# 实训1 Servlet及MVC

**一、实训目的**

1. 掌握Servlet的核心技术。
2. 能够基于JSP、JavaBean和Servlet实现MVC架构。
3. **实训仪器**

1．需配备安装有Windows7及以上操作系统的计算机，每人一台。

**三、实训原理**

MVC模式中，M是指model业务模型，V是指view用户界面，C则是controller控制器，使用MVC的目的是将M和V的实现代码分离，从而使同一个程序可以使用不同的表现形式。其中，View的定义比较清晰，就是用户界面。

1. M:模型层。

Model 通常由 JavaBean 来充当，所以业务逻辑、数据访问逻辑都在 Model 中实现。

2.V视图层。

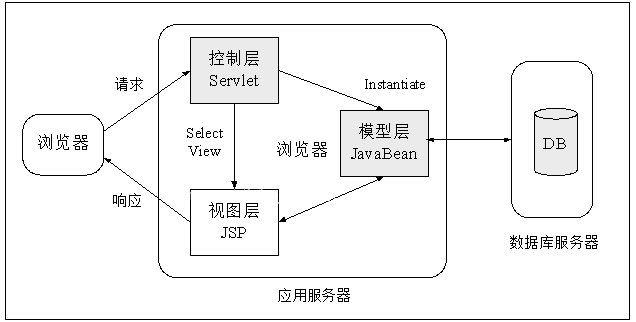
JSP 仅能作为表现层（View）技术，作用有两点：

1. 负责收集用户请求参数。

2. 将应用的处理结果、状态数据呈现给用户。

3.C:控制层。

Servlet 则仅充当控制器（Controller）角色，用户类似于调度员：（所以用户的请求发送给 Servlet , Servlet 调用 Model 来处理用户请求，并调用 JSP 来呈现处理结果;）



4.Servlet

Servlet 是在服务器上运行的小程序。Servlet 的主要功能在于交互式地浏览和修改数据，生成动态 Web 内容。这个过程为：

1. 客户端发送请求至服务器端；
2. 服务器将请求信息发送至 Servlet；
3. Servlet 生成响应内容并将其传给服务器。响应内容动态生成，通常取决于客户端的请求；
4. 服务器将响应返回给客户端。

为了简化 Servlet 的配置，Servlet 3.0 中增加了注解支持，例如：@WebServlet、@WebInitParm 、@WebFilter 和 @WebLitener 等，这使得 web.xml 从 Servlet 3.0 开始不再是必选项了。  
 @WebServlet 用于将一个类声明为 Servlet，该注解会在部署时被容器处理，容器根据其具体的属性配置将相应的类部署为 Servlet。该注解具有下表给出的一些常用属性。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 属性名 | 类型 | 标签 | 描述 | 是否必需 |
| name | String | <servlet-name> | 指定 Servlet 的 name 属性。 如果没有显式指定，则取值为该 Servlet 的完全限定名，即包名+类名。 | 否 |
| value | String[ ] | <url-pattern> | 该属性等价于 urlPatterns 属性，两者不能同时指定。 如果同时指定，通常是忽略 value 的取值。 | 是 |
| urlPatterns | String[ ] | <url-pattern> | 指定一组 Servlet 的 URL 匹配模式。 | 是 |
| loadOnStartup | int | <load-on-startup> | 指定 Servlet 的加载顺序。 | 否 |
| initParams | WebInitParam[ ] | <init-param> | 指定一组 Servlet 初始化参数。 | 否 |
| asyncSupported | boolean | <async-supported> | 声明 Servlet 是否支持异步操作模式。 | 否 |
| description | String | <description> | 指定该 Servlet 的描述信息。 | 否 |
| displayName | String | <display-name> | 指定该 Servlet 的显示名。 | 否 |

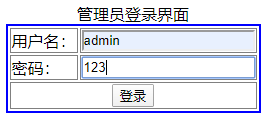
举例：

@Web​Servlet("/MyServlet")

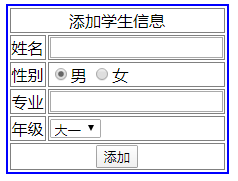
@WebServlet(name="AnnotationServlet",urlPatterns="/ MyServlet ")

