

N2: Java EE/MySQL 学生成绩管理系统

本实训基于 Java EE 7(Struts 2.3)实现学生成绩管理系统,Web 服务器使用 Tomcat 9.0, 访问 MySQL 5.6 数据库。

N2.1 Java EE 开发平台搭建

N2.1.1 安装软件

1. 安装 JDK 8

本实训使用的版本是 JDK 8 Update 121, 安装可执行文件为 jdk-8u121-windows-i586, 双击启动安装向导, 如图 N2.1 所示。



图 N2.1 安装 JDK

按照向导的步骤操作, 完成后, JDK 安装到了目录 “C:\Program Files\Java\jdk1.8.0_121”。接下来设置环境变量以便系统找到和使用此 JDK。

1) 打开【环境变量】对话框

右击桌面上的“计算机”图标, 选择【属性】, 在弹出的控制面板主页中单击“高级系统设置”链接项, 在弹出的【系统属性】对话框中单击【环境变量】按钮, 弹出【环境变量】对话框, 操作如图 N2.2 所示。

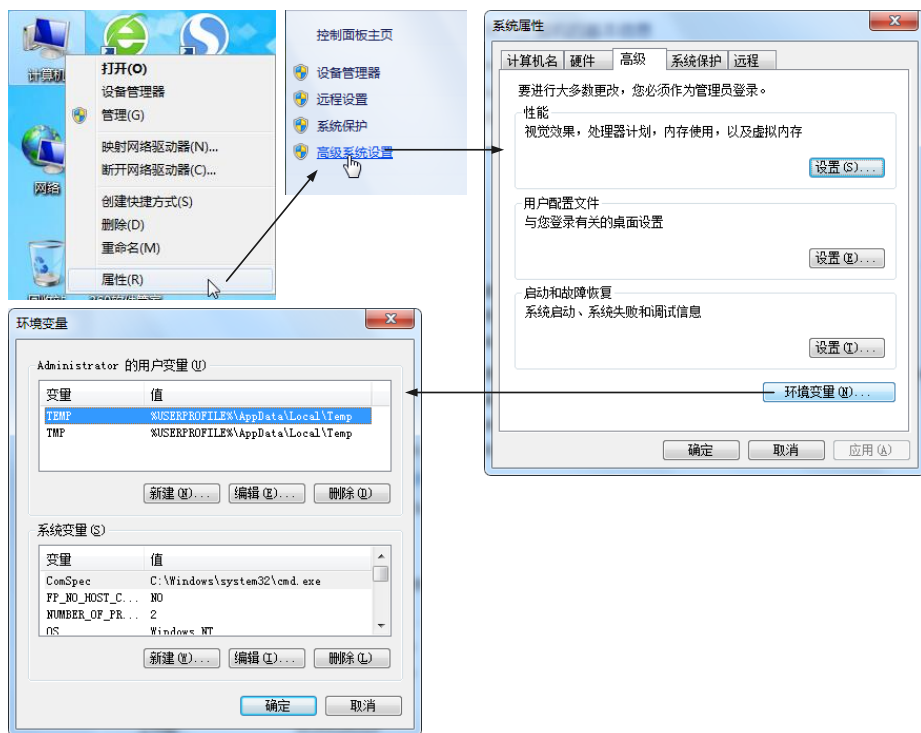


图 N2.2 打开【环境变量】对话框

2) 新建系统变量 JAVA_HOME

在“系统变量”列表下单击【新建】按钮，弹出【新建系统变量】对话框。在“变量名”栏中输入“JAVA_HOME”，在“变量值”栏中输入 JDK 安装路径“C:\Program Files\Java\jdk1.8.0_121”，如图 N2.3 (a) 所示，单击【确定】按钮。

3) 设置系统变量 Path

在“系统变量”列表中找到名为“Path”的变量，单击【编辑】按钮，在“变量值”字符串中加入路径“%JAVA_HOME%\bin;”，如图 N2.3 (b) 所示，单击【确定】按钮。



(a) 新建 JAVA_HOME 变量



(b) 编辑 Path 变量

图 N2.3 设置环境变量

选择任务栏中的【开始】→【运行】，输入“cmd”后回车，在命令行中输入“java -version”，如果环境变量设置成功就会出现 Java 的版本信息，如图 N2.4 所示。

