

Lab 1.1 Web Environment Setup-Java/Tomcat/Eclipse

姓名：应承峻

学号：3170103456

Lab Topic

Web programming is a really big topic, which is related to a series of work. But In this lab, the only thing you have to do is: learn how to setup the web environment, include Java, Tomcat and Eclipse. It is the very base of all our further work about Web.

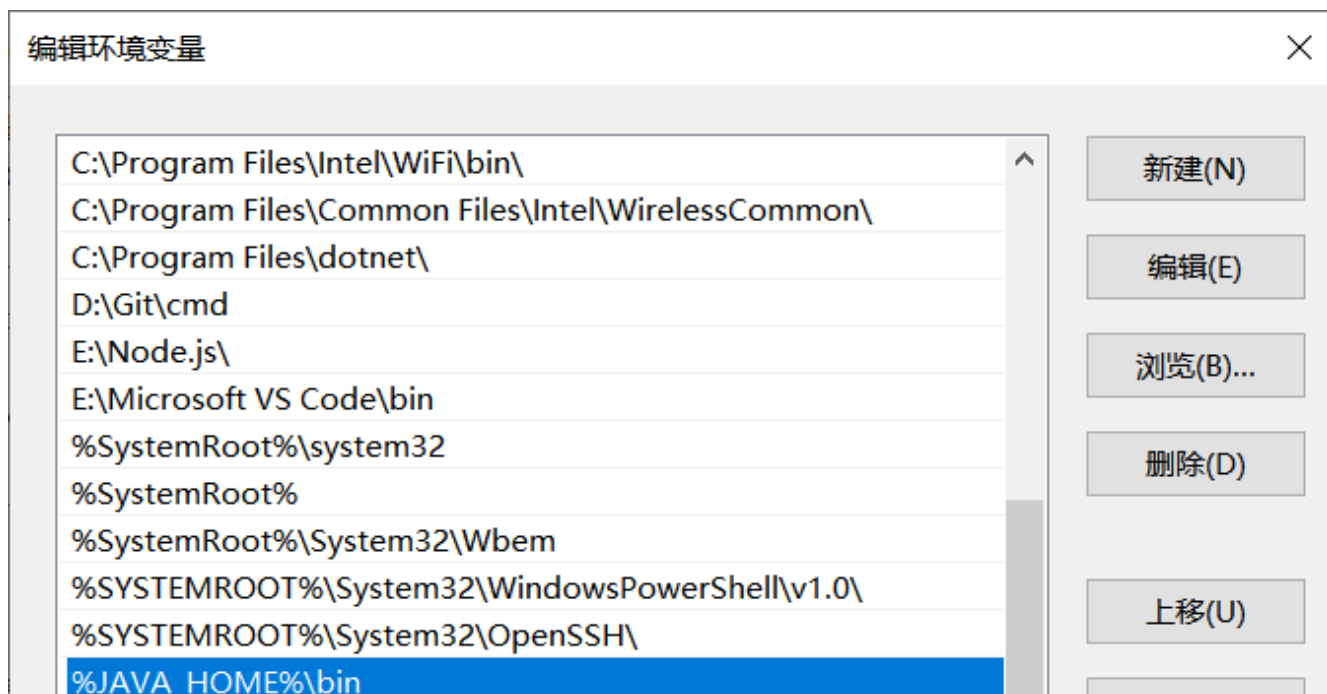
Before you start, you should choose the OS platform first, windows or linux. Actually, both platforms are suitable, just up to you!

Back to the lab, what's our objective:

1. Setup the Java, and make sure it's correct.
2. Setup the tomcat, and check the tomcat welcome page.
3. Setup the Eclipse, and learn how to use it.

Step 1:

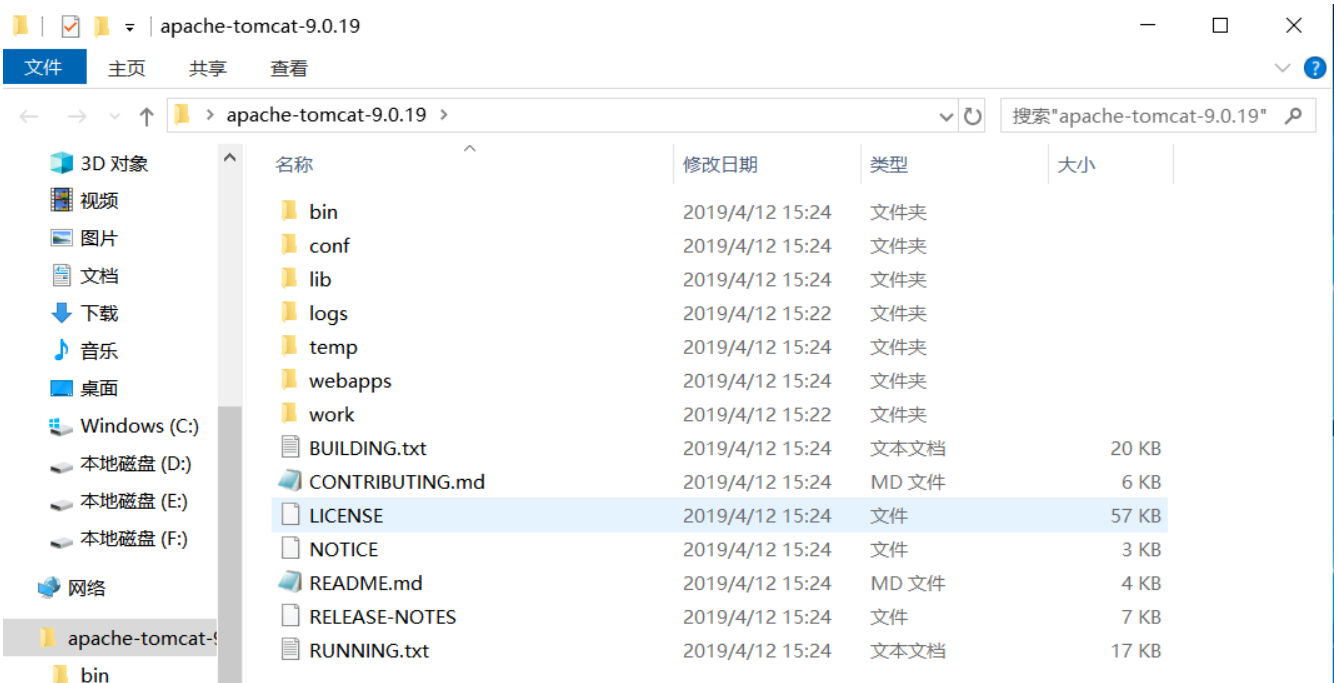
下载与安装JAVA，由于本电脑此前已经安装过JAVA，安装的主要麻烦之处主要在于环境变量的配置。因此在这仅做环境变量的配置结果与版本测试结果：



