

# 轻量级 J2EE 框架应用

## E 1 A Simple Controller

学号：SA18225433

姓名：杨帆

报告撰写时间：2018/11/25

# 1.主题概述

## 1、MVC 模式。

Mvc 全称为 model、view、controller，是用于应用程序开发的一种规范。

Model 层为模型层，表示一个能存取数据的对象或 Java POJO，只需要定义一次，就能被多个视图重用；

View 层为视图层，表示能够被用户看到且能与之交互的界面；

Controller 层为控制层，作用于 model 层和 view 层，控制数据进入模型层中进行处理，当数据修改后控制视图层更新视图，使得模型层与视图层分离。

该模式具有低耦合，高可重用性，部署快，高可维护性的优点。

## 2、MVC 中 Controller 的实现。

最典型的 MVC 模式的实现即为 jsp+servlet+javabean 模式，本次作业中使用到了 servlet 作为 Controller 层的实现，servlet 是运行在 Web 服务器上的程序，接受服务器传入的请求，处理请求，并返回响应给客户。

## 3、Http request 在 web Container 中的处理流程。

Tomcat web 服务器分为 Connector 和 Container 两个组件，其中 Container 作为装载 Servlet 的容器，当向该 web 服务器发送一个 Http request 请求，根据请求的 URL 去 web.xml 中获得 Servlet 的描述信息并访问对应的 Servlet，如果该 Servlet 还没被创建则会装载该 Servlet 并实例化，同时 web 服务器的 Connector 组件会解析该 Http 请求，封装成 HttpRequest 类型的对象，并创建 HttpResponse 类型的对象，并传 HttpRequest 和 HttpResponse 对象到 Container 中的 Servlet 的 service()方法中处理请求，请求中的数据可以通过 HttpRequest 对象调用对应的方法获得，当处理结束返回响应信息，Servlet 会将响应信息封装到 HttpResponse 对象中返回给 web 服务器。

## 4、比较 JSP Model 1 与 Model 2 架构，说明各自的优缺点及适用场景。

Model 1 架构为 JSP+javabean，javabean 充当数据模型，用来存取数据对象，而 JSP 负责接受用户请求，处理请求，并返回响应结果的功能。Model 2 架构为 JSP+Servlet+javabean，是典型的 MVC 架构模式，JSP 充当视图层，负责与用户的交互，数据的转递与更新的操作，Servlet 则作为 Controller 负责创建 Javabean 对象来处理数据，并根据用户的动作返回 javabean 中动态的数据到 JSP 页面中显示。Model 1 的优点是架构分层简单，容易理解实现，缺点是 JSP 页面中既有控制页面显示信息的功能代码又有与 javabean 完成业务逻辑交互的 java 代码，影响页面的加载速度，不便于后期的维护工作，适用于小型项目的快速开发。Model 2 的优点是将视图与逻辑控制分离开来，便于开发工作的分工进行，也便于后期的维护，比较适用于大型项目的开发。

## 2.假设

本次作业能够实现使用 `eclipse` 实现一个简单的 `Controller` 类并打包为 `jar` 包，并构建测试工程进行测试。

## 3. 实现或证明

### 1. 实现成果

e1: <https://github.com/saaaaaail/J2eee1>

### 2. 搭建 Java Web 开发环境(Eclipse, JDK, Tomcat)

3. 新建 Java 工程 SimpleController，在该工程下新建源码包 sc.ustc.controller，并在此包中新建 HttpServlet 子类 SimpleController，实现 SimpleController 类的功能如下：

3.1 实现 doPost()方法，能够将 Http 请求默认响应输出为：

```
<html>
  <head>
    <title>SimpleController</title>
  </head>
  <body>欢迎使用SimpleController!</body>
</html>
```

重写 doPost()方法，设置 request 对象字符集，设置 response 对象字符集，获得 PrintWriter 对象，使用 write 方法向浏览器返回上面的字符串信息，

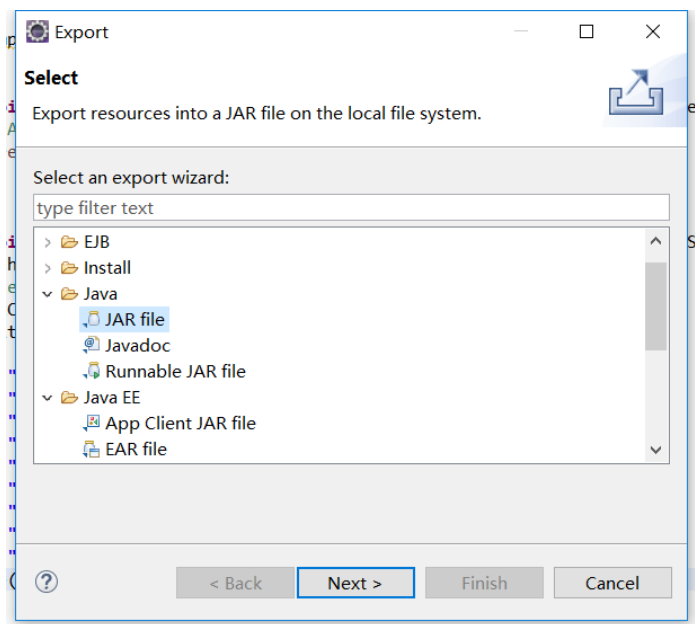
```
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
    req.setCharacterEncoding("utf-8");
    //resp.setContentType("text/html;charset=utf-8");
    resp.setCharacterEncoding("utf-8");
    PrintWriter pw = resp.getWriter();
    String s = "<!DOCTYPE html>\r\n" +
        "<html>\r\n" +
        "<head>\r\n" +
        "<meta charset=\"UTF-8\">\r\n" +
        "<title>SimpleController</title>\r\n" +
        "</head>\r\n" +
        "<body>\r\n" +
        "欢迎使用SimpleController! ! \r\n" +
        "</body>\r\n" +
        "</html>";
    pw.write(s);
}
```

