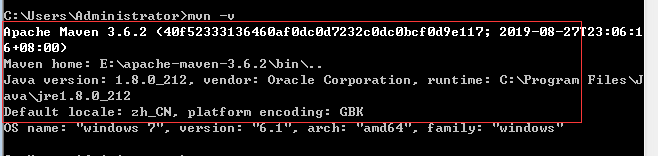
### 1.3、Maven：

验证安装是否成功

1、打开cmd窗口

2、输入mvn -v

如图所示



3、出现上图中红线部分即安装maven成功

Maven详细使用教程请参考尚硅谷官方免费教程

<http://www.atguigu.com/download_detail.shtml?v=286>

### 1.4、Tomcat：

只需要解压到目录下即可，最好目录中不要出现中文。

Tomcat详细使用教程请参考尚硅谷官方免费教程

<http://www.atguigu.com/download_detail.shtml?v=280>

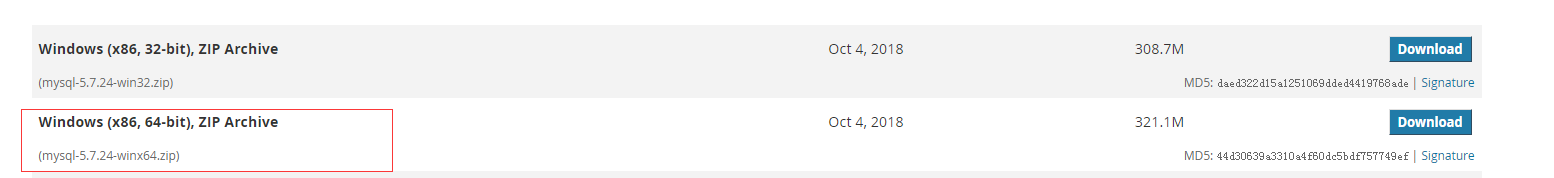
### 1.5、Mysql：

Windows安装mysql服务端

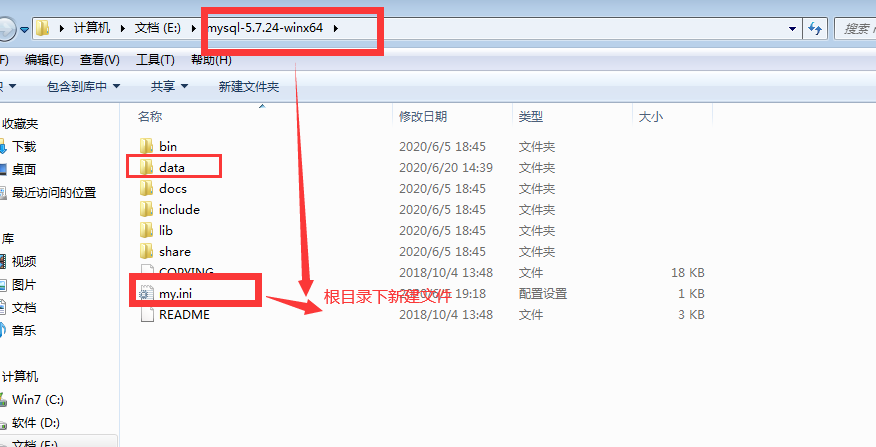
1. 下载5.7.24的解压版

下载地址：<https://downloads.mysql.com/archives/community/>

1. 如图所示



1. 在mysql根目录添加配置文件my.ini；改成自己的安装目录，并且一定要新建data目录；



my.ini文件内容

|  |
| --- |
| [mysql]  # 设置mysql客户端默认字符集  default-character-set=utf8  [mysqld]  #设置3306端口  port = 3306  # 设置mysql的安装目录  basedir=E:\mysql-5.7.24-winx64  # 设置mysql数据库的数据的存放目录  datadir=E:\mysql-5.7.24-winx64\data  # 允许最大连接数  max\_connections=200  # 服务端使用的字符集默认为8比特编码的latin1字符集  character-set-server=utf8  # 创建新表时将使用的默认存储引擎  default-storage-engine=INNODB |

1. 安装并启动服务

|  |
| --- |
| cmd用管理员身份进入mysql的bin目录 1.初始化数据库（第一次一定要做，后续如果卸载后重新安装则跳过这一步） mysqld --initialize 2.安装服务 mysqld install MySQL57 3.启动服务 net start MySQL57 |

1. 配置数据库

|  |
| --- |
| 配置数据库做两件事情：修改root默认密码；设置root可以远程访问； 1.登陆：mysql -uroot -p 初始密码在data文件夹中以.err结尾的文件中，如：A temporary password is generated for root@localhost: **wEsWp:pvB8Py** 2.修改默认密码：alter user root@localhost identified by '123456'; 3.切换数据库：use mysql; 3.设置用户可以远程访问（%表示不限制访问ip）：update user set host = '%' where user = 'root'; 4.确认设置（或重启服务器让设置生效）：FLUSH PRIVILEGES; |