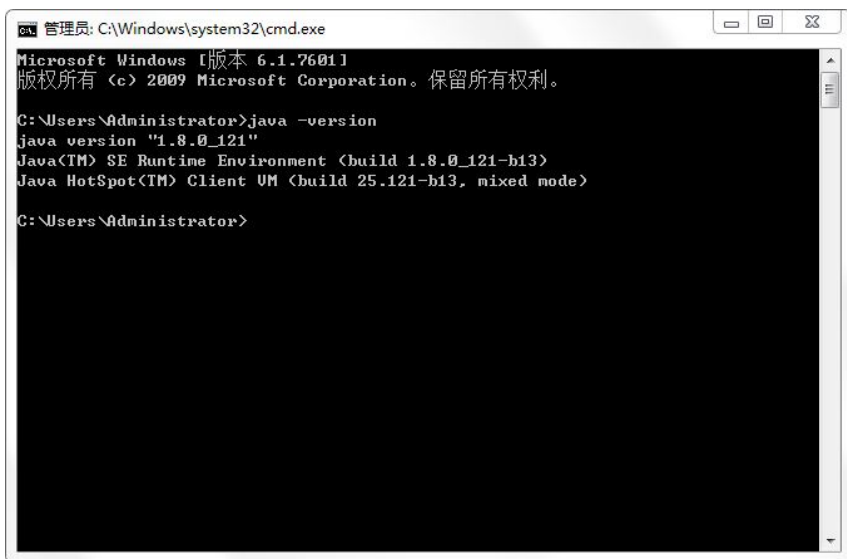


图 N2.4 JDK 安装成功

2. 安装 Tomcat 9

本实训采用最新 Tomcat 9 作为承载 Java EE 应用的服务器，可在其官网：<http://tomcat.apache.org/> 下载。图 N2.5 是 Tomcat 的发布页，其中，Core 下的 Windows Service Installer（图上标注的）是一个安装版软件。