**四、实训内容**

设计程序，以管理员身份登录学生信息管理系统并进行学生信息管理。

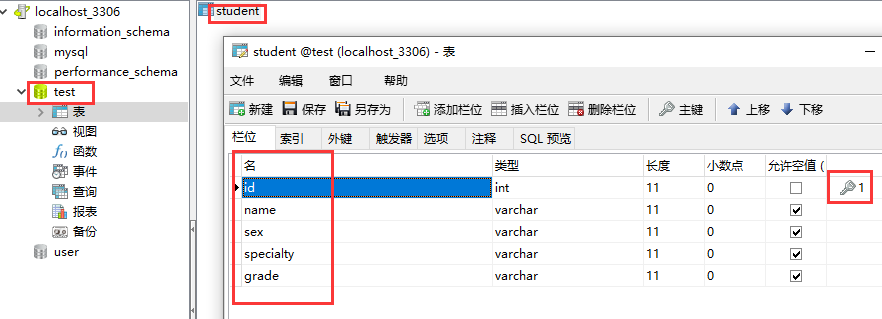




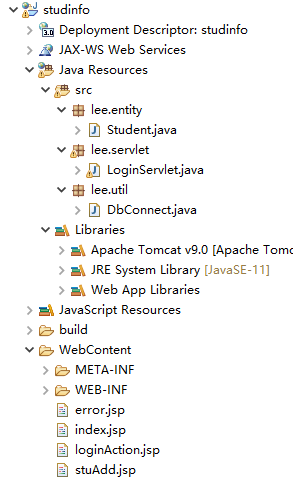


**五、实训步骤**

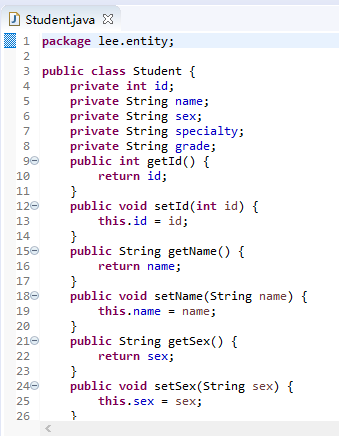
1、设计数据库



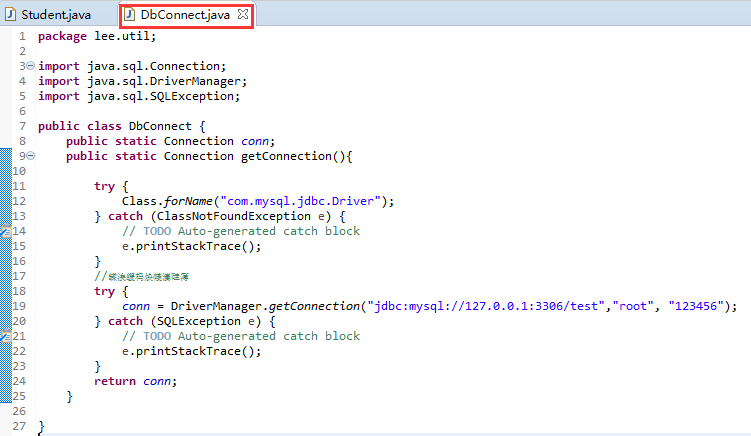
2、创建项目架构



3、定义模型层：实体类

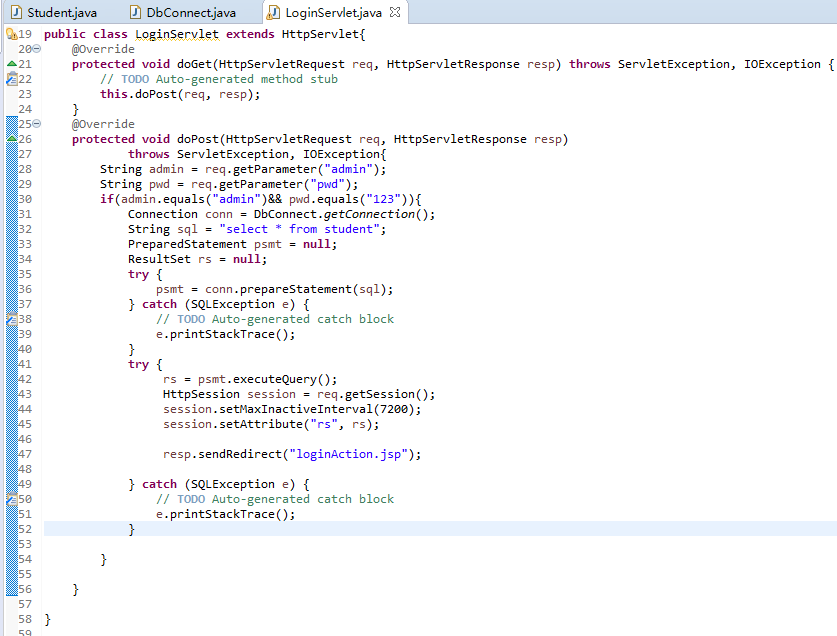


4、定义模型层：数据库连接工具



5、定义控制器：登录业务的servlet

注意Servlet的类定义上，应当加注解，@Web​Servlet("/loginServlet")。



6、定义视图层：管理员登录界面



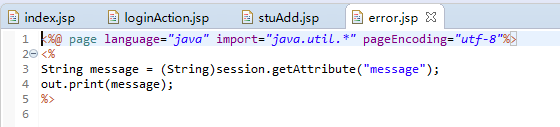
7、定义视图层：管理员登录后，学生信息管理界面



8、定义视图层：添加学生信息界面



9、定义视图层：信息错误页面



**六、数据记录及处理**

1．记录各类输入数据的输出。

**七、实训注意事项**

1．注意中英文乱码的问题。

**八、思考题**

1．继续完善项目，使学生信息的插入、修改和删除功能按照MVC的思想予以实现。