以上变完成了mysql服务端的安装

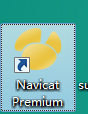
Mysql详细使用教程请参考尚硅谷官方免费教程

<http://www.atguigu.com/download_detail.shtml?v=3>

### 1.6、数据库连接工具

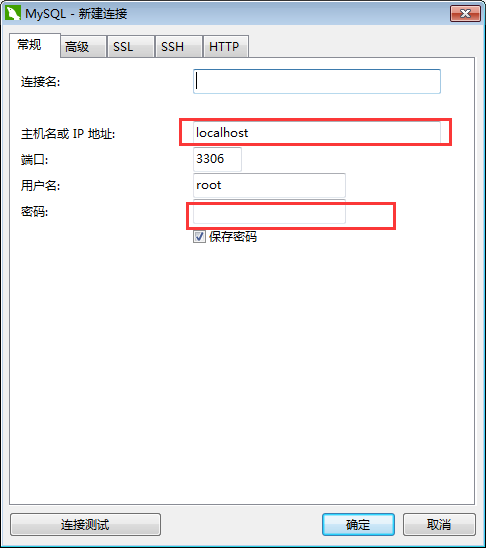
推荐使用Navicat

图标如图所示



安装完成连接mysql

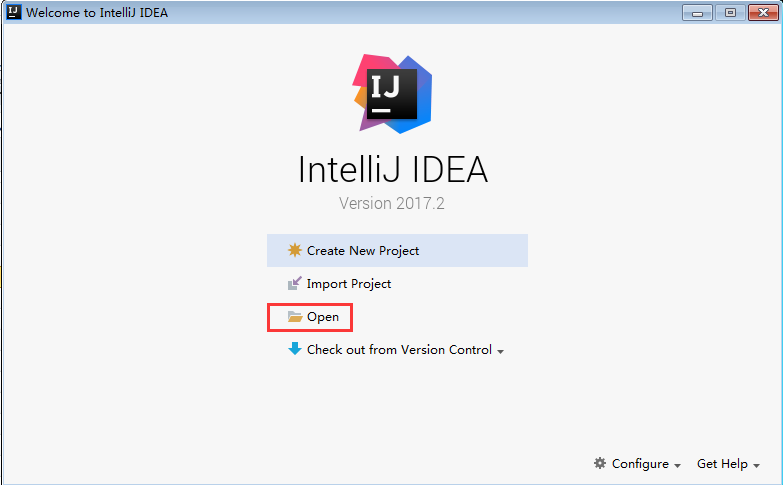
1、输入连接ip地址，用户名和密码，如果连接成功，则表示mysql安装没有问题



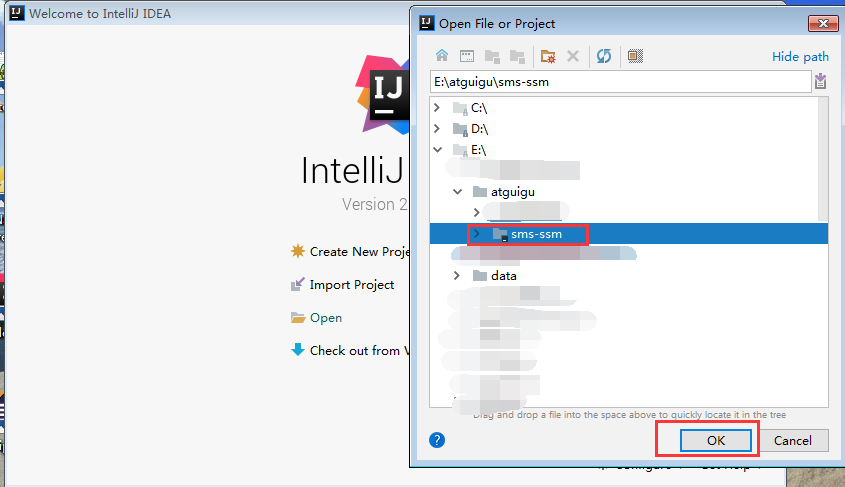
## 项目运行

### 2.1、Idea，tomcat，代码配置

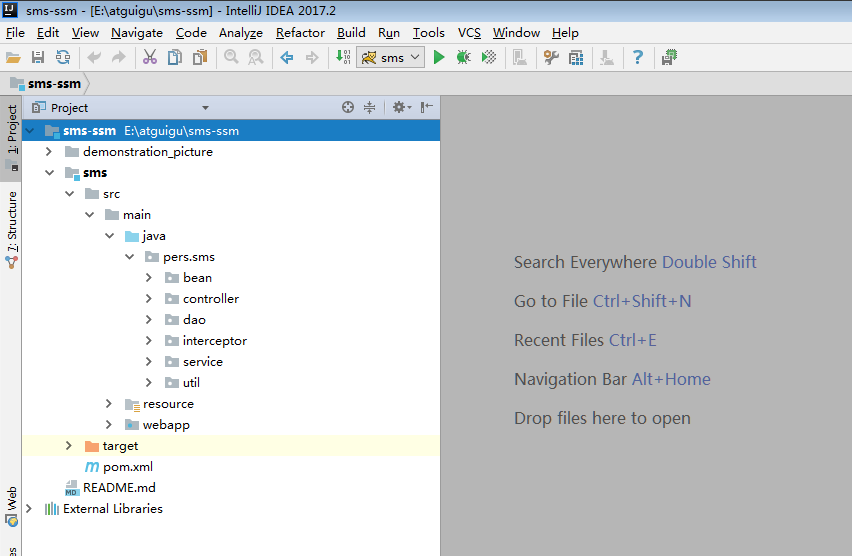
1. 打开idea，选择open，如图所示



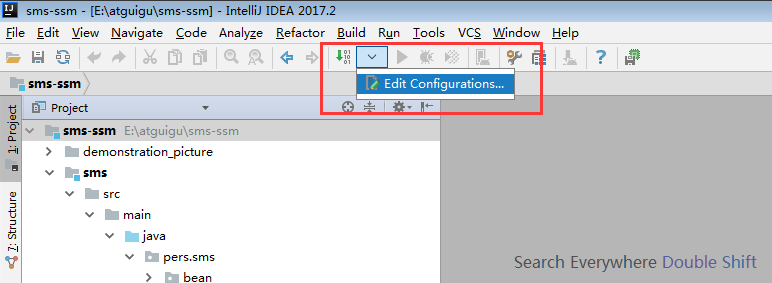
1. 选中项目文件夹，点击ok按钮如图所示



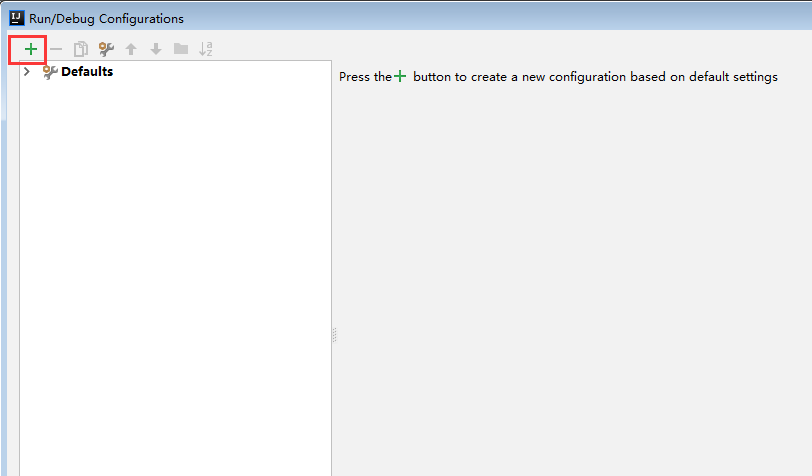
1. 进入idea界面，如图所示