下载获得执行文件 `apache-tomcat-9.0.0.M17.exe`，双击启动安装向导，如图 N2.6 所示，安装过程均取默认选项，不再详细说明。

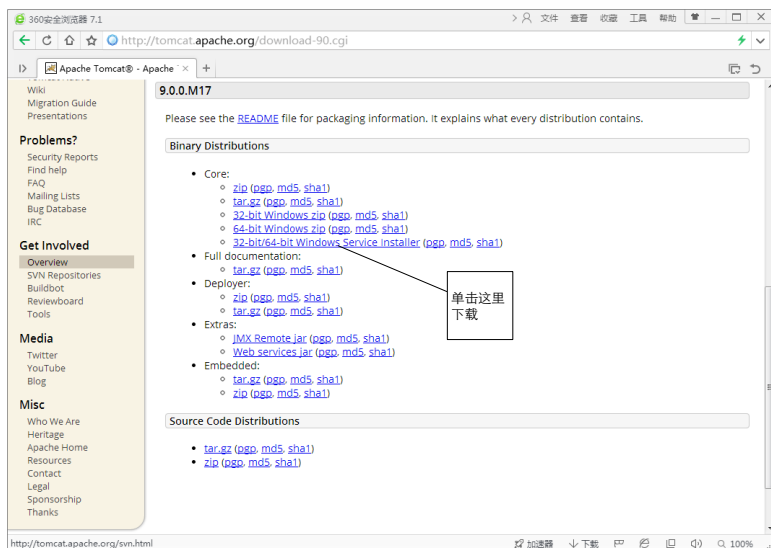


图 N2.5 Apache 官网上的 Tomcat 发布页



图 N2.6 Tomcat 9 安装向导

安装完 Tomcat 后会自行启动，打开浏览器输入“http://localhost:8080”后回车测试。若呈现如图 N2.7 所示的页面，就表明安装成功。

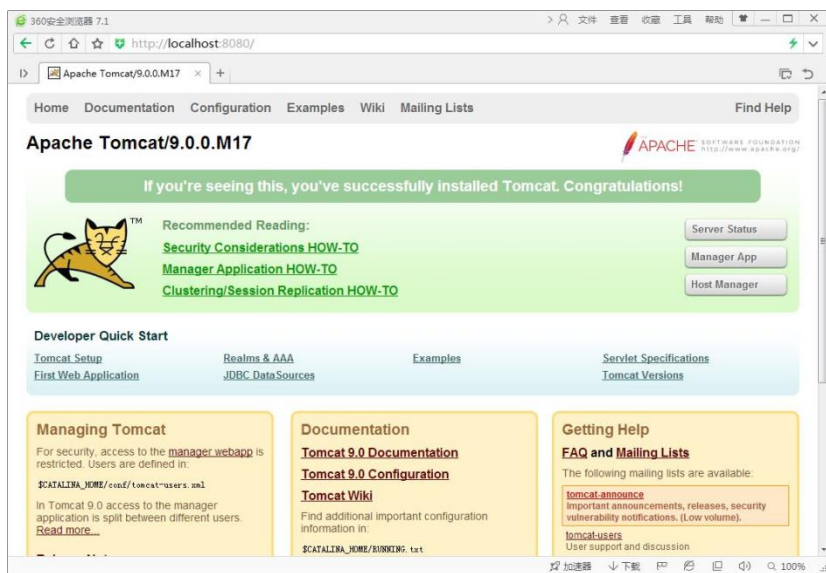


图 N2.7 Tomcat 9 安装成功

3 . 安装 MyEclipse 2017

MyEclipse 企业级工作平台（MyEclipse Enterprise Workbench，简称 MyEclipse）是一个功能强大的 Java EE 集成开发环境（IDE），MyEclipse 中文官网：<http://www.myeclipsecn.com/>。本实训使用最新版 MyEclipse 2017，从官网下载安装包执行文件 myeclipse-2017-ci-1-offline-installer-windows.exe，双击启动安装向导，如图 N2.8 所示。

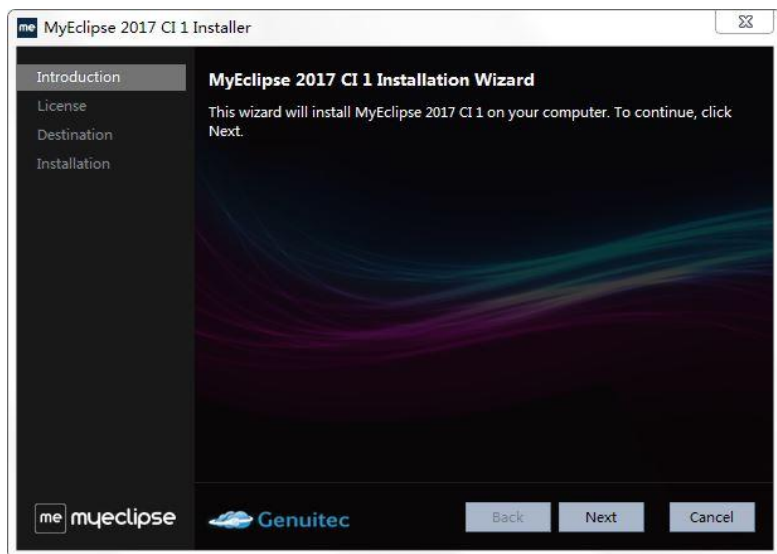


图 N2.8 MyEclipse 2017 安装向导

按照向导的指引往下操作，安装过程从略。

N2.1.2 环境整合

1. 配置 MyEclipse 2017 所用的 JRE

在 MyEclipse 2017 中内嵌了 Java 编译器，但为了使用我们安装的最新 JDK，需要手动配置。启动 MyEclipse 2017，选择主菜单【Window】→【Preferences】，出现如图 N2.9 所示的窗口。

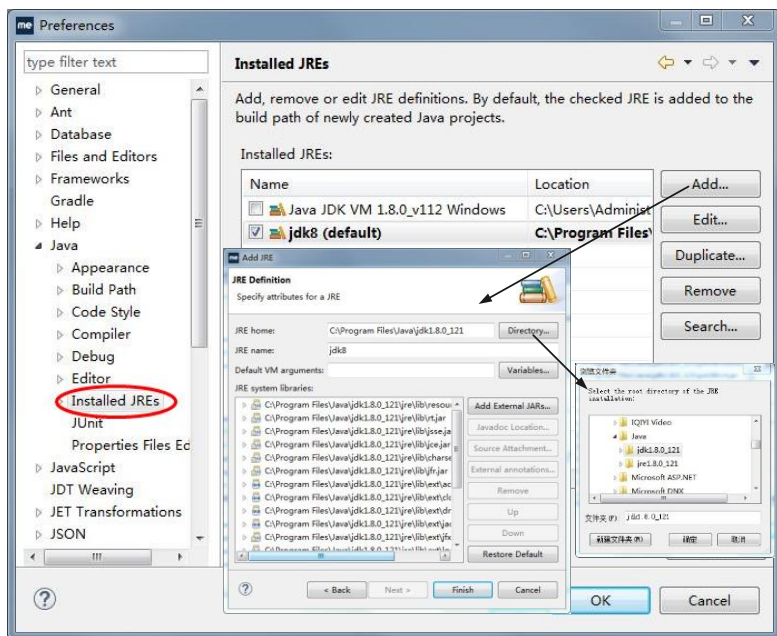


图 N2.9 配置 MyEclipse 2017 的 JRE

在展开的左侧项目树中选中“Java”→“Installed JREs”项，单击右边【Add...】按钮，添加自己安装的 JDK 并命名为 jdk8。

2. 集成 MyEclipse 2017 与 Tomcat 9

启动 MyEclipse 2017，选择主菜单【Window】→【Preferences】，在展开的左侧项目树中选中“Servers”→“Runtime Environments”项，单击右边【Add...】按钮，从弹出的【New Server Runtime Environment】对话框列表中选“Tomcat”→“Apache Tomcat v9.0”项，单击【Next】，如图 N 2.10 所示。