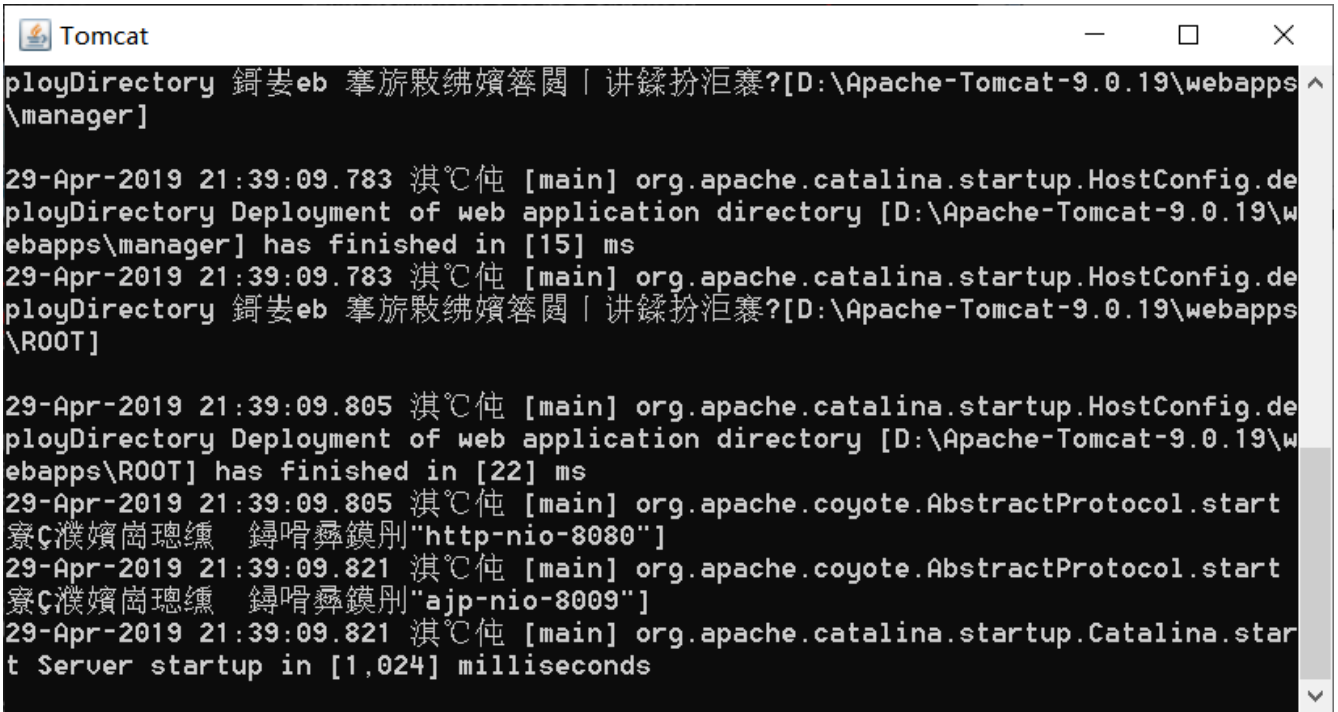
```
Windows PowerShell
PS C:\Users\YingChengJun\Desktop> java -version
java version "10.0.2" 2018-07-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.2+13)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.2+13, mixed mode)
PS C:\Users\YingChengJun\Desktop>
```

Step 2:

在Tomcat官网上下载安装包后解压:



运行bin目录下的startup.bat文件, 下图说明了tomcat能正常运行, 但是存在乱码的问题:

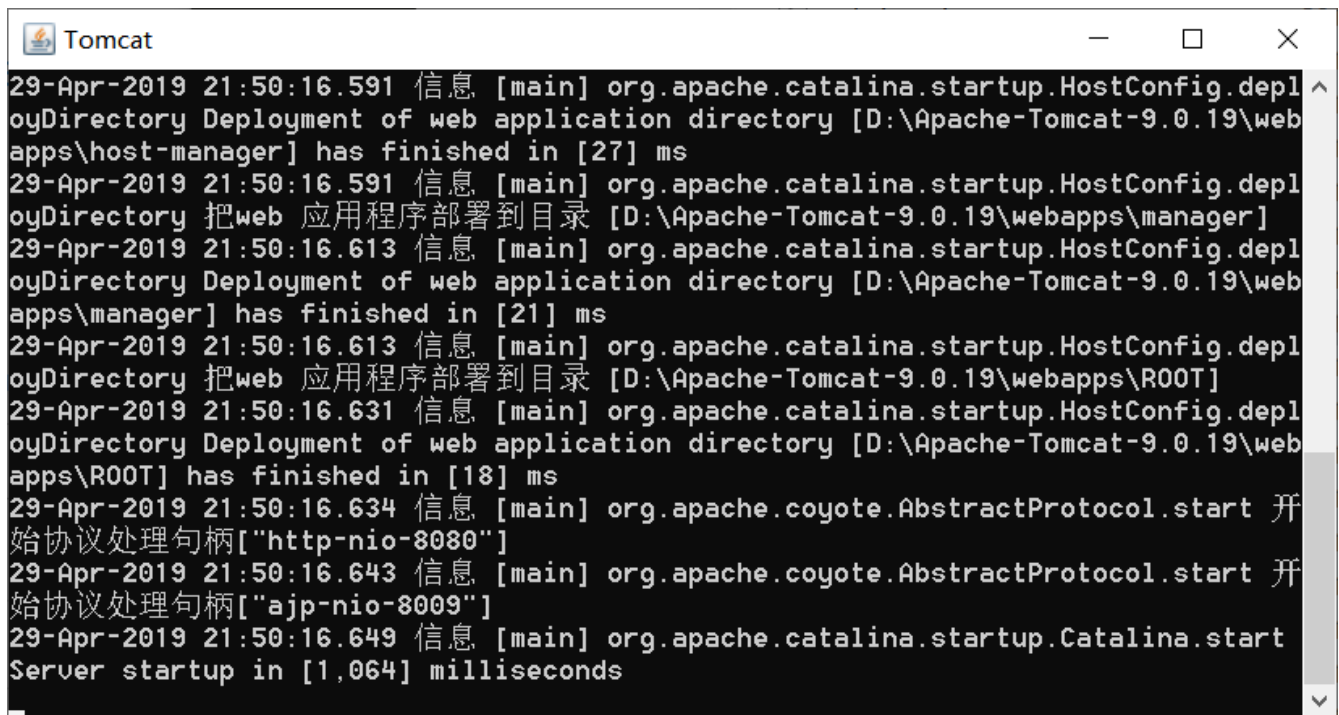


参考了https://blog.csdn.net/New_Yao/article/details/86607071 这篇文档后，将tomcat/conf/logging.properties 中做如下图所示更改即可：