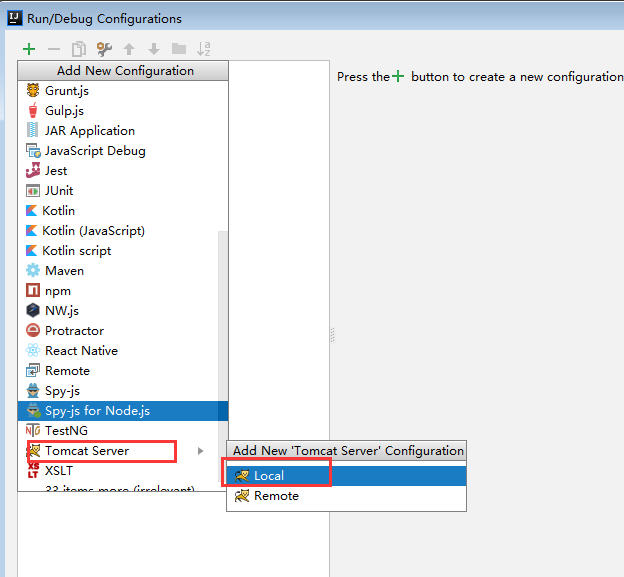
1. 配置tomcat，如图所示，点击Add Configuration



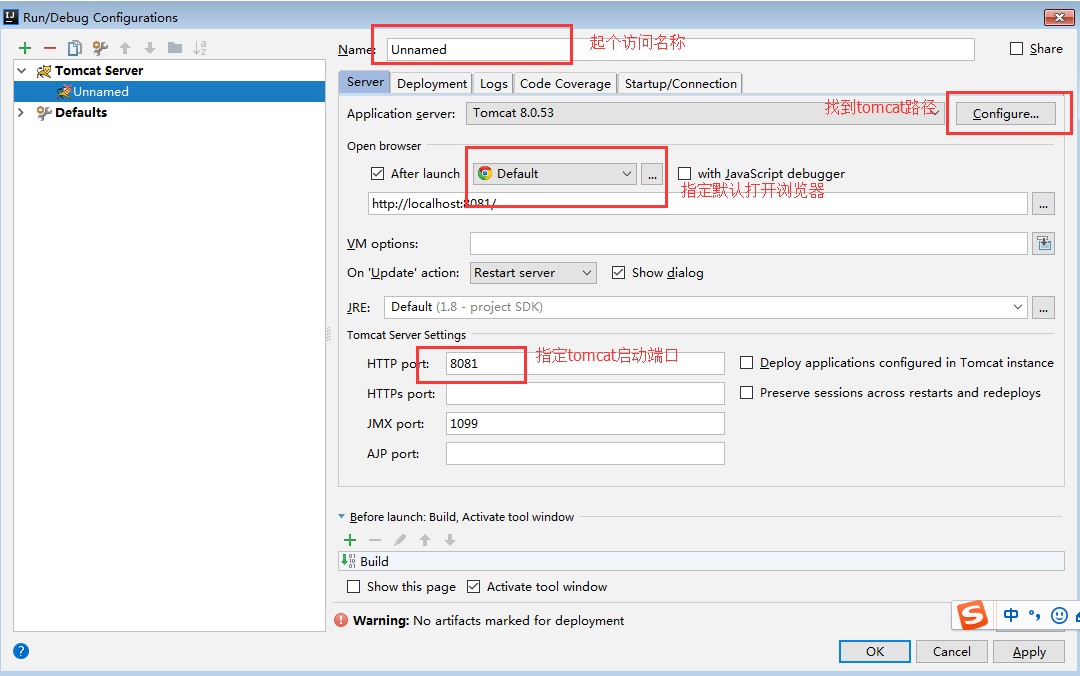
1. 点击+号按钮，如图所示



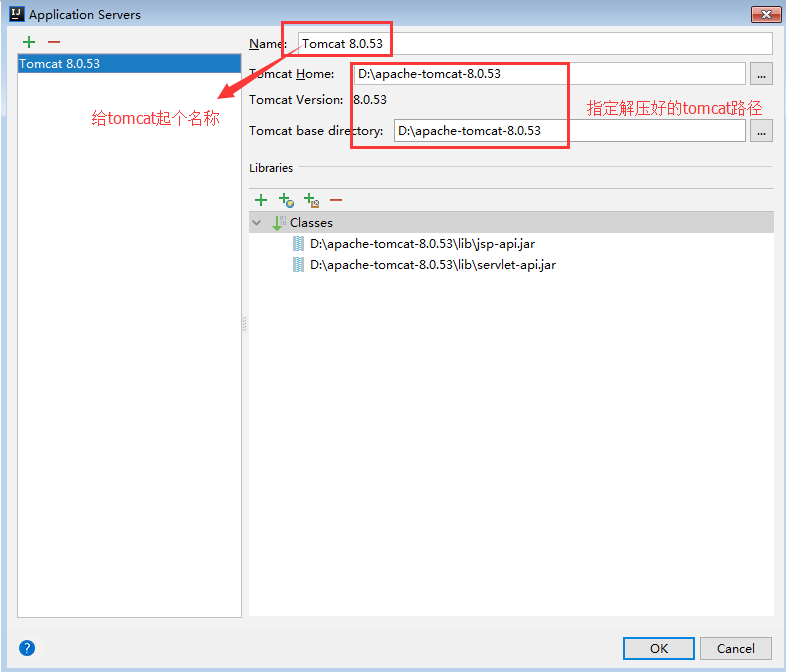
1. 选中tomcat->local，如图所示



1. 配置tomcat，如图所示

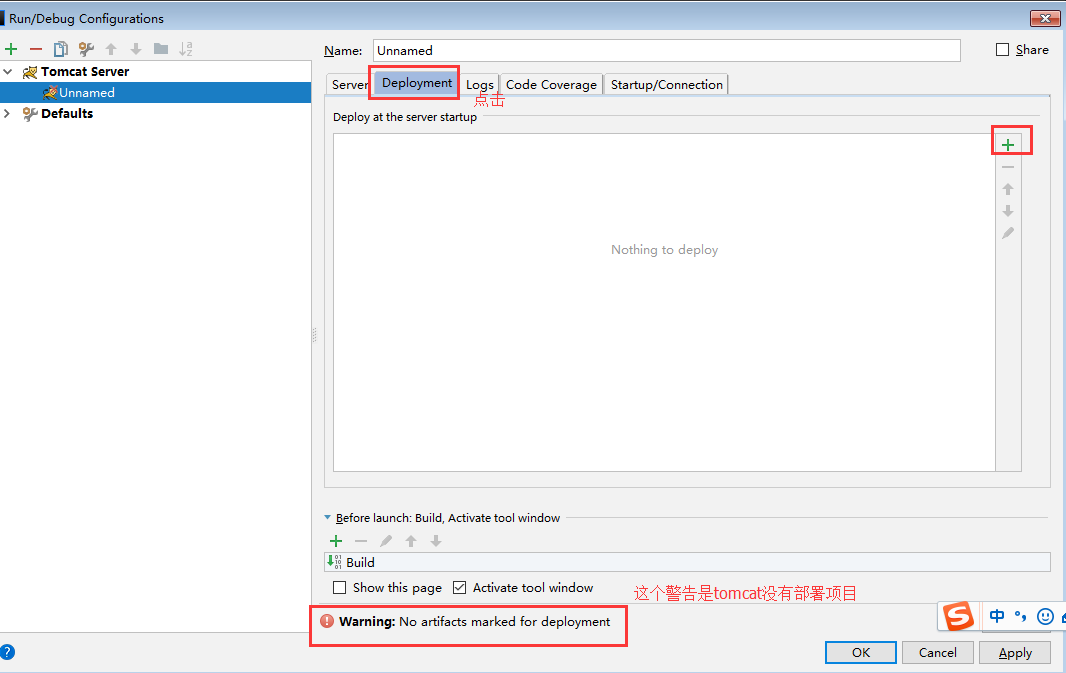


1. 点击configure，如图所示，

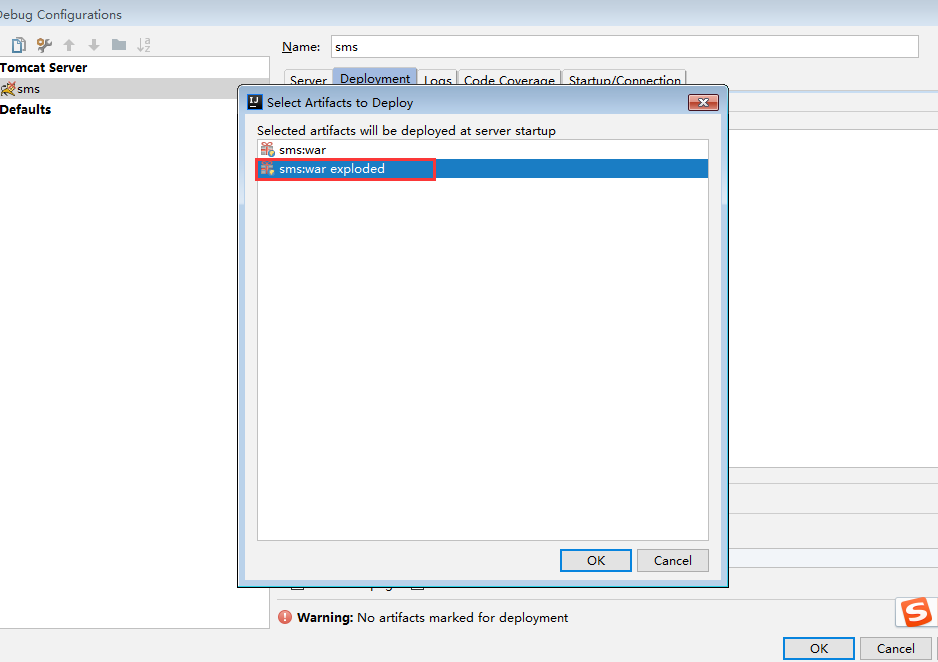


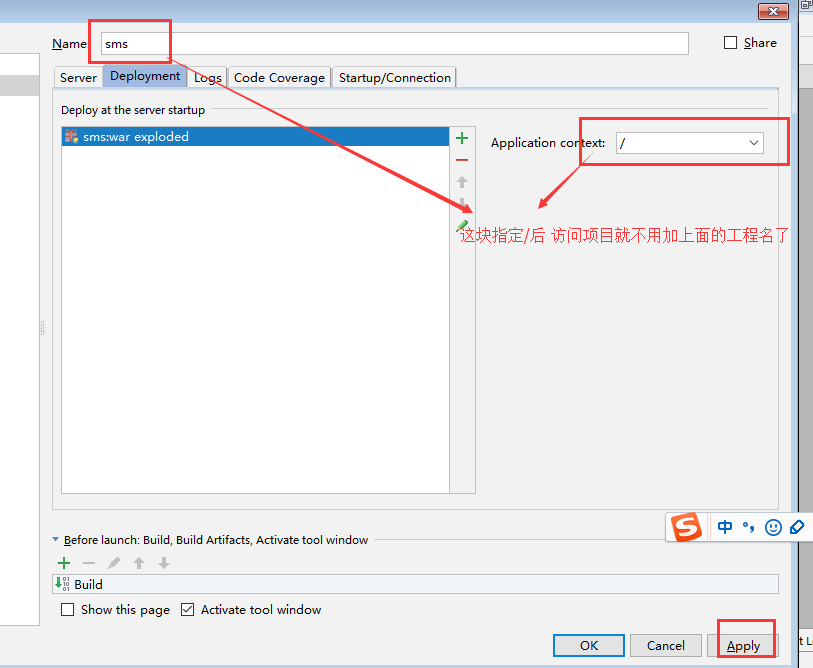
只要选中我们自己电脑上tomcat目录即可

1. 添加artifict，选中+号，如图所示



1. 选中explored.war即可，如图所示，点击确定

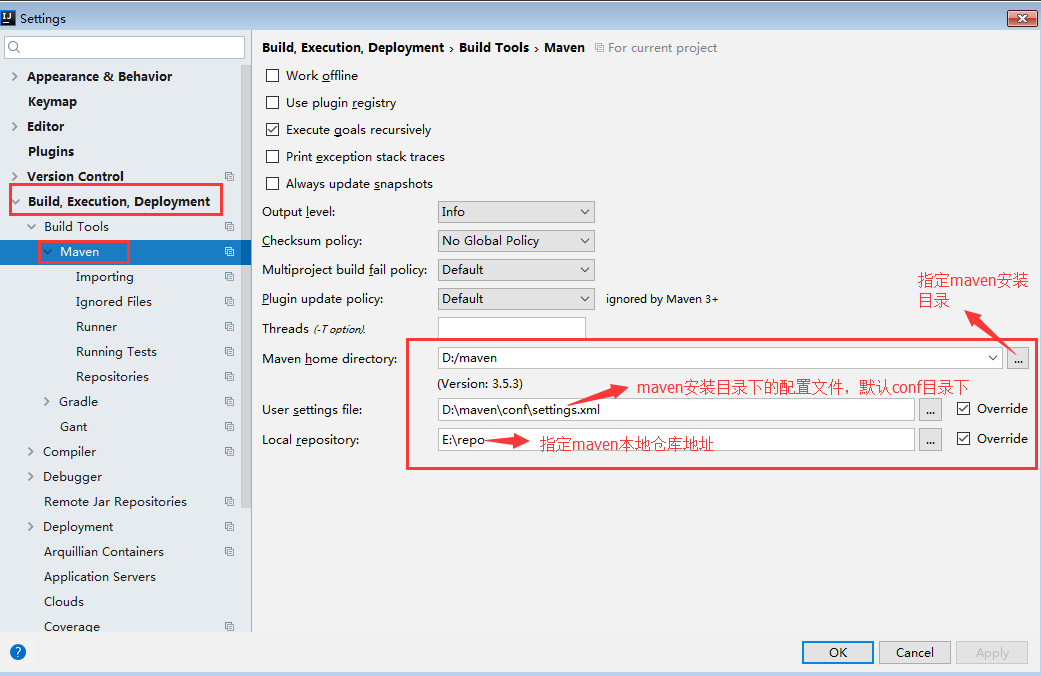




修改为图上的两个选项，表示更新修改的文件，主要作用于前端，点击右下角apply即完成服务器配置。

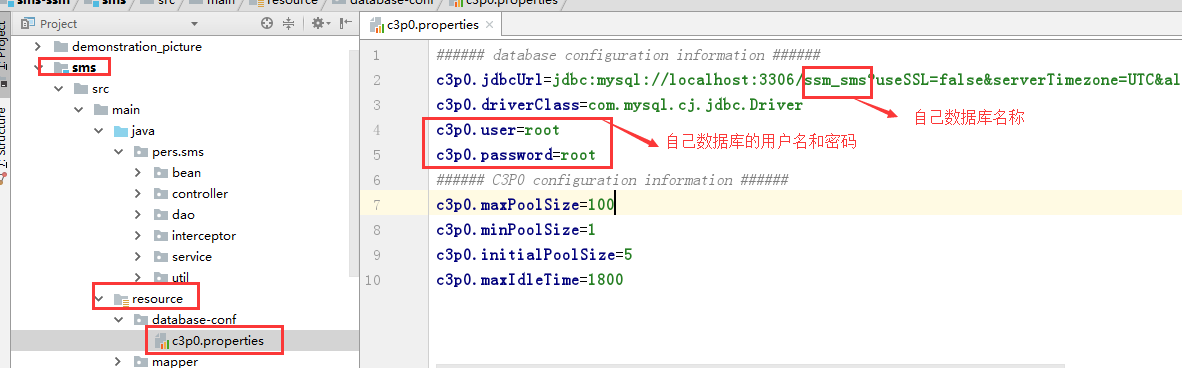