在【Tomcat Server】对话框中设置 Tomcat 9 的安装路径及所用 JRE（从下拉列表中选择前面刚设置的名为 jdk8 的 Installed JRE），如图 N2.11 所示。

在 MyEclipse 2017 工具栏上单击复合按钮  右边的下箭头，选择【Tomcat v9.0 Server at localhost】→【Start】，主界面下方控制台区就会输出 Tomcat 的启动信息，如图 N2.12 所示，说明服务器已开启。

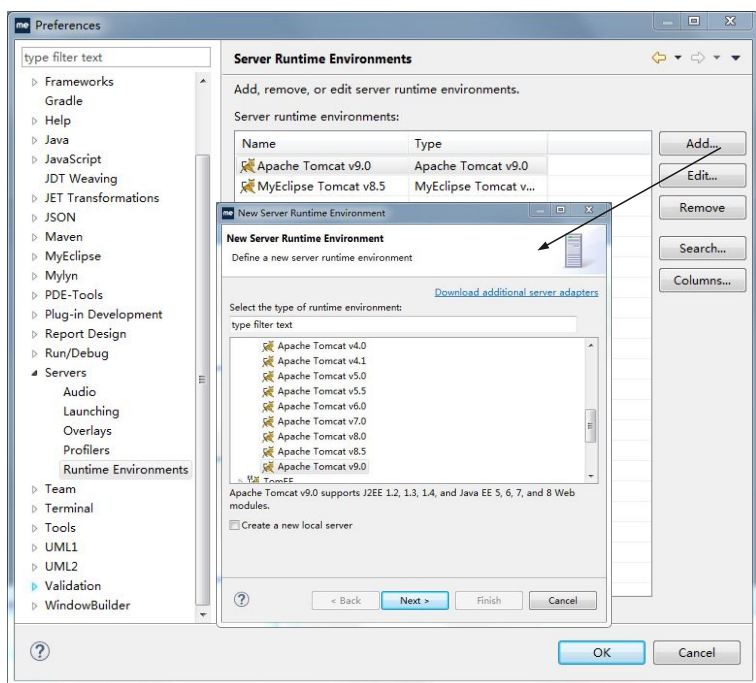


图 N2.10 配置 MyEclipse 2017 服务器

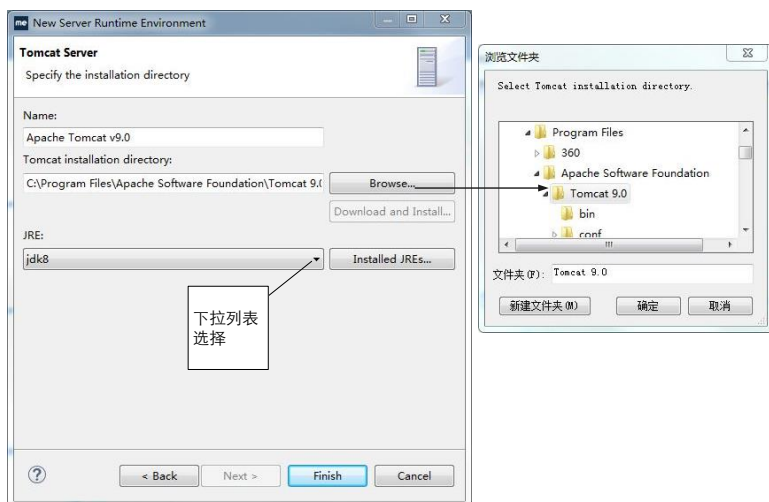


图 N2.11 配置 Tomcat 9 路径及所使用的 JRE

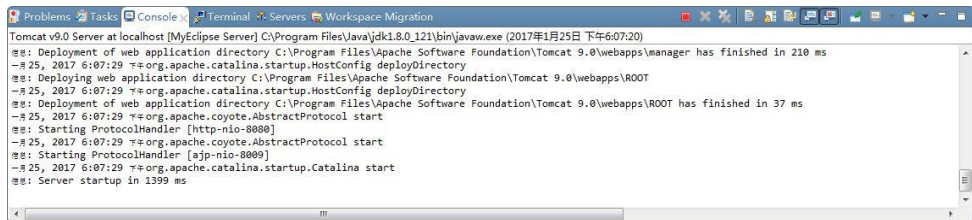


图 N2.12 由 MyEclipse 2017 来启动 Tomcat 9

打开浏览器，输入 `http://localhost:8080` 后回车。如果配置成功，将出现与前图 N2.7 一模一样的 Tomcat 9 首页，表示 MyEclipse 2017 已经与 Tomcat 9 紧密集成了。