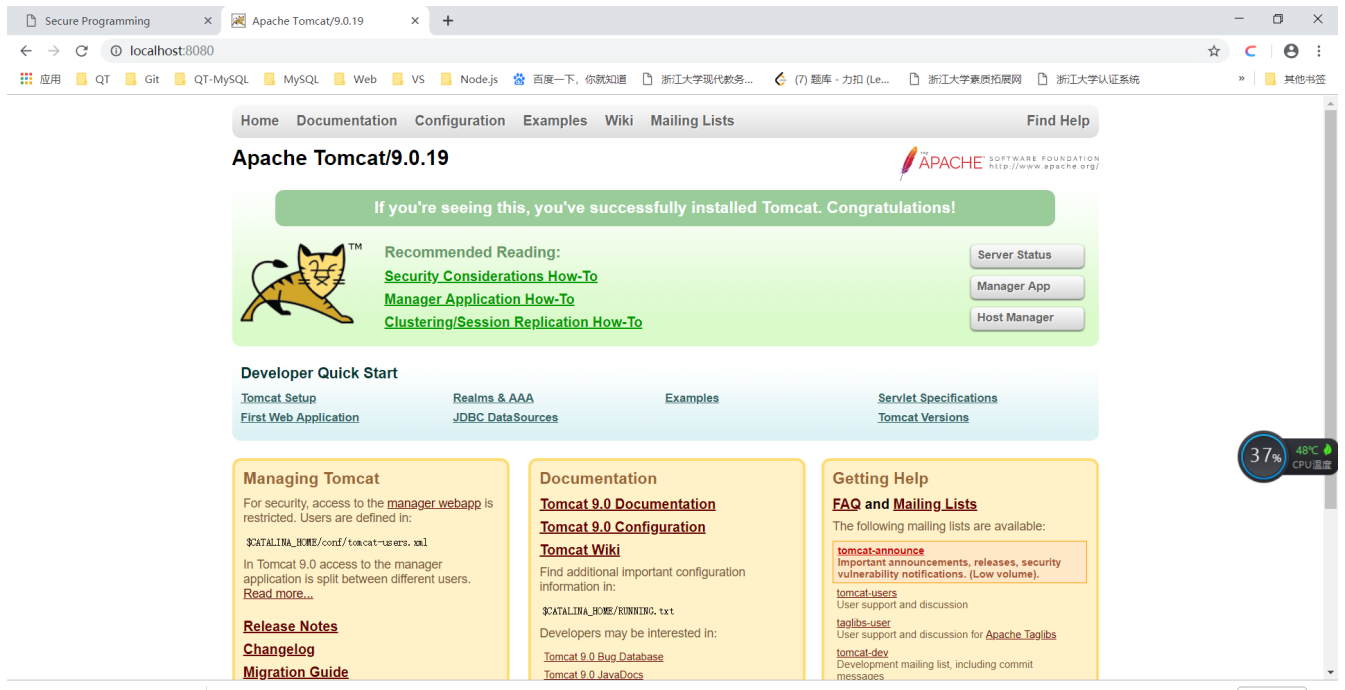
```
java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = org.apache.juli.OneLineFormatter
java.util.logging.ConsoleHandler.encoding = GBK

java.util.logging.ConsoleHandler.level = FINE
java.util.logging.ConsoleHandler.formatter = org.apache.juli.OneLineFormatter
java.util.logging.ConsoleHandler.encoding = UTF-8
```

重启tomcat，恢复正常：

A screenshot of a Tomcat console window. The title bar says "Tomcat". The window contains a log of startup messages. The messages show the deployment of web application directories and the start of the HTTP and AJP protocols. The logs are as follows:
29-Apr-2019 21:50:16.591 信息 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [D:\Apache-Tomcat-9.0.19\webapps\host-manager] has finished in [27] ms
29-Apr-2019 21:50:16.591 信息 [main] org.apache.catalina.startup.HostConfig.deployDirectory 把web 应用程序部署到目录 [D:\Apache-Tomcat-9.0.19\webapps\manager]
29-Apr-2019 21:50:16.613 信息 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [D:\Apache-Tomcat-9.0.19\webapps\manager] has finished in [21] ms
29-Apr-2019 21:50:16.613 信息 [main] org.apache.catalina.startup.HostConfig.deployDirectory 把web 应用程序部署到目录 [D:\Apache-Tomcat-9.0.19\webapps\ROOT]
29-Apr-2019 21:50:16.631 信息 [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [D:\Apache-Tomcat-9.0.19\webapps\ROOT] has finished in [18] ms
29-Apr-2019 21:50:16.634 信息 [main] org.apache.coyote.AbstractProtocol.start 开始协议处理句柄["http-nio-8080"]
29-Apr-2019 21:50:16.643 信息 [main] org.apache.coyote.AbstractProtocol.start 开始协议处理句柄["ajp-nio-8009"]
29-Apr-2019 21:50:16.649 信息 [main] org.apache.catalina.startup.Catalina.start Server startup in [1,064] milliseconds

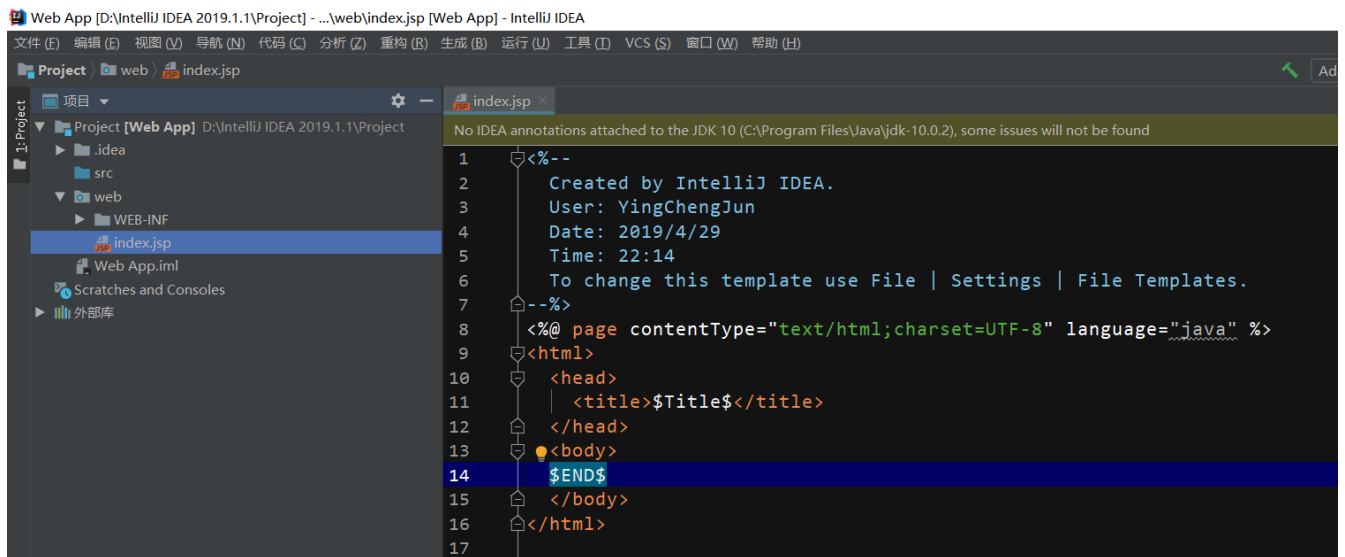
浏览器中输入：localhost:8080即可启动（端口号配置在conf/server.xml的文件中）



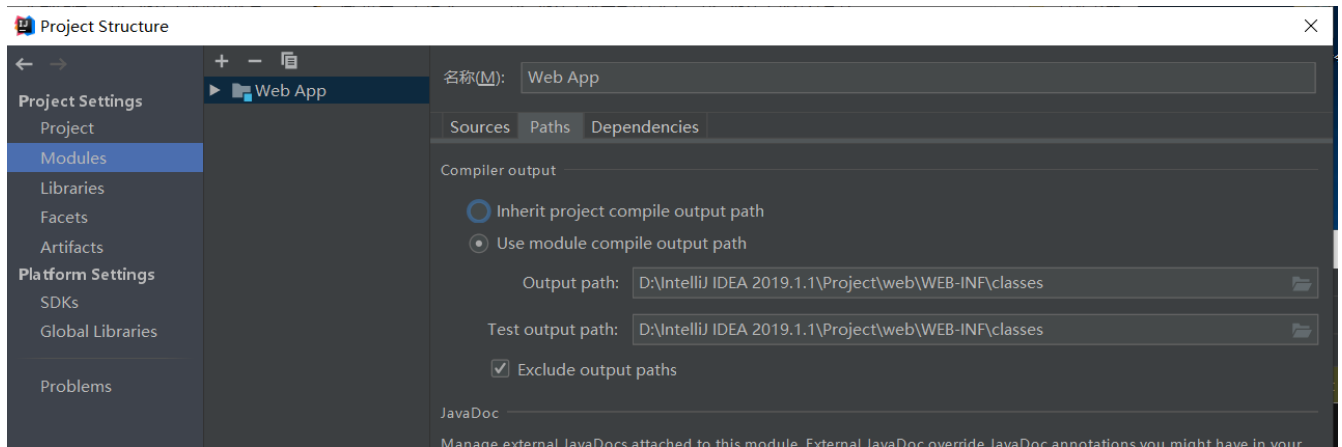
Step 3:

IntelliJ IDEA此前在本电脑中已有安装，这里不赘述安装过程。

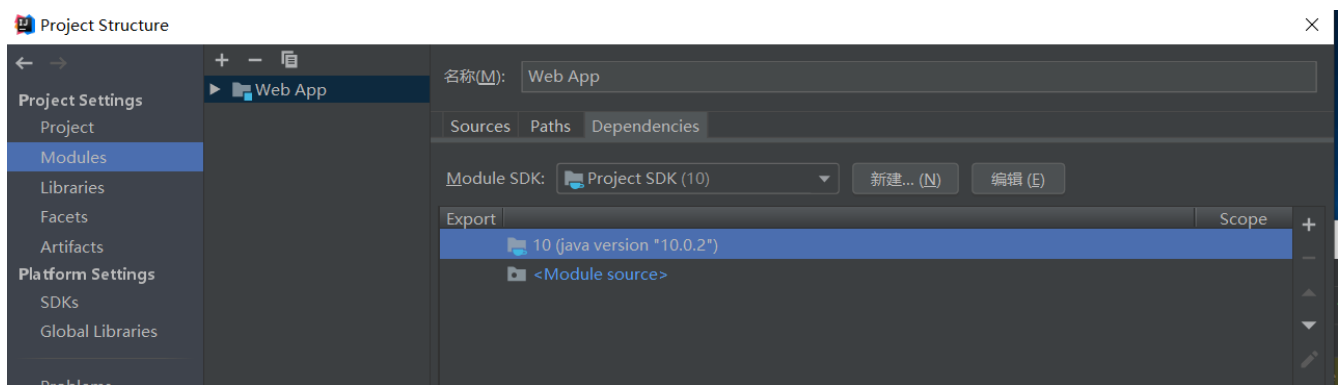
打开IDEA后，新建一个Java Web项目，建立后如下所示：

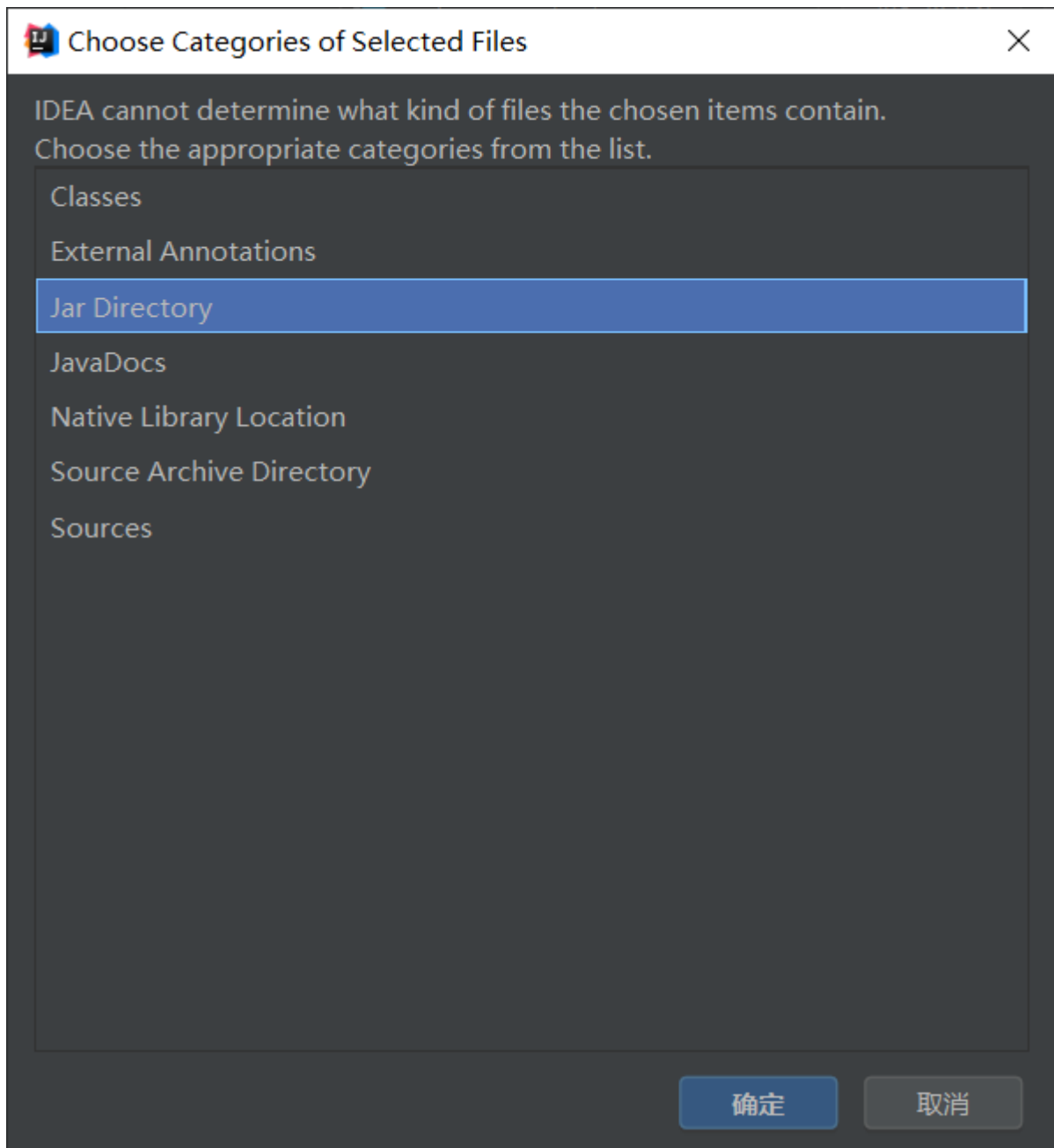


建立完项目后，在WEB-INF目录下新建两个子目录：lib和classes，分别存放jar包以及字节码文件。接下来在File菜单栏中的Project Settings里，将Modules模块的Paths改为如下图配置：



然后选中Dependencies中的 10(java version "10.0.2")，再点击右边 + 号选择第一项，选择Jar Directory并将目录定位到lib





接下来在run中的Edit Configurations里新建一个Tomcat Server，选择Local并做相应配置：

+ - [icon] [icon] [icon] [icon] [icon] [icon]