### 2.2、maven配置

打开idea->file->settings配置，选中maven安装目录，点击确认，如图所示



### 2.3、修改mysql配置

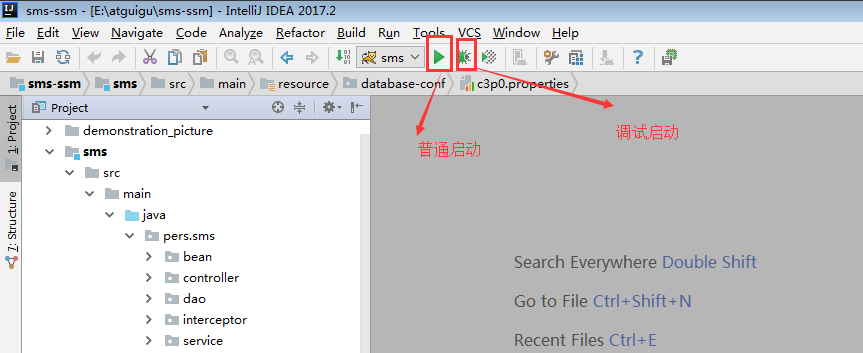
打开c3p0.properties配置文件，修改数据库连接地址，用户名和密码，保存即可



### 2.4、项目启动

两种启动方式：

1. 直接启动
2. Debug启动，建议使用debug启动



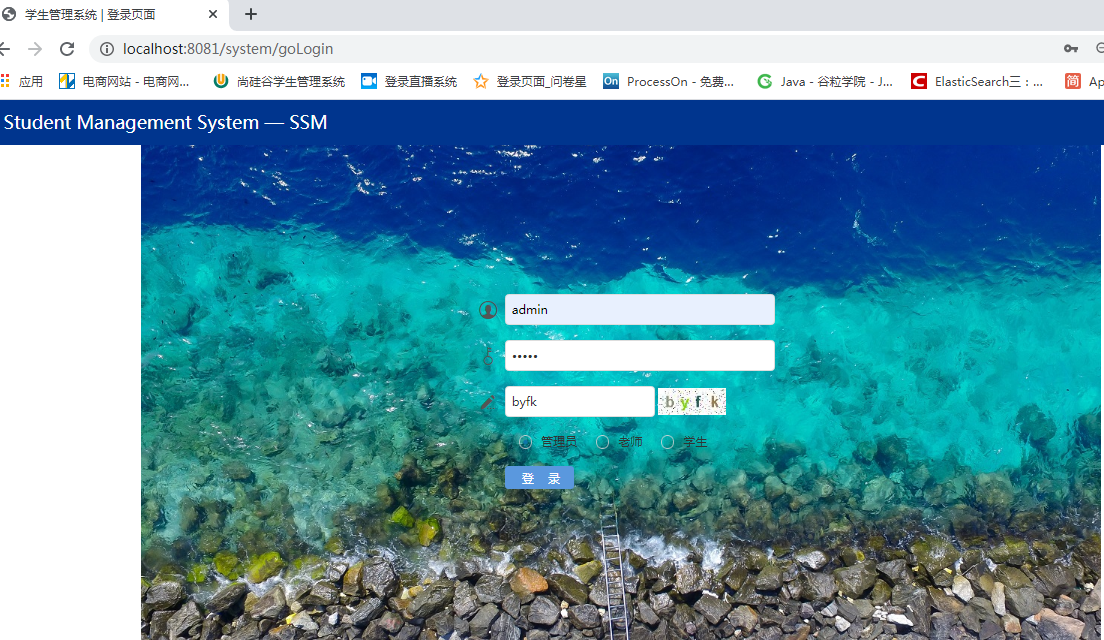
### 2.5、启动成功

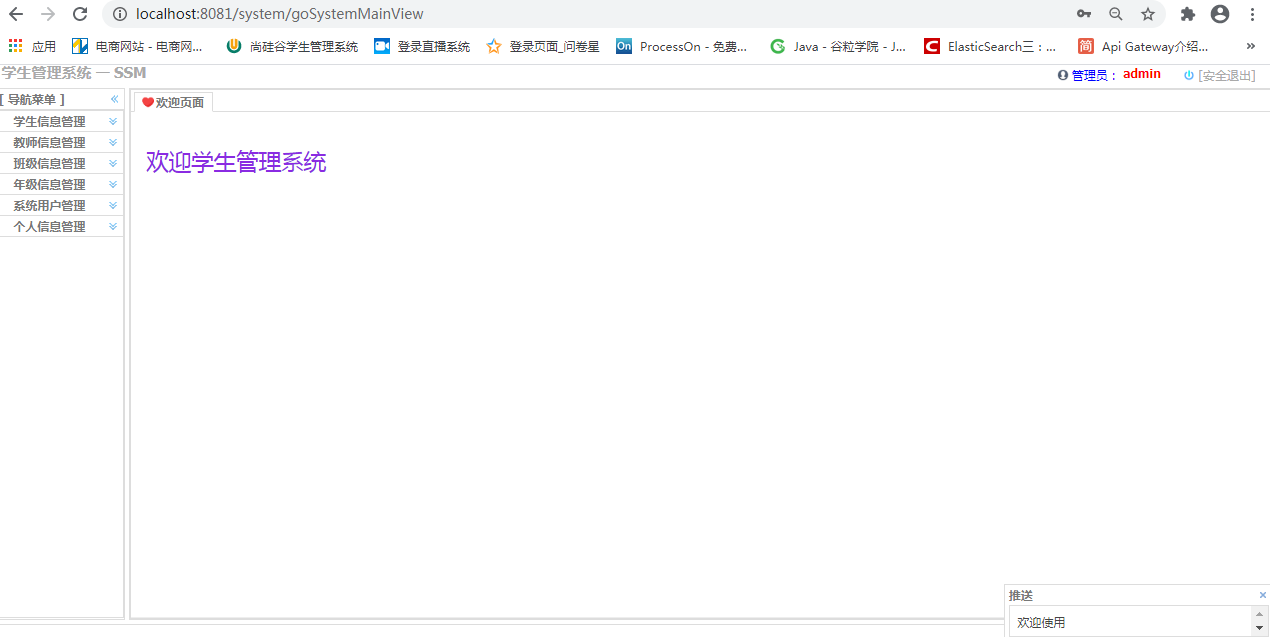
效果如图所示

访问地址：<http://localhost:8080>

登录名：admin

密码：admin

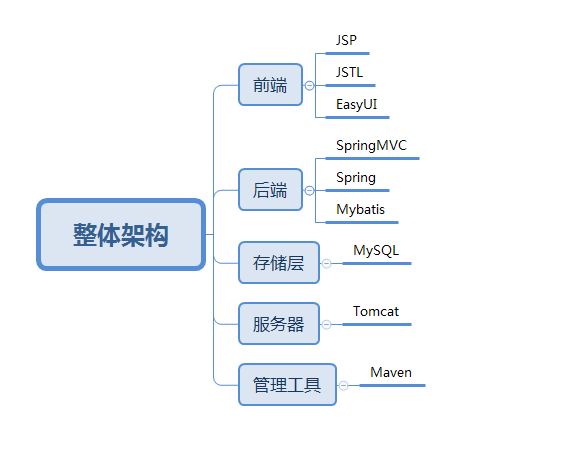




## 项目架构

### 3.1、整体架构图：

如图所示，项目前端采用JSP+JSTL+EasyUI来实现，后端服务采用SpringMVC+Spring+Mybatis框架。存储层使用高性能的MySQL，服务使用的是Tomcat服务器，使用Maven来管理jar包和项目构建。