### 3.2 实现 doGet()方法，调用 doPost()

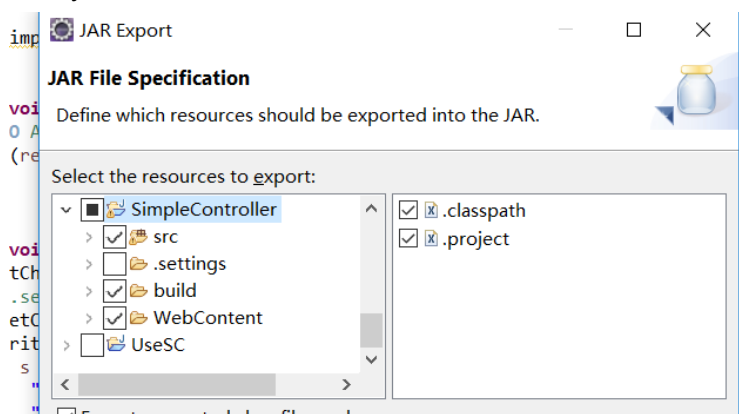
```
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
    // TODO Auto-generated method stub
    doPost(req, resp);
}
```

### 3.3 将 SimpleController 工程打包导出为 simple-controller.jar

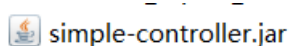
使用 eclipse 导出 jar 包，右击项目工程名，选择 Export 项，



选择 jar file 文件点击 next,

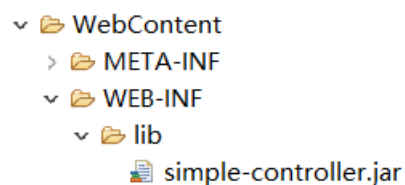


选择要导出的文件，然后点击 finish，生成 jarfile，



#### 4. 新建 Java Web 工程 UseSC

4.1 将步骤 2 中的 simple-controller.jar 放入到 UseSC 工程的 lib 库中  
将 jar 包放到 WebContent 目录下 WEB-INF 目录的 lib 目录中，



4.2 在 UseSC 工程的配置文件 web.xml 中，定义 servlet 结点名称为 sc，将其 servlet-class 指向 sc.ustc.controller.SimpleController，如下：

```
<servlet>
  <servlet-name>sc</servlet-name>
  <servlet-class>sc.ustc.controller.SimpleController</servlet-class>
</servlet>
```

4.3 在 UseSC 工程的配置文件 web.xml 中，定义 servlet-mapping 结点， 将其

`servlet-name` 指向 `sc`，声名 `url-pattern` 为 `/*`，如下：

```
<servlet-mapping>
  <servlet-name>sc</servlet-name>
  <url-pattern>/*</url-pattern>
</servlet-mapping>
```

4.4 在 UseSC 工程中定义 `welcome.html` 并设置为该工程的欢迎页面  
WebContent 目录下定义 `welcome.html`，并编写其中的内容，

```
<welcome-file>welcome.html</welcome-file>
```

```
welcome.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5 <title>SimpleController</title>
6 </head>
7 <body>
8 欢迎使用SimpleController!
9 </body>
10 </html>
```

5. 将 UseSC 工程打包为 `.war`，并部署在 Tomcat 中。部署完成后，启动 Tomcat，并在浏览器输入：`http://host:port/UseSC`，观察页面是否为“欢迎使用 SimpleController!”，如果否，调试代码，直到出现该结果

打包为 `.war` 包，并部署到 Tomcat 的 `webapps` 目录下，启动 tomcat，

UseSC.war

使用浏览器访问，

localhost:8080/UseSC/

欢迎使用SimpleController!

## 6. 结论

对主题的总结，结果评论，发现的问题，或你的建议和看法。如 MVC 中 Controller 优点与缺点，个人看法（文字、图标、代码辅助等）

本次作业了解了什么是 MVC 模式的，MVC 的优点和缺点，学习了如何自己使用 Servlet 构建一个 Controller 类，并部署到服务器上运行，并学习了将一个程序部署为 jar 包，添加到另外一个项目工程依赖库中去运行，学习了 tomcat 服务器与 Servlet 的关系，学习了当一个 http 请求传到服务器里面后，在 Servlet 中具体的处理流程，以及最后返回响应给服务器，学习了 JSP 的 Model 1 和 Model 2，并对两种架构模式进行了比较分析其优点和缺点。

## 7.参考文献

- 1.[java 中 jar 包内的类访问 jar 包内部的资源文件的路径问题](#)
2. [Web 服务请求处理过程全解](#)
- 3.[MVC 简介](#)
- 4.[MVC 模式](#)
- 5.[Java Web\(一\) Servlet 详解](#)
- 6.[web 请求：servlet 处理请求](#)
- 7.[JSP 的 Model1 和 Model2 架构模式的分析比较](#)
- 8.[浅析 Java 开发模式—Model1、Model2 和三层](#)