Add New Configuration

JavaScript Debug

JBoss Server ▶

Jest

Jetty Server ▶

JSR45 Compatible Server ▶

JUnit

Kotlin

Kotlin script

Maven

MXUnit

npm

NW.js

OpenShift Deployment

OSGi Bundles

Plugin

Protractor

React Native

Resin ▶

Spring Boot

Spring dmServer ▶

Spy-js

Spy-js for Node.js

TestNG

Tomcat Server ▶

TomEE Server ▶

WebLogic Server ▶

WebSphere Server ▶

XSLT

小应用程序

应用程序

远程

Click the + 按钮来创建一个新的配置基于默认设置

Configurations available in Run Dashboard

添加新'Tomcat Server'配置

Local

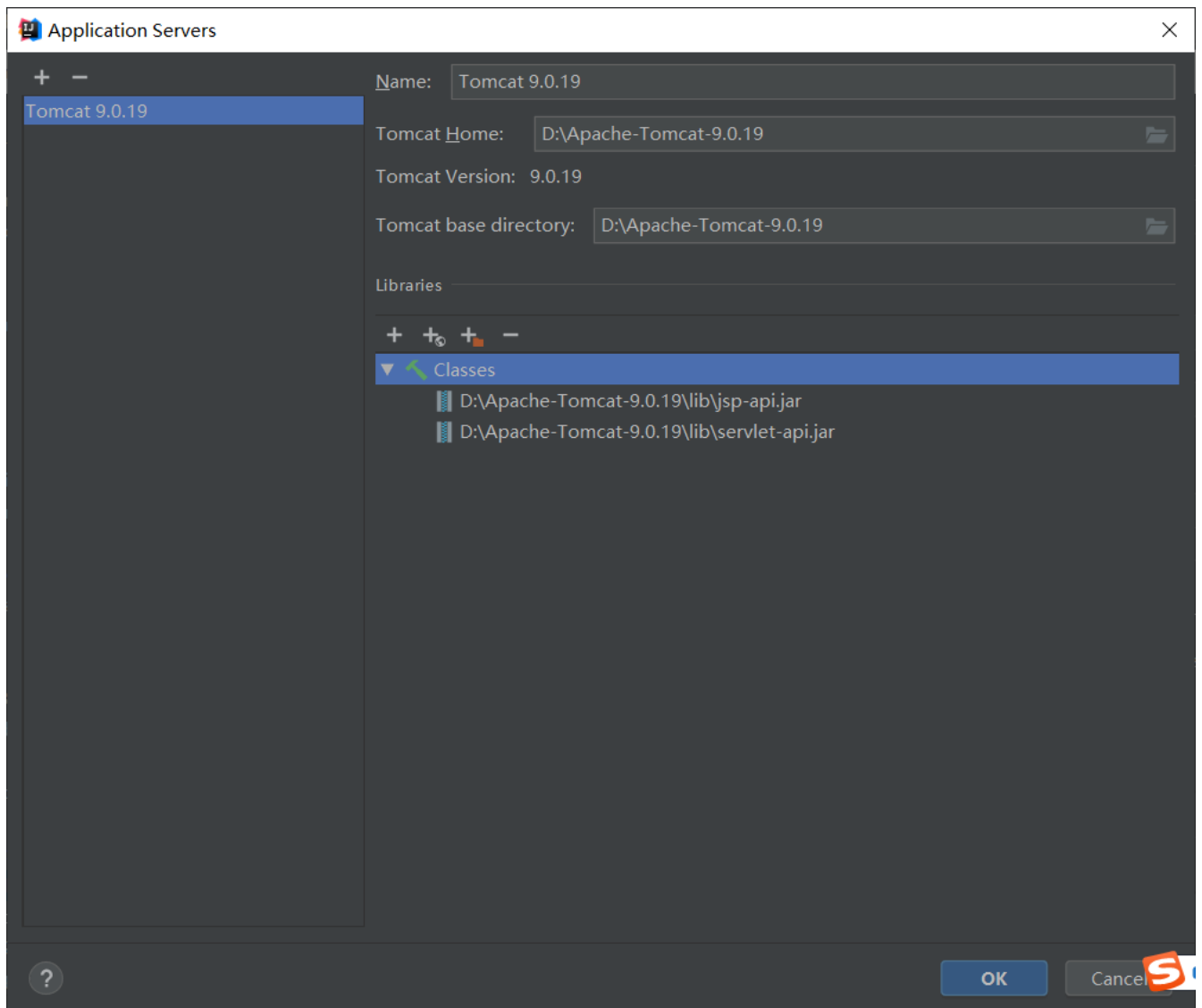
Remote

过程终止

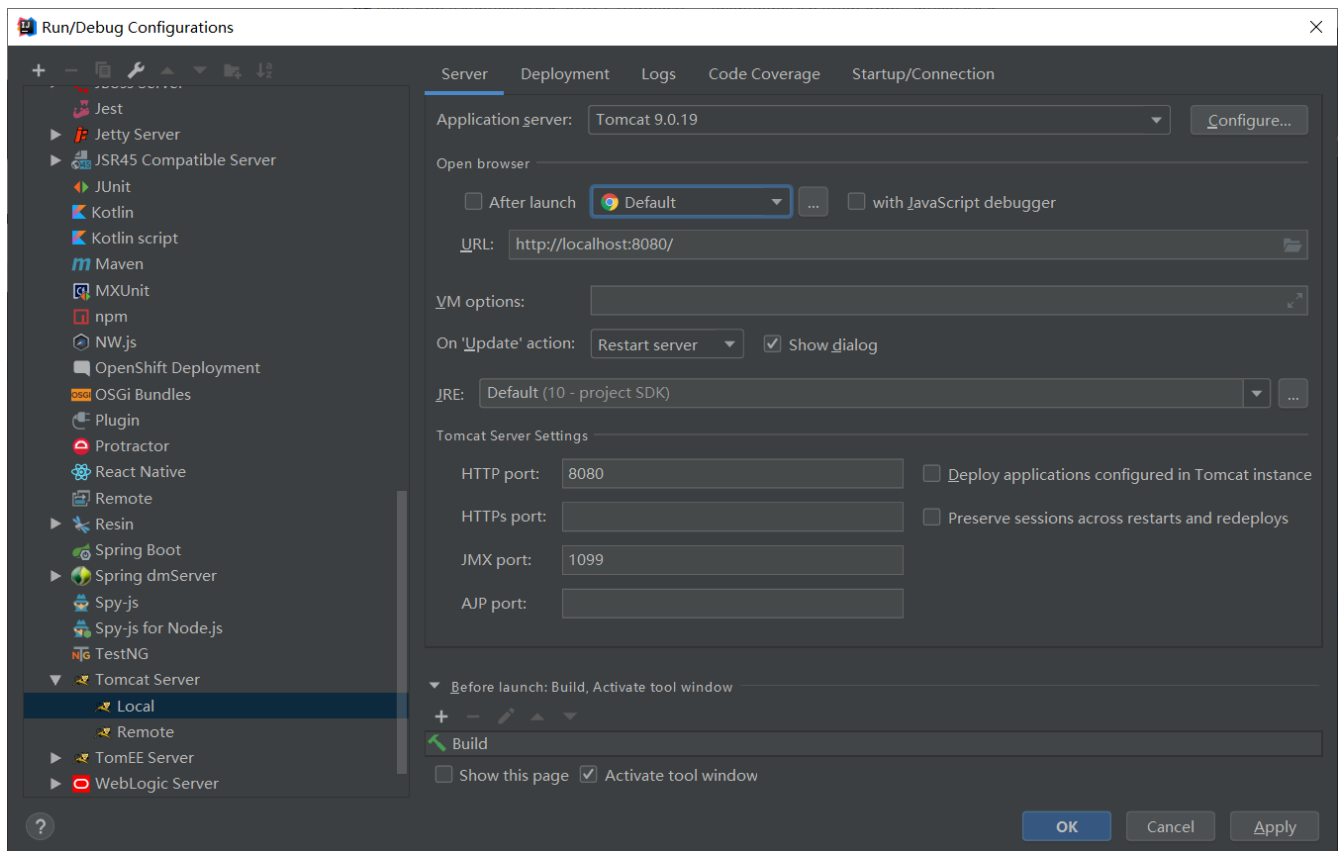
n.confirmation!