### 3.2、后端架构：

**Spring**  
　　Spring就像是整个项目中装配bean的大工厂，在配置文件中可以指定使用特定的参数去调用实体类的构造方法来实例化对象。也可以称之为项目中的粘合剂。  
　　Spring的核心思想是IoC（控制反转），即不再需要程序员去显式地`new`一个对象，而是让Spring框架帮你来完成这一切。  
**SpringMVC**  
　　SpringMVC在项目中拦截用户请求，它的核心Servlet即DispatcherServlet承担中介或是前台这样的职责，将用户请求通过HandlerMapping去匹配Controller，Controller就是具体对应请求所执行的操作。SpringMVC相当于SSH框架中struts。  
**mybatis**  
　　mybatis是对jdbc的封装，它让数据库底层操作变的透明。mybatis的操作都是围绕一个sqlSessionFactory实例展开的。mybatis通过配置文件关联到各实体类的Mapper文件，Mapper文件中配置了每个类对数据库所需进行的sql语句映射。在每次与数据库交互时，通过sqlSessionFactory拿到一个sqlSession，再执行sql命令。

页面发送请求给控制器，控制器调用业务层处理逻辑，逻辑层向持久层发送请求，持久层与数据库交互，后将结果返回给业务层，业务层将处理逻辑发送给控制器，控制器再调用视图展现数据。

### 3.3、管理工具：

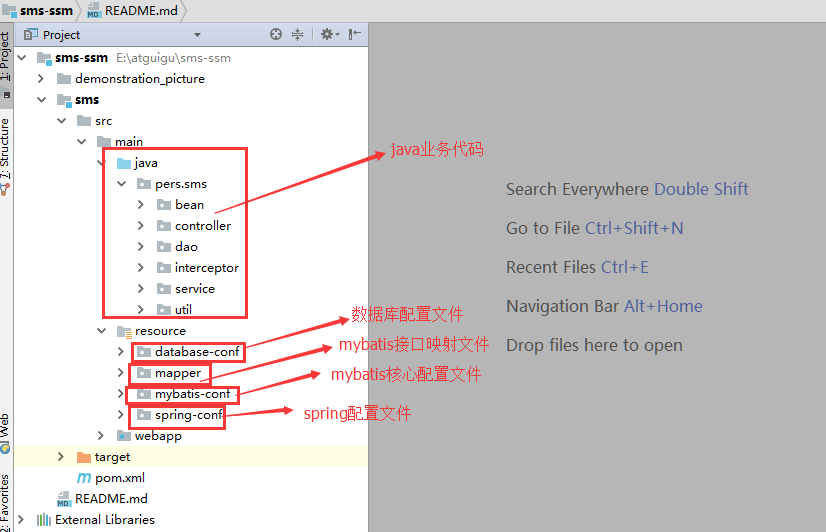
采用maven来进行项目构建，管理，jar包下载等功能。

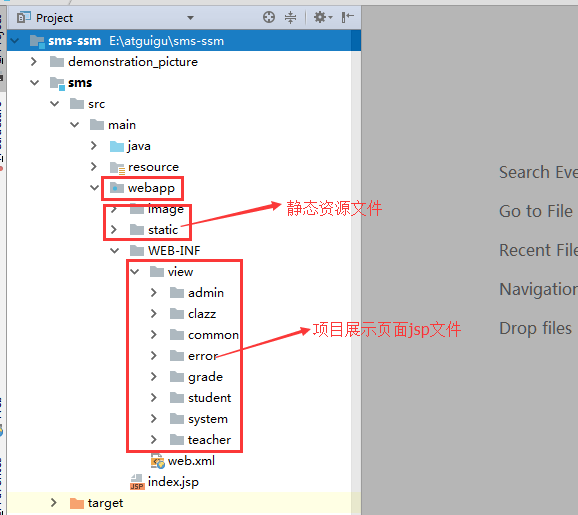
下面是引入的jar包管理

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd"**>  <**modelVersion**>4.0.0</**modelVersion**>   <**groupId**>io.github.yubuntu0109</**groupId**>  <**artifactId**>sms</**artifactId**>  <**version**>1.0-SNAPSHOT</**version**>  <**packaging**>war</**packaging**>   <**name**>sms Maven Webapp</**name**>  *<!--* ***FIXME change it to the project's website*** *-->* <**url**>http://www.example.com</**url**>   <**properties**>  <**project.build.sourceEncoding**>UTF-8</**project.build.sourceEncoding**>  <**project.build.spring.webmvc.version**>5.1.7.RELEASE</**project.build.spring.webmvc.version**>  <**project.build.spring.webmvc.file.version**>1.4</**project.build.spring.webmvc.file.version**>  <**project.build.spring.webmvc.json.version**>2.9.10</**project.build.spring.webmvc.json.version**>  <**project.build.spring.jdbc.version**>5.1.7.RELEASE</**project.build.spring.jdbc.version**>  <**project.build.mybaits.version**>3.5.1</**project.build.mybaits.version**>  <**project.build.myabtis.pagehelper.version**>5.1.9</**project.build.myabtis.pagehelper.version**>  <**project.build.mybatis.spring.version**>2.0.1</**project.build.mybatis.spring.version**>  <**project.build.servlet.version**>4.0.1</**project.build.servlet.version**>  <**project.build.jstl.version**>1.2</**project.build.jstl.version**>  <**project.build.mysql.jdbc.version**>8.0.16</**project.build.mysql.jdbc.version**>  <**project.build.mysql.c3p0.version**>0.9.5.4</**project.build.mysql.c3p0.version**>  <**project.build.logging.version**>1.2</**project.build.logging.version**>  <**maven.compiler.source**>1.8</**maven.compiler.source**>  <**maven.compiler.target**>1.8</**maven.compiler.target**>  </**properties**>   <**dependencies**>  *<!-- spring mvc -->* <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-webmvc</**artifactId**>  <**version**>${project.build.spring.webmvc.version}</**version**>  </**dependency**>  *<!-- Spring mvc:JSON -->* <**dependency**>  <**groupId**>com.fasterxml.jackson.core</**groupId**>  <**artifactId**>jackson-databind</**artifactId**>  <**version**>${project.build.spring.webmvc.json.version}</**version**>  </**dependency**>  *<!-- Spring mvc:file -->* <**dependency**>  <**groupId**>commons-fileupload</**groupId**>  <**artifactId**>commons-fileupload</**artifactId**>  <**version**>${project.build.spring.webmvc.file.version}</**version**>  </**dependency**>  *<!-- Spring JDBC -->* <**dependency**>  <**groupId**>org.springframework</**groupId**>  <**artifactId**>spring-jdbc</**artifactId**>  <**version**>${project.build.spring.jdbc.version}</**version**>  </**dependency**>  *<!-- MyBatis -->* <**dependency**>  <**groupId**>org.mybatis</**groupId**>  <**artifactId**>mybatis</**artifactId**>  <**version**>${project.build.mybaits.version}</**version**>  </**dependency**>  *<!-- MyBatis 分页插件 -->* <**dependency**>  <**groupId**>com.github.pagehelper</**groupId**>  <**artifactId**>pagehelper</**artifactId**>  <**version**>${project.build.myabtis.pagehelper.version}</**version**>  </**dependency**>  *<!-- Mybatis与Spring的整合包 -->* <**dependency**>  <**groupId**>org.mybatis</**groupId**>  <**artifactId**>mybatis-spring</**artifactId**>  <**version**>${project.build.mybatis.spring.version}</**version**>  </**dependency**>  *<!-- Servelt API -->* <**dependency**>  <**groupId**>javax.servlet</**groupId**>  <**artifactId**>javax.servlet-api</**artifactId**>  <**version**>${project.build.servlet.version}</**version**>  <**scope**>provided</**scope**>  </**dependency**>  *<!-- JSTL -->* <**dependency**>  <**groupId**>javax.servlet</**groupId**>  <**artifactId**>jstl</**artifactId**>  <**version**>${project.build.jstl.version}</**version**>  </**dependency**>  *<!-- 数据库驱动 -->* <**dependency**>  <**groupId**>mysql</**groupId**>  <**artifactId**>mysql-connector-java</**artifactId**>  <**version**>${project.build.mysql.jdbc.version}</**version**>  </**dependency**>  *<!-- 数据库连接池 -->* <**dependency**>  <**groupId**>com.mchange</**groupId**>  <**artifactId**>c3p0</**artifactId**>  <**version**>${project.build.mysql.c3p0.version}</**version**>  </**dependency**>  *<!-- 日志组件 -->* <**dependency**>  <**groupId**>commons-logging</**groupId**>  <**artifactId**>commons-logging</**artifactId**>  <**version**>${project.build.logging.version}</**version**>  </**dependency**>  </**dependencies**>   <**build**>  <**finalName**>sms</**finalName**>  <**pluginManagement**><**plugins**>  <**plugin**>  <**artifactId**>maven-clean-plugin</**artifactId**>  <**version**>3.1.0</**version**>  </**plugin**><**plugin**>  <**artifactId**>maven-resources-plugin</**artifactId**>  <**version**>3.0.2</**version**>  </**plugin**>  <**plugin**>  <**artifactId**>maven-compiler-plugin</**artifactId**>  <**version**>3.8.0</**version**>  </**plugin**>  <**plugin**>  <**artifactId**>maven-surefire-plugin</**artifactId**>  <**version**>2.22.1</**version**>  </**plugin**>  <**plugin**>  <**artifactId**>maven-war-plugin</**artifactId**>  <**version**>3.2.2</**version**>  </**plugin**>  <**plugin**>  <**artifactId**>maven-install-plugin</**artifactId**>  <**version**>2.5.2</**version**>  </**plugin**>  <**plugin**>  <**artifactId**>maven-deploy-plugin</**artifactId**>  <**version**>2.8.2</**version**>  </**plugin**>  </**plugins**>  </**pluginManagement**>  </**build**> </**project**> |