至此，一个以 MyEclipse 2017 为核心的 Java EE 应用开发平台搭建成功。

N2.2 创建 Struts 2 项目

N2.2.1 创建 JavaEE 项目

启动 MyEclipse 2017，选择主菜单【File】→【New】→【Web Project】，出现如图 N2.13 所示的对话框，填写“Project Name”栏（项目名）为“xscj”，在“Java EE version”下拉列表中选“JavaEE 7 - Web 3.1”，其余保持默认。

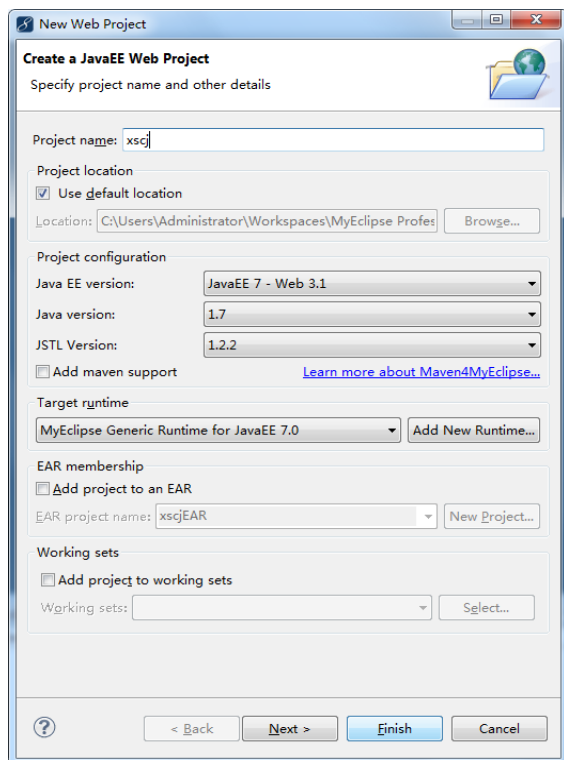


图 N2.13 创建 JavaEE 项目

单击【Next】按钮，在“Web Module”页中勾选“Generate web.xml deployment descriptor”（自动生成项目的 web.xml 配置文件）；在“Configure Project Libraries”页中勾选“JavaEE 7.0 Generic Library”，同时取消选择“JSTL 1.2.2 Library”，如图 N2.14 所示。

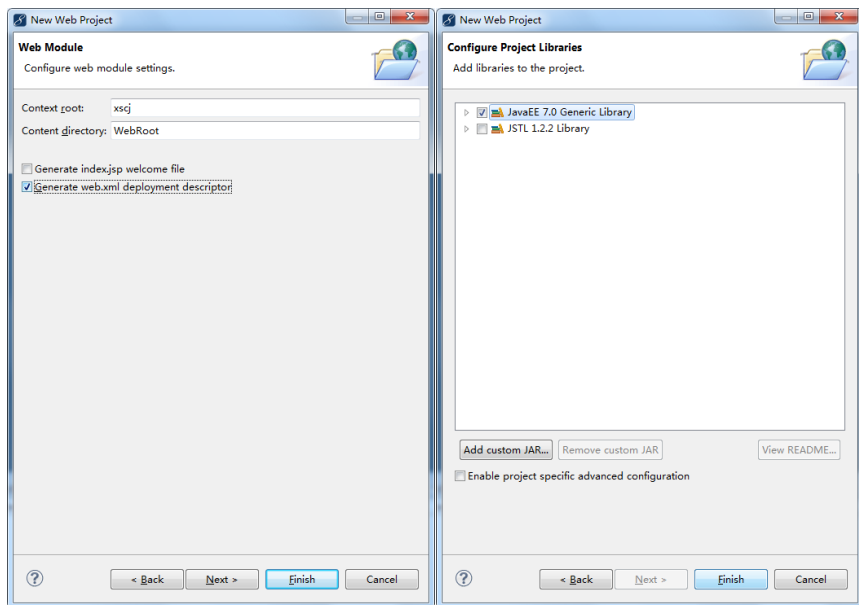


图 N2.14 项目设置

设置完成，单击【Finish】按钮，MyEclipse 会自动生成一个 JavaEE 项目。

N2.2.2 加载 Struts 2 包

登录 <http://struts.apache.org/> 下载 Struts 2 完整版，本实训使用的是 Struts 2.3.20。将下载的文件 struts-2.3.20-all.zip 解压，得到文件夹包含的目录如图 N2.15 所示。