Temporary configurations limit: 5

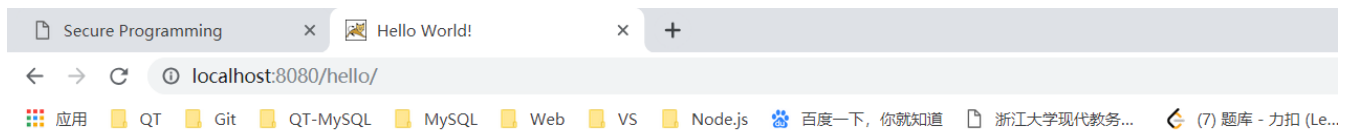
S 英 ' [icon] [icon] [icon] [icon] [icon] [icon] [icon] [icon]



Server配置如下（基本保持默认配置）：



运行启动浏览器访问localhost:8080/hello即可看到如下界面：



Hello World!

Lab 1.2 Implementation of Web Application

姓名：应承峻

学号：3170103456

Lab Topic:

In lab 1.1, you have learned to setup the Web Programming Environment. It is easy, but very important, which is the base of lab 1.2.

In this lab, you will learn how to implement a web application. Usually, a web application is related with database, web server, data access, service, page design and so on. So there are really much you have to learn before you start the Implementation.

Back to the lab, what's our objective:

1. Design and access the database (MySQL recommended).
2. Learn to use web server and deploy your web application (Tomcat recommended).
3. Implement the application (Html, JSP recommended).
4. CSS is required to be used for controlling the page display in the “External Style Sheet” way.

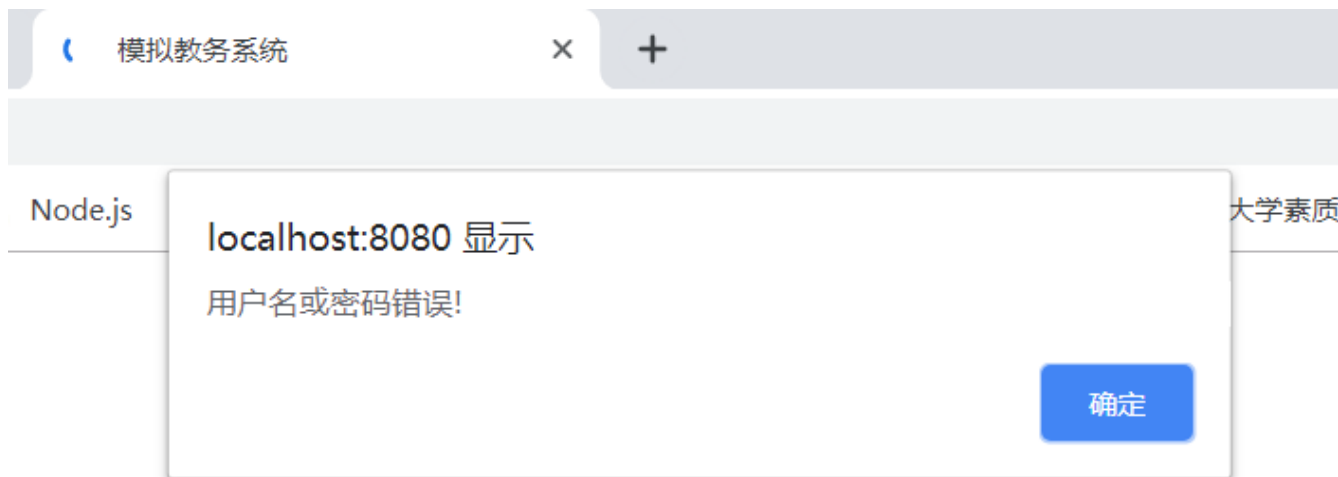
项目描述

本项目模拟了简单的教务系统，面向的群体为学生和教师。

登录界面：学生和教师能够分别在登录界面中进行登录操作，输入用户名和密码点击登录时将会进行验证。



验证后会给出相应的反馈（以下是验证不通过的反馈）：



学生界面：当系统根据数据库结果判定用户身份是学生时，会显示该学生的个人信息以及所有课程的课程编号、课程名称、任课教师、课程学分、最终成绩以及评价的情况，如下图所示：

欢迎，应承峻

退出

个人成绩查询

学号：3170103456

姓名：应承峻

部门：计算机科学与技术学院

课程编号	课程名称	任课教师	课程学分	最终成绩	评价
10001	安全编程技术	admin	2.5	70	及格
10002	数据库系统	admin	6.0	100	优秀
10003	高级数据结构与算法分析	admin	8.0	60	及格
10004	计算机组成	admin	3.5	85	良好
10005	面向信息技术的沟通技巧	admin	3.0	59	不合格
10006	信息安全原理	tacher	4.0	暂无成绩	暂无评价

教师界面：当系统根据数据库结果判定用户身份是教师时，会显示所有教师开设的课程和课程成绩情况。

欢迎，教师A

退出

教师开课情况

工号：admin

姓名：教师A

部门：软件学院

课程编号	课程名称	课程学分	成绩登记
10001	安全编程技术	2.5	
10002	数据库系统	6.0	
10003	高级数据结构与算法分析	8.0	
10004	计算机组成	3.5	
10005	面向信息技术的沟通技巧	3.0	

课程成绩情况

课程编号：10001

课程名称：安全编程技术

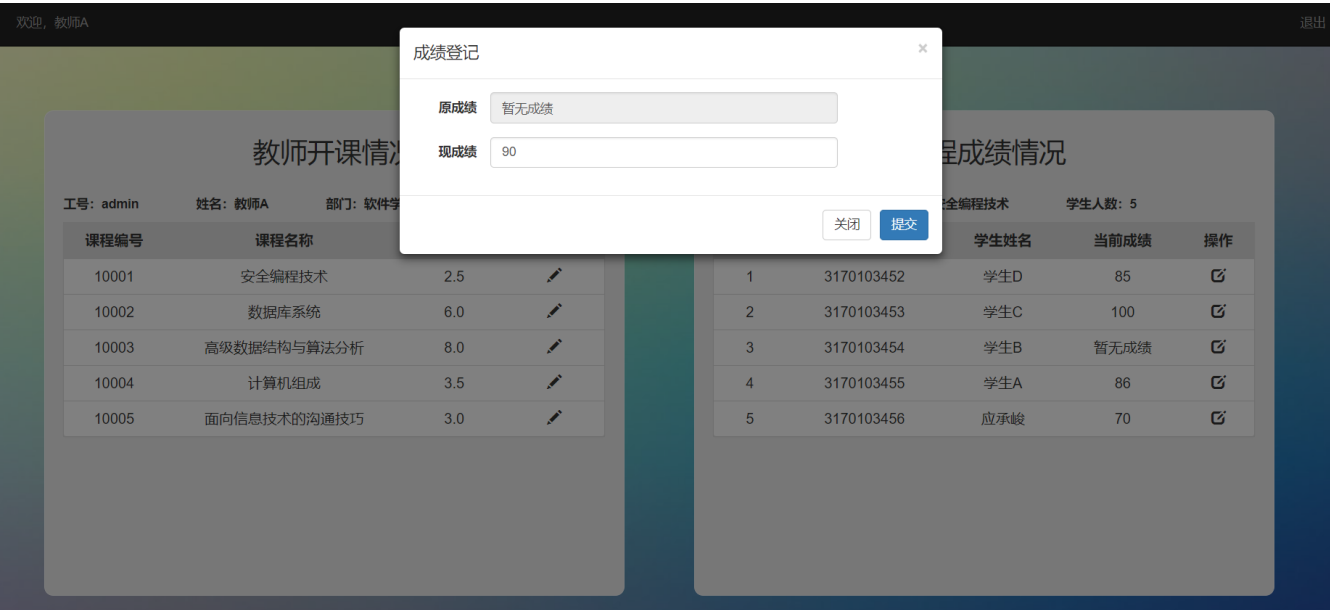
学生人数：5

序号	学生学号	学生姓名	当前成绩	操作
1	3170103452	学生D	85	
2	3170103453	学生C	100	
3	3170103454	学生B	暂无成绩	
4	3170103455	学生A	86	
5	3170103456	应承峻	70	