## 项目功能使用介绍

### 4.1、项目工程结构说明



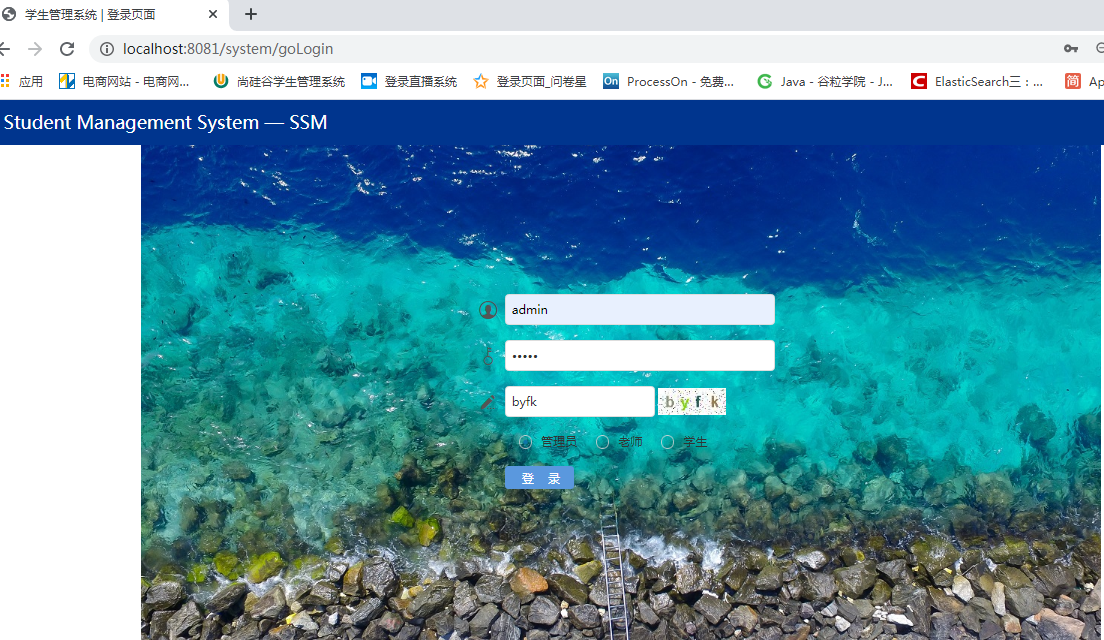


### 4.2、后台用户登录

#### 4.2.1、功能演示

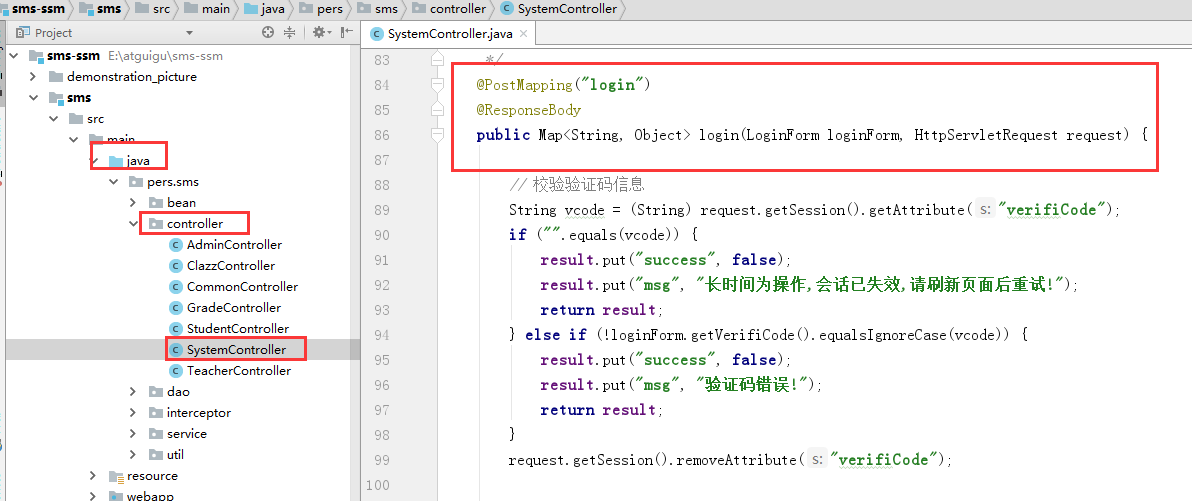
用户访问 <http://localhost:8081/> 默认用户名：admin 密码：admin