apps: 包含基于 Struts 2 的示例应用，对学习者来说是非常有用的资料。

docs: 包含 Struts 2 的相关文档，如 Struts 2 的快速入门、Struts 2 的 API 文档等内容。

lib: 包含 Struts 2 框架的核心类库，以及 Struts 2 的第三方插件类库。

src: 包含 Struts 2 框架的全部源代码。

在大多数情况下，使用 Struts 2 的 JavaEE 应用并不需要用到 Struts 2 的全部特性，开发 Struts 2 程序一般只需用到 lib 下的 9 个 jar 包，包括：

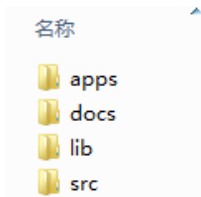


图 N2.15 Struts 2.3.20 目录

① 传统 Struts 2 的 5 个基本类库。

```
struts2-core-2.3.20.jar
xwork-core-2.3.20.jar
ognl-3.0.6.jar
commons-logging-1.1.3.jar
freemarker-2.3.19.jar
```

② 附加的 4 个库。

```
commons-io-2.2.jar
commons-lang3-3.2.jar
javassist-3.11.0.GA.jar
commons-fileupload-1.3.1.jar
```

将它们一起复制到项目的 WebRoot\WEB-INF\lib 路径下，右击项目名，从弹出的菜单中选【Refresh】刷新即可。

然后在 WebRoot/WEB-INF 目录下配置 web.xml 文件，如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
    <display-name>xscj</display-name>
    <filter>
        <filter-name>struts2</filter-name>
        <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
        <init-param>
            <param-name>actionPackages</param-name>
            <param-value>com.mycompany.myapp.actions</param-value>
        </init-param>
    </filter>
    <filter-mapping>
        <filter-name>struts2</filter-name>
        <url-pattern>/*</url-pattern>
    </filter-mapping>
    <welcome-file-list>
        <welcome-file>main.jsp</welcome-file>
    </welcome-file-list>
</web-app>
```

N2.2.3 连接 MySQL

1. 加载数据库驱动

操作与前加载 Struts 2 包的方式一样。从网上下载得到 MySQL 的 JDBC 驱动包 `mysql-connector-java-5.1.40-bin.jar`，将其复制到项目的 `WebRoot\WEB-INF\lib` 路径下，右击项目名刷新。当然，也可以将 MySQL 驱动包与 Struts 2 的 9 个 jar 包一次性地加载到项目中。

2. 编写 JDBC 驱动类

编写用于连接 MySQL 的 Java 类（JDBC 驱动类），在项目 `src` 下建立 `org.easybooks.xscj.jdbc` 包，其下创建 `MySQLConn.java`，代码如下：

```
package org.easybooks.xscj.jdbc;
import java.sql.*;
public class MySQLConn {
    public static Connection conns; //连接对象（定义为"public static"便于程序随时获取和使用该连接）
    static {
        try {
            /**加载并注册 MySQL 的 JDBC 驱动*/
            Class.forName("com.mysql.jdbc.Driver");
            /**创建到 MySQL 的连接*/
            conns = DriverManager.getConnection("jdbc:mysql://localhost:3306/xscj?user=root&password=njnu123456&useUnicode=true&useSSL=false&characterEncoding=GBK");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

3. 构造值对象

为了能用 Java 面向对象的方式访问 MySQL，要预先创建“学生”、“课程”和“成绩”的值对象，它们都位于 `src` 下的 `org.easybooks.xscj.vo` 包中。

(1) “学生”值对象

`Student.java` 构建“学生”的值对象，代码如下：

```
package org.easybooks.xscj.vo;
import java.util.*;
public class Student implements java.io.Serializable {
    private String xm; //姓名
    private String xb; //性别
    private Date cssj; //出生时间
    private int kcs; //课程数
    private String bz; //备注
    private byte[] zp; //照片（字节数组）
    public Student() { } //构造方法
    /**各属性的 getter/setter 方法*/
    /**xm（姓名）属性*/
    public String getXm() { //getter 方法
        return this.xm;
    }
}
```

```

    }
    public void setXm(String xm) {           //setter 方法
        this.xm = xm;
    }
    //省略其余属性的 getter/setter 方法
    ...
}