点击成绩登记时，右边会显示该课程所有学生的成绩情况：



点击操作按钮可以登记/修改学生的课程成绩：



欢迎, 教师A

退出

教师开课情况

工号: admin 姓名: 教师A 部门: 软件学院

课程编号	课程名称	课程学分	成绩登记
10001	安全编程技术	2.5	
10002	数据库系统	6.0	
10003	高级数据结构与算法分析	8.0	
10004	计算机组成	3.5	
10005	面向信息技术的沟通技巧	3.0	

课程成绩情况

课程编号: 10001 课程名称: 安全编程技术 学生人数: 5

序号	学生学号	学生姓名	当前成绩	操作
1	3170103452	学生D	85	
2	3170103453	学生C	100	
3	3170103454	学生B	90	
4	3170103455	学生A	86	
5	3170103456	应承峻	70	

成绩更新后，学生可在其主页查询到信息。

欢迎, 学生B

退出

个人成绩查询

学号: 3170103454 姓名: 学生B 部门: 软件学院

课程编号	课程名称	任课教师	课程学分	最终成绩	评价
10001	安全编程技术	admin	2.5	90	优秀

数据库构建

数据库主要分为三个表，user表存储所有用户的信息，course表存储课程信息，detail表存储学生的选课信息。下面是数据表结构介绍：

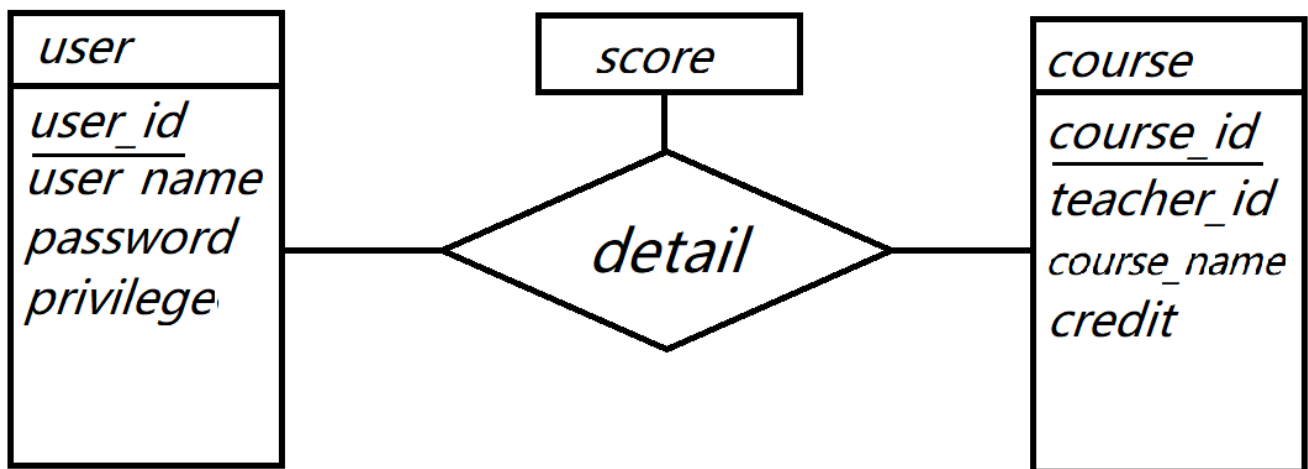
```
1 create table user (
2     user_id varchar(30),    /*学生学号或教师工号*/
3     username varchar(30) not null,    /*姓名*/
4     password varchar(30) not null,    /*密码*/
5     privilege boolean not null default 0, /*权限*/
6     primary key (user_id) /*表的主键为学工号*/
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
8
9 create table course (
10     course_id varchar(30),    /*课程编号*/
11     course_name varchar(30) not null,    /*课程名称*/
12     teacher_id varchar(30) not null,    /*上课教师*/
13     credit float(3,1) not null,    /*学分*/
14     primary key (course_id),    /*表的主键为课程编号*/
```

```

15     foreign key (`teacher_id`) references `user`(`user_id`) on delete cascade on
    update cascade /*上课教师属性的外键约束*/
16 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
17
18 create table detail (
19     user_id varchar(30), /*学生学号*/
20     course_id varchar(30), /*课程编号*/
21     score int(3), /*成绩*/
22     primary key (user_id,course_id), /*主键由学生学号与课程编号共同组成*/
23     foreign key (user_id) references user(user_id) on delete cascade on update cascade
    , /*学生学号属性的外键约束*/
24     foreign key (course_id) references course(course_id) on delete cascade on update
    cascade /*课程编号属性的外键约束*/
25 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

数据库的ER图如下：



项目架构与开发过程

整个项目的技术栈为：HTML+CSS+BootStrap+JavaScript+jQuery+JSP

项目 `web` 目录下的文件架构为：

- public #存放样式表、JS文件、静态图片文件#
- WEB-INF
 - classes #字节码文件#
 - lib #jar包#
 - tld #JSTL核心库
 - web.xml
- index.jsp #登录界面#
- main.jsp #用户主界面
- score.jsp #更新成绩的处理页面

项目的开发过程：

样式库、JS库的导入：

```

1 <script src="public/jquery/jquery.js"></script> /*jQuery库*/
2 <script src="public/bootstrap/js/bootstrap.min.js"></script> /*Bootstrap框架*/
3 <link rel="stylesheet" href="public/bootstrap/css/bootstrap.min.css"/>
4 <link rel="stylesheet" href="public/stylesheet/index.css"/> /*自定义外部样式表*/

```

登录功能的实现：将index.jsp中登录表单的数据以POST方式传送到main.jsp中，由main.jsp中相关的代码进行处理：

其处理过程如下代码所示，我们先检查服务器与用户之间的对话Session是否仍然有效。如果不存在对应的Session并且用户在请求中没有附带参数，则要求用户重新进行登录；如果用户的请求中带有对应的用户名和密码数据，则使用JSTL标签库中的`<sql>`标签进行数据库处理，如果查询到记录则将用户的学工号、姓名、岗位、权限、密码等信息存储到Session中。最后根据是否查询到记录来给用户相应的反馈。

```

1 //登录状态过期
2 <c:if test="${param.username==null || param.password==null}">
3     <c:if test="${sessionScope.uid==null}">
4         <script>
5             alert("请先登录!");
6             window.location.href = "./index.jsp";
7         </script>
8     </c:if>
9 </c:if>
10
11 //数据库信息验证与反馈
12 <c:if test="${param.username!=null && param.password != null}">
13     <sql:query dataSource="${conn}" var="user">
14         select * from user where user_id = ? and password = ?;
15     <sql:param value="${param.username}" />
16     <sql:param value="${param.password}" />
17 </sql:query>
18 <c:if test="${user.rowCount==1}">
19     <% session.setMaxInactiveInterval(60*2); %>
20     <c:set var="uid" value="${user.rows[0].user_id}" scope="session"/>
21     <c:set var="name" value="${user.rows[0].username}" scope="session"/>
22     <c:set var="dept" value="${user.rows[0].department}" scope="session"/>
23     <c:set var="pri" value="${user.rows[0].privilege}" scope="session"/>
24     <c:set var="pwd" value="${user.rows[0].password}" scope="session"/>
25     <script>
26         alert("登录成功!");
27     </script>
28 </c:if>
29 <c:if test="${user.rowCount!=1}">
30     <script>
31         alert("用户名或密码错误!");
32         window.location.href = "./index.jsp";
33     </script>
34 </c:if>
35 </c:if>

```

根据用户渲染界面：验证通过后，根据用户的身份（从Session中获得）选择对应的页面进行渲染。通过`if`标签进行条件渲染。