在登录页面用户输入用户名、密码和验证码点击登录后跳转首页，如图所示：



#### 4.2.2、代码实现

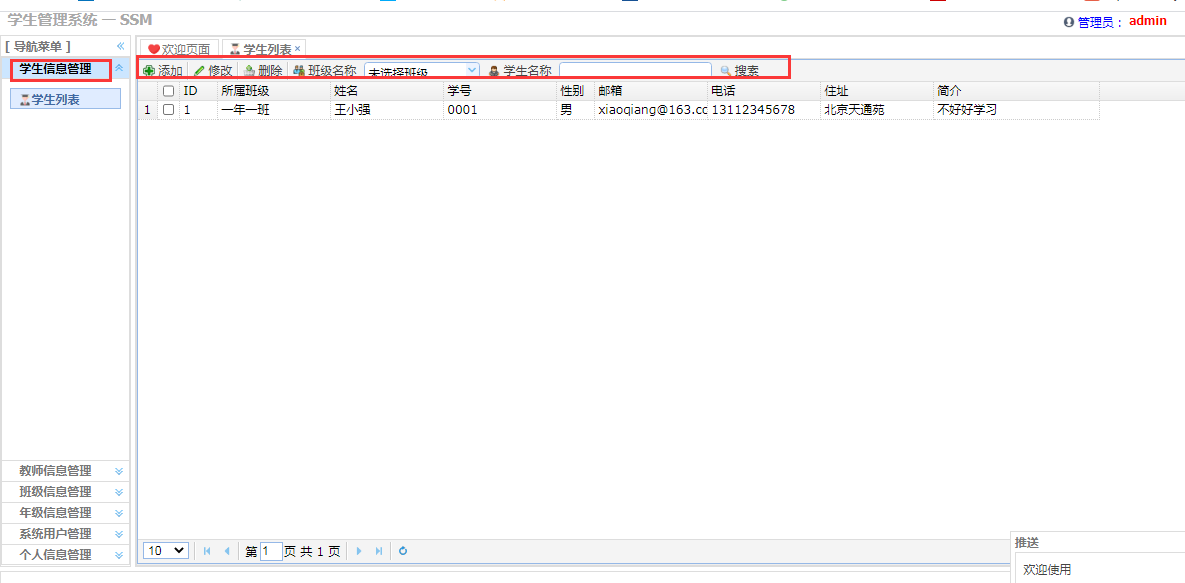
用户登录的代码位置如下图所示



### 4.3、学生信息管理模块

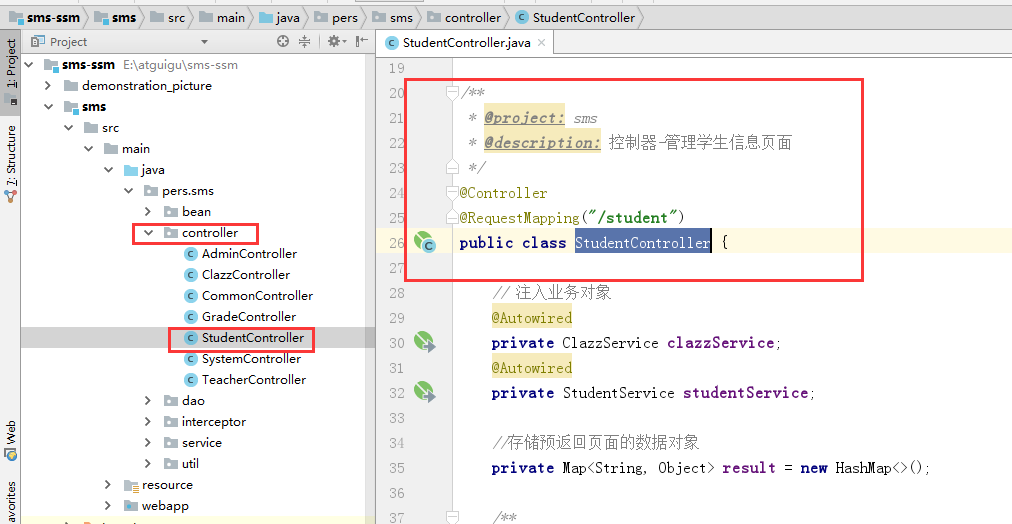
#### 4.3.1、功能演示

主要是学生信息进行管理的功能，以列表的形式展示学生信息，可以进行学生添加、修改、删除以及多条件查询学生等功能。



#### 4.3.2、代码实现

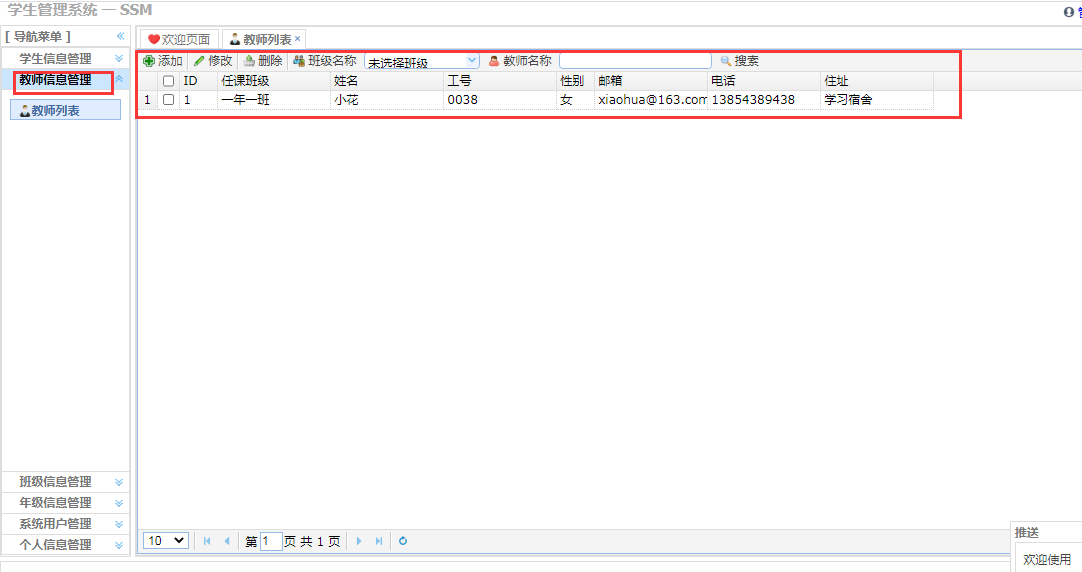
请求StudentController，通过controller调用service进行业务处理。



### 4.4、教师信息管理模块

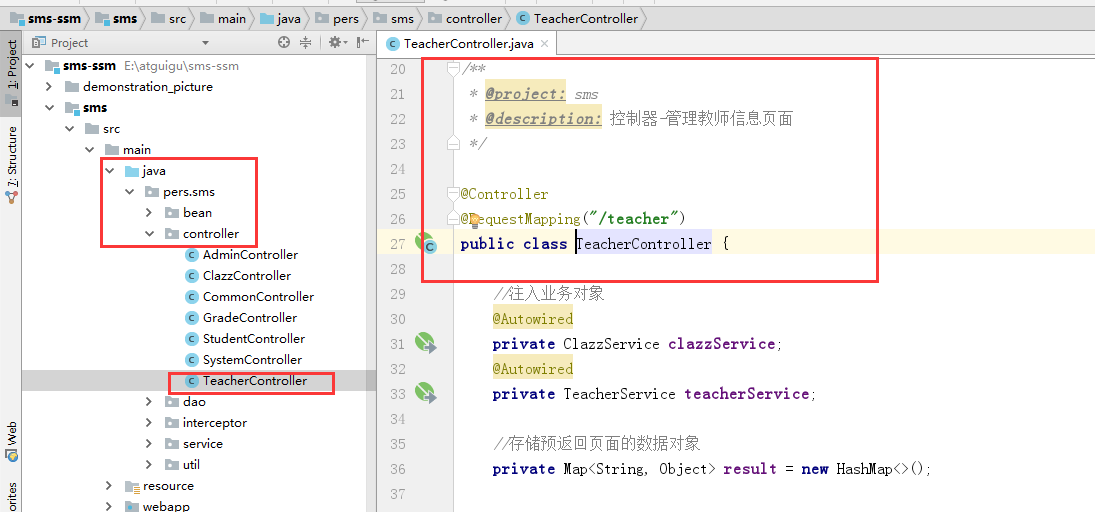
#### 4.4.1、功能演示

主要是教师进行管理的功能，以列表的形式展示教师信息，可以进行教师添加、修改、删除以及多条件查询等功能。



#### 4.4.2、代码实现

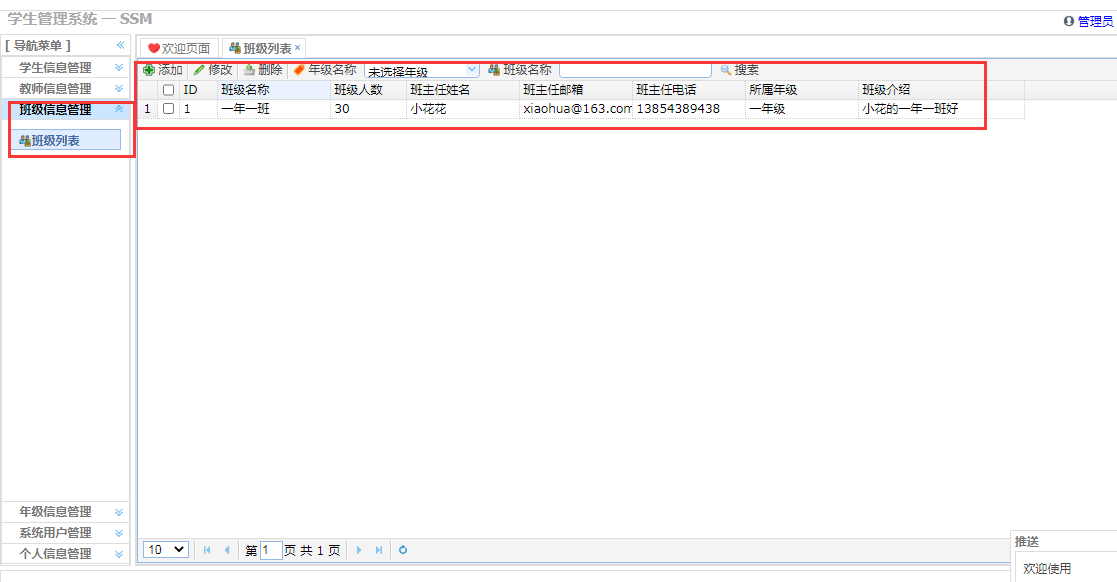
请求TeacherController，通过controller调用service进行业务处理。