```

Java 值对象是为实现对数据库面向对象的持久化访问而构造的，它有着固定的格式，包括属性声明、构造方法以及各个属性的 getter/setter 方法，其实质就是一个 **JavaBean**。值对象的属性成员变量一般要与数据库表的字段一一对应，从而便于将 Java 对象操作映射为对数据库中表的操作。各属性的 getter/setter 方法书写形式类同，为节省篇幅，这里省略，详见本书提供的完整源代码。

(2) “课程” 值对象

Course.java 构建“课程”的值对象，代码如下：

```

package org.easybooks.xscj.vo;
public class Course implements java.io.Serializable {
    private String kcm;           //课程名
    private int xs;               //学时
    private int xf;               //学分
    public Course() { }           //构造方法
    /**各属性的 getter/setter 方法*/
    ...
}

```

(3) “成绩” 值对象

Score.java 构建“成绩”的值对象，代码如下：

```

package org.easybooks.xscj.vo;
public class Score implements java.io.Serializable {
    private String xm;           //姓名
    private String kcm;          //课程名
    private int cj;              //成绩
    public Score() { }           //构造方法
    /**各属性的 getter/setter 方法*/
    ...
}

```

N2.3 系统主页设计

N2.3.1 主界面

本系统主界面采用框架网页实现，下面先给出各前端网页的 html 源码。

1. 启动页

启动页面为 index.html，代码如下：

```

<html>
<head>
    <title>学生成绩管理系统</title>
</head>

```

```
<body topMargin="0" leftMargin="0" bottomMargin="0" rightMargin="0">
  <table width="675" border="0" align="center" cellpadding="0" cellspacing="0" style="width: 778px;">
    <tr>
      <td></td>
    </tr>
    <tr>
      <td><iframe src="main_frame.html" width="790" height="313"></iframe></td>
    </tr>
    <tr>
      <td></td>
    </tr>
  </table>
</body>
</html>
```

页面上、中、下三部分，其中上、下两部分都只是一张图片，中间部分为一框架页（加黑代码为源文件名），运行时往框架页中加载具体的导航页和相应功能界面。

2. 框架页

框架页为 `main_frame.html`，代码如下：

```
<html>
<head>
  <meta http-equiv="Content-type" content="text/html; charset=GB2312"/>
  <title>学生成绩管理系统</title>
</head>
<frameset cols="217,*">
  <frame frameborder=0 src="http://localhost:8080/xscj" name="frmleft" scrolling="no" noresize>
  <frame frameborder=0 src="body.html" name="frmmain" scrolling="no" noresize>
</frameset>
</html>
```

其中，加黑处“`http://localhost:8080/xscj`”默认装载的是系统导航页 `main.jsp`（因之前在 `web.xml` 文件中已配置了 `<welcome-file-list>` 元素的 `<welcome-file>`），页面装载后位于框架左区。框架右区则用于显示各个功能界面，初始默认为 `body.html`，源码如下：

```
<html>
<head>
  <title>内容网页</title>
</head>
<body topMargin="0" leftMargin="0" bottomMargin="0" rightMargin="0">
  
</body>
</html>
```

这只是一个填充了背景图片的空白页，在运行时，系统会根据用户操作，往框架右区中动态加载不同功能的 JSP 页面来替换该页。在项目 `\WebRoot` 目录下创建 `images` 文件夹，其中放入用到的三幅图片资源：“学生成绩管理系统.gif”、“底端图片.gif”和“主页.gif”。右击项目名，从弹出的菜单中选择【Refresh】刷新。