```

1 <!-- 学生界面, 学生的权限值为0 -->
2 <c:if test="${sessionScope.pri == false}" >
3     <!-- Code -->
4 </c:if>
5 <!-- 教师界面, 教师的权限值为1 -->
6 <c:if test="${sessionScope.pri == true}" >
7     <!-- Code -->
8 </c:if>

```

学生成绩渲染：先通过Session中的uid值得到学生的学号，然后从数据库中查询该学生所选课程以及成绩信息，将结果集保存在result中，然后用JSTL的 `c` 标签的 `forEach` 函数将结果集中的每一行信息打印出。

```

1 <!-- 数据库查询 -->
2 <sql:query dataSource="${conn}" var="result">
3     select * from detail natural join course where user_id = ?;
4     <sql:param value="${sessionScope.uid}" />
5 </sql:query>
6
7 <div class="cbox">
8     <h2 class="title">个人成绩查询</h2>
9     <div class="infogroup"> <!-- 个人信息显示 -->
10         <b class="info">学号: <c:out value="${sessionScope.uid}"/></b>
11         <b class="info">姓名: <c:out value="${sessionScope.name}"/></b>
12         <b class="info">部门: <c:out value="${sessionScope.dept}"/></b>
13     </div>
14     <table class="table table-hover panel panel-default">
15         <thead> <!-- 表头 -->
16             <tr>
17                 <th>课程编号</th>
18                 <th>课程名称</th>
19                 <th>任课教师</th>
20                 <th>课程学分</th>
21                 <th>最终成绩</th>
22                 <th>评价</th>
23             </tr>
24         </thead>
25         <tbody>
26             <c:forEach var="row" items="${result.rows}"> <!-- 显示每门课成绩 -->
27                 <tr>
28                     <td><c:out value="${row.course_id}"/></td> <!-- 课程号 -->
29                     <td><c:out value="${row.course_name}"/></td> <!-- 课程名称 -->
30                     <td><c:out value="${row.teacher_id}"/></td> <!-- 教师姓名 -->
31                     <td><c:out value="${row.credit}"/></td> <!-- 学分 -->
32                     <c:if test="${row.score != null}" > <!-- 成绩与评价 -->
33                         <td><c:out value="${row.score}"/></td>
34                         <c:if test="${row.score < 60}" >
35                             <td class="fail">不合格</td>
36                         </c:if>
37                         <c:if test="${row.score >= 60 && row.score < 80}" >
38                             <td class="fine">及格</td>
39                         </c:if>
40                         <c:if test="${row.score >= 80 && row.score < 90}" >

```



```

41         <td class="fine">良好</td>
42     </c:if>
43     <c:if test="${row.score >= 90}" >
44         <td class="fine">优秀</td>
45     </c:if>
46 </c:if>
47 <c:if test="${row.score == null}" >
48     <td>暂无成绩</td>
49     <td>暂无评价</td>
50 </c:if>
51 </tr>
52 </c:forEach>
53 </tbody>
54 </table>
55 </div>

```

教师信息渲染的模块类似于学生信息的渲染，这里介绍一下点击教师开设课程最右边的图标来显示这门课所有学生成绩的功能的实现：

我们通过jQuery来完成这一个功能，通过**类选择器**将“铅笔图标”的点击事件与触发函数进行绑定。然后通过选择器找到该课程在教师课程结果集中的下标，然后使用GET方式重新访问该网页并带上下标的参数。

```

1  $(".cursor.glyphicon-pencil").click(function () {
2      var id = $(this).parents("tr").index();
3      window.location.href = "./main.jsp?rid="+id;
4  });

```

网页会根据下标参数对数据结果进行相应的查询和渲染：

```

1  <!-- 默认下标为0 -->
2  <c:if test="${param.rid==null}">
3      <c:set var="rid" value="0"/>
4  </c:if>
5  <c:if test="${param.rid!=null}">
6      <c:set var="rid" value="${param.rid}"/>
7  </c:if>
8
9  <!-- 查询课程号对应的学生成绩信息 -->
10 <c:if test="${courses.rowCount>0}">
11     <sql:query dataSource="${conn}" var="course_status">
12         select * from detail natural join user where course_id = ?;
13         <sql:param value="${courses.rows[rid].course_id}" />
14     </sql:query>
15 </c:if>

```

登记分数功能：在教师点击学生成绩中的操作按钮时，会弹出模态框，模态框会显示用户的原有成绩和更新成绩。

如下图所示，可以通过选择器来得到用户原来的分数并进行显示

```

1 $(".glyphicon-edit").click(function () {
2     var oscore = $(this).parents("tr").find(".osc").text(); //原分数
3     var sid = $(this).parents("tr").find(".sid").text(); //学生id
4     $("input[name='oscore']").val(oscore); //显示原分数
5     $("input[name='cscore']").val(""); //清空输入框
6     $("input[name='suid']").val(sid); //保存学号和课程号
7     $("input[name='cuid']").val(<c:out value="${courses.rows[rid].course_id}"/>);
8     $('#scoremodal').modal('show');
9 });

```

在表单中加入如下两个隐藏域，用以保存学号和课程号，方便后续信息处理。

```

1 <input type="hidden" class="form-control" name="suid">
2 <input type="hidden" class="form-control" name="cuid">

```

提交后，页面转到 `score.jsp` 中，通过如下代码进行处理：首先验证Session中的用户权限，然后进行数据库操作，并根据返回值 `flag` 确定反馈结果。

```

1 <!-- 验证权限 -->
2 <c:if test="${sessionScope.uid==null || sessionScope.pri==false}">
3     <%
4         session.invalidate();
5         response.sendRedirect("index.jsp");
6     %>
7 </c:if>
8
9 <!-- 修改成绩 -->
10 <sql:update dataSource="${conn}" var="flag">
11     update detail set score = ? where user_id = ? and course_id = ?;
12     <sql:param value="${param.cscore}" />
13     <sql:param value="${param.suid}" />
14     <sql:param value="${param.cuid}" />
15 </sql:update>
16
17 <!-- 反馈结果 -->
18 <c:if test="${flag==1}">
19     <script>alert("登记成功!");window.location.href = "./main.jsp";</script>
20 </c:if>
21 <c:if test="${flag==0}">
22     <script>alert("登记失败!");window.location.href = "./main.jsp";</script>
23 </c:if>

```

项目部署与测试

项目通过Tomcat来实现部署，部署的过程在实验1.1中已有描述

项目的配置如下：

Name: ☐ Share

Server Deployment Logs Code Coverage Startup/Connection

Application server:

Open browser

☐ After launch ... ☐ with JavaScript debugger

URL:

VM options:

On 'Update' action: ☒ Show dialog

On frame deactivation:

JRE: ...

Tomcat Server Settings

HTTP port: ☐ Deploy applications configured in Tomcat instance

HTTPs port: ☐ Preserve sessions across restarts and redploys

JMX port:

AJP port:

► Before launch: Build, Build Artifacts, Activate tool window

测试过程已经在项目介绍中给出。