### 4.5、班级信息管理模块

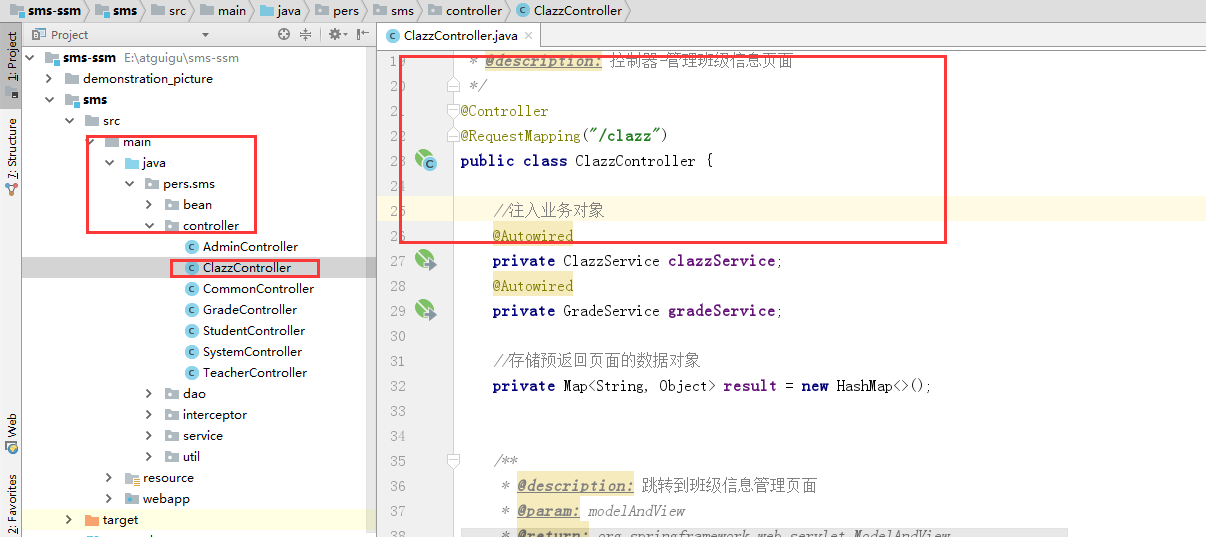
#### 4.5.1、功能演示

主要是班级信息进行管理的功能，以列表的形式展示班级信息，可以进行班级添加、修改、删除以及多条件查询等功能。



#### 4.5.2、代码实现

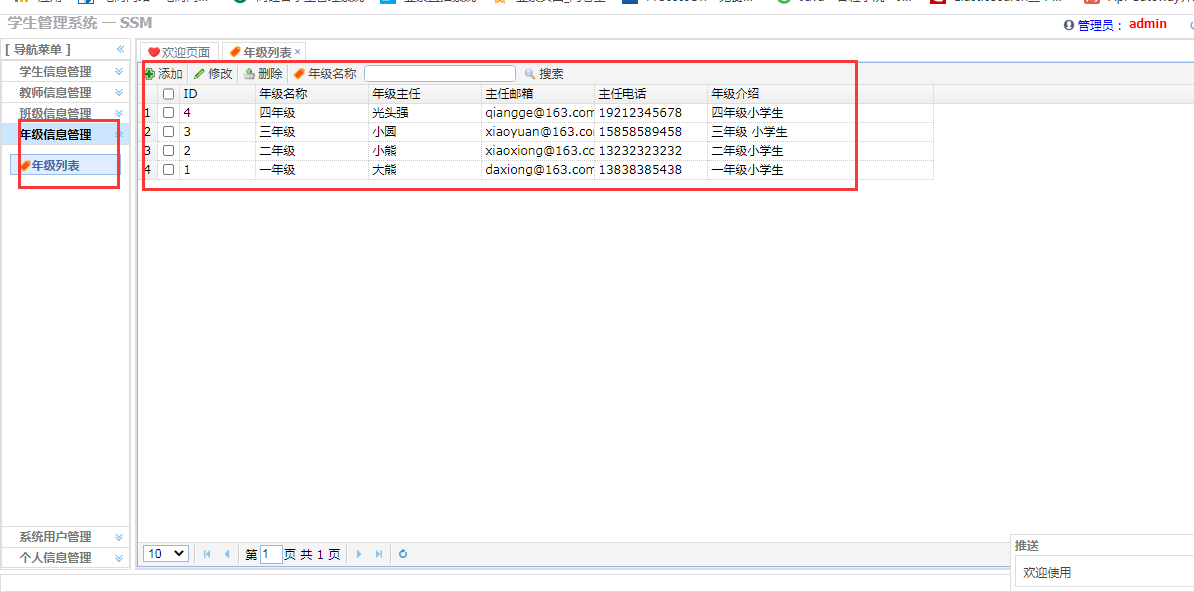
班级管理的代码位置如下图所示



### 4.6、年级信息管理模块

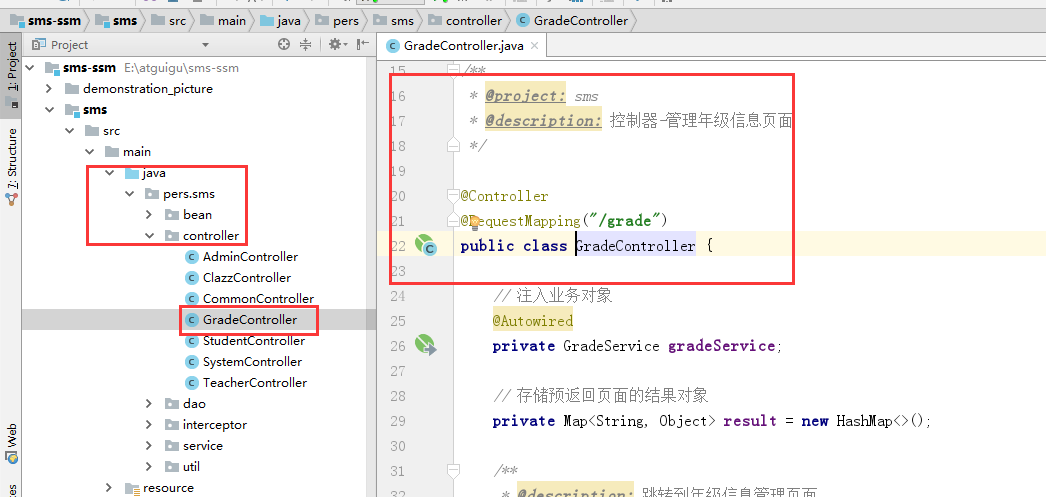
#### 4.6.1、功能演示

主要是年级信息进行管理的功能，以列表的形式展示年级信息，可以进行年级添加、修改、删除以及多条件查询等功能。



#### 4.6.2、代码实现

年级信息管理的代码位置如下图所示

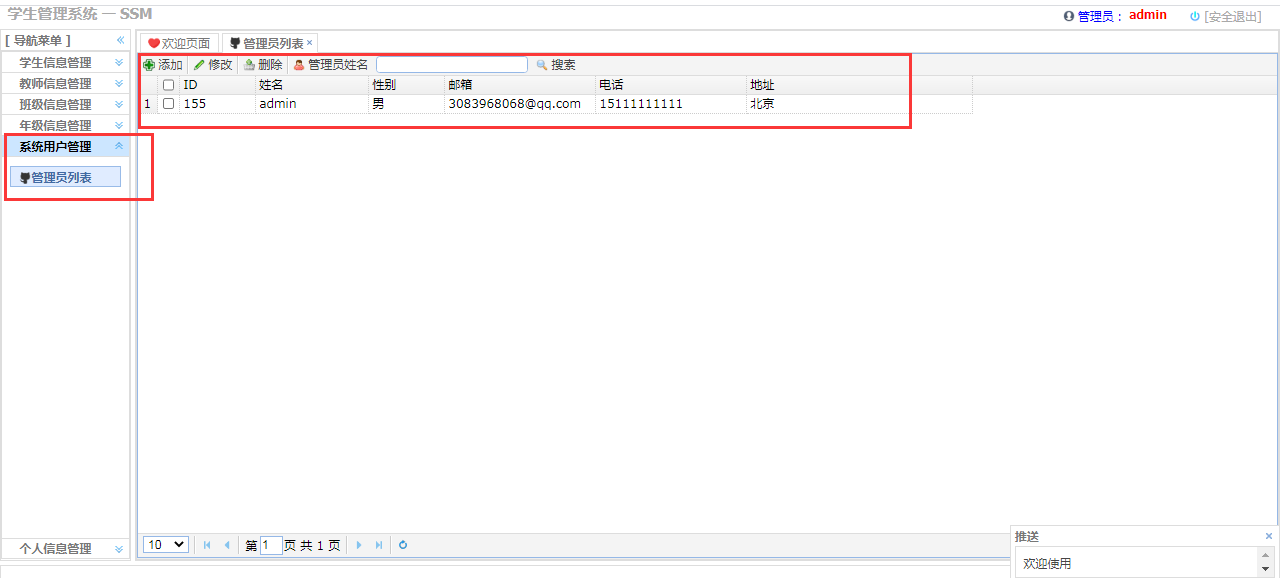


### 4.7、系统用户管理模块

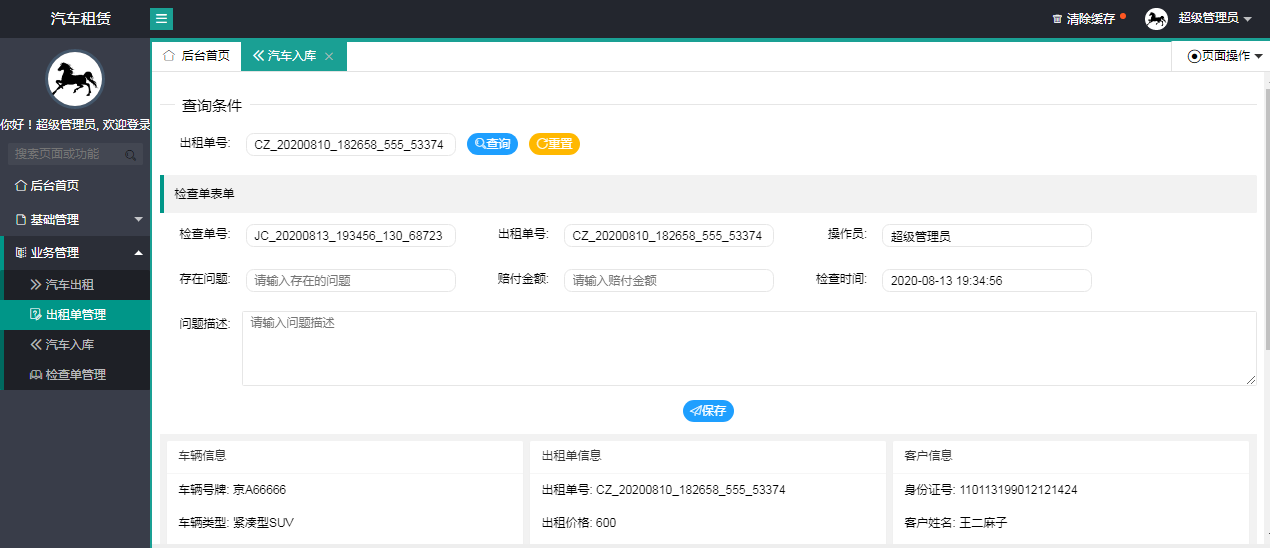
#### 4.7.1、功能演示

主要是管理员信息进行管理的功能，以列表的形式展示管理员信息，可以进行管理员添加、修改、删除以及多管理员名称查询等功能。

。



在此页面填写车辆存在问题、问题描述以及赔付金额信息，进行检查单保存。



#### 4.7.2、代码实现

系统用户管理的代码位置如下图所示