N2.3.2 功能导航

本系统的导航页上有两个按钮，单击后可分别进入“学生管理”和“成绩管理”两个不同功能的

界面，如图 N2.16 所示。

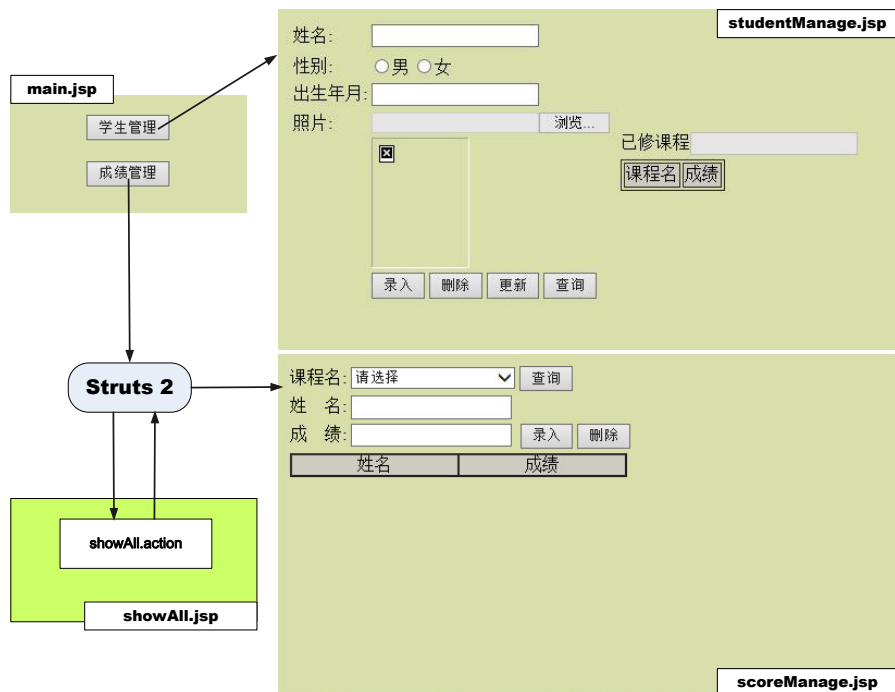


图 N2.16 功能导航

其中，“成绩管理”界面需要预先加载“课程名”下拉列表，这通过 showAll.jsp 页面上的一个 Action (showAll) 来实现，当单击【成绩管理】按钮时会触发这个 Action，在 Struts 2 控制下，调用相应的程序模块来实现加载功能，完成后再由 Struts 2 控制页面跳转到“成绩管理”功能界面 (scoreManage.jsp)。

源文件 main.jsp 实现功能导航页面，代码如下：

```
<%@ page language="java" pageEncoding="gb2312"%>
<html>
<head>
<title>功能选择</title>
</head>
<body bgcolor="D9DFAA">
<table bgcolor="D9DFAA" width="200" height="85">
<tr>
<td align="center">
<input type="button" value="学生管理" onclick="parent.frmmain.location='studentManage.jsp'">
</td>
</tr>
<tr>
<td align="center">
<input type="button" value="成绩管理" onclick="parent.frmmain.location='showAll.jsp'">
</td>
</tr>
</table>
</body>
</html>
```

其中，加黑处是两个导航按钮分别要定位到的 JSP 源文件：studentManage.jsp 实现“学生管理”功能界面（具体实现将在稍后给出），showAll.jsp 上安置了一个 Action（showAll），它的功能是往“成绩管理”界面上的“课程名”下拉列表中加载所有课程的名称供用户选择。

编写 showAll.jsp 代码如下：

```
<%@ page language="java" pageEncoding="utf-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
    <title>加载课程</title>
</head>
<body bgcolor="D9DFAA">
    <s:action name="showAll" executeResult="true"/>
</body>
</html>
```

打开 IE，在地址栏中输入 http://localhost:8080/xscj/index.html，显示如图 N2.17 所示的页面。



图 N2.17 “学生成绩管理系统”主页

在 src 下创建 struts.xml 文件，它是 Struts 2 的核心配置文件，负责管理各 Action 控制器到 JSP 页间的跳转，配置如下：

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
    <package name="default" extends="struts-default">
        <!-- 加载课程名 -->
        <action name="showAll" class="org.easybooks.xscj.action.ScoreAction" method="showAll">
            <result name="result">/scoreManage.jsp</result>
        </action>
    </package>
    <constant name="struts.multipart.saveDir" value="/tmp"></constant>
    <constant name="struts.enable.DynamicMethodInvocation" value="true" />
</struts>
```

配置文件中定义了 name 为 showAll 的 Action。当客户端发出 showAll.actionURL 请求时，Struts 2 会

根据 class 属性调用相应 Action 类（这里是 org.easybooks.xscj.action 包中的 ScoreAction 类）。method 属性指定该类中有一个 showAll()方法，将常量 struts.enable.DynamicMethodInvocation 的值设为 true，Struts 2 就会自动调用此方法来处理用户的请求，处理完后，该方法返回“result”字符串，请求被转发到/scoreManage.jsp 页（即“成绩管理”界面）。

N2.4 学 生 管 理

N2.4.1 界面设计

“学生管理”功能界面，如图 N2.18 所示。



图 N2.18 “学生管理”功能界面

它由源文件 studentManage.jsp 实现，代码如下：

```
<%@ page language="java" pageEncoding="utf-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<html>
<head>
    <title>学生管理</title>
</head>
<body bgcolor="D9DFAA">
<s:set name="student" value="#request.student"/>
<s:form name="frm" method="post" enctype="multipart/form-data">
    <table>
        <tr>
            <td>
                <table>
                    <tr>
                        <td>姓名:</td>
                        <td><input type="text" name="xm" value="#student.xm"/></td>
                    </tr>
                    <tr>
                        <td>性别:</td>
                        <td><s:radio list="{ '男','女' }" label="性别" name="student.xb" value=
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/058055056051006100>