

轻量级 J2EE 框架应用

E 1 A Simple Controller

学号：SA18225235

姓名：刘瑾瑾

报告撰写时间：2018 / 11 / 29

1.主题概述

简要介绍主题的核心内容，如 MVC，及 MVC 中 Controller 的作用与实现

MVC 是一种使用 MVC（Model View Controller 模型-视图-控制器）设计创建 Web 应用程序的模式：

- Model（模型）表示应用程序核心（比如数据库记录列表）。
- View（视图）显示数据（数据库记录）。
- Controller（控制器）处理输入（写入数据库记录）。

MVC 模式同时提供了对 HTML、CSS 和 JavaScript 的完全控制。

Model（模型）是应用程序中用于处理应用程序数据逻辑的部分。

通常模型对象负责在数据库中存取数据。

View（视图）是应用程序中处理数据显示的部分。

通常视图是依据模型数据创建的。

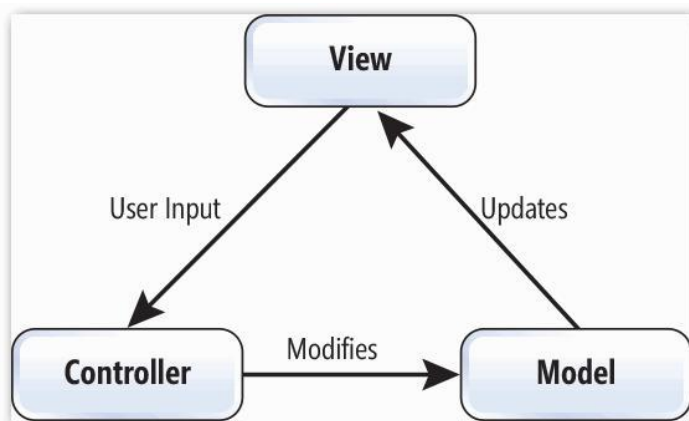
Controller（控制器）是应用程序中处理用户交互的部分。

通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

MVC 分层有助于管理复杂的应用程序，因为您可以在一个时间内专门关注一个方面。例如，您可以在不依赖业务逻辑的情况下专注于视图设计。同时也让应用程序的测试更加容易。

MVC 分层同时也简化了分组开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。

最典型的 MVC 就是 Jsp + Servlet + JavaBean 的模式



Controller 可以决定要显示哪一个 View。

Controller 负责定义和调用 Model。

控制器接受用户的输入并调用模型和视图去完成用户的需求。当 web 页面中的超链接和发送 HTML 表单时，控制器本身不输出任何东西和做任何处理。它只接受请求并决定调用哪个模型构件去处理请求，然后决定用哪个视图来显示模型处理返回的数据。

2.假设

主题内容所参照的假设条件，或假定的某故事场景，如调试工具或软硬件环境

软件环境：Windows 10、JDK1.8、MyEclipse、Tomcat

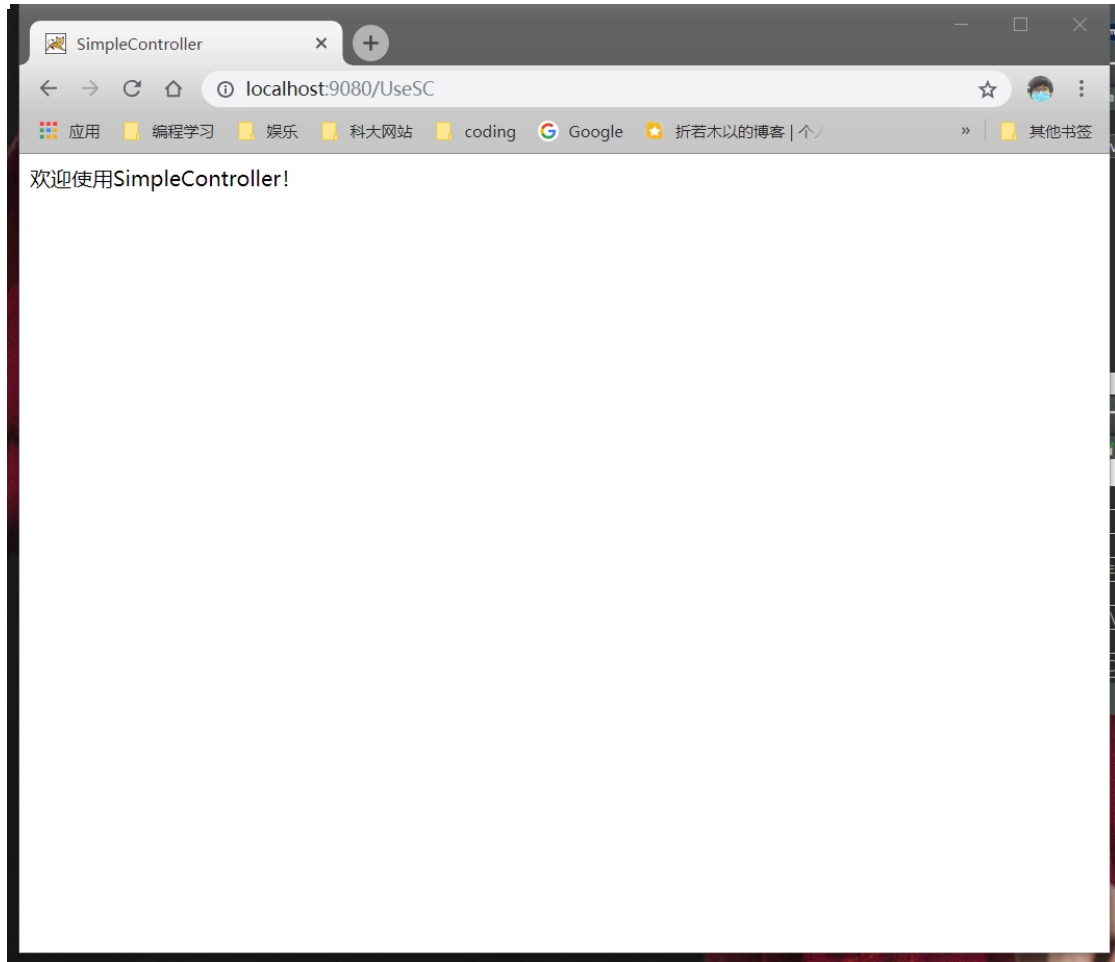
硬件环境：个人笔记本电脑

调试工具：Chrome 浏览器

3.实现或证明

对主题内容进行实验实现，或例举证明，需描述实现过程及数据。如对 MVC 中 Controller 功能的实现及例证（图示、数据、代码等）

源代码附在压缩文件中，运行结果（8080 已被使用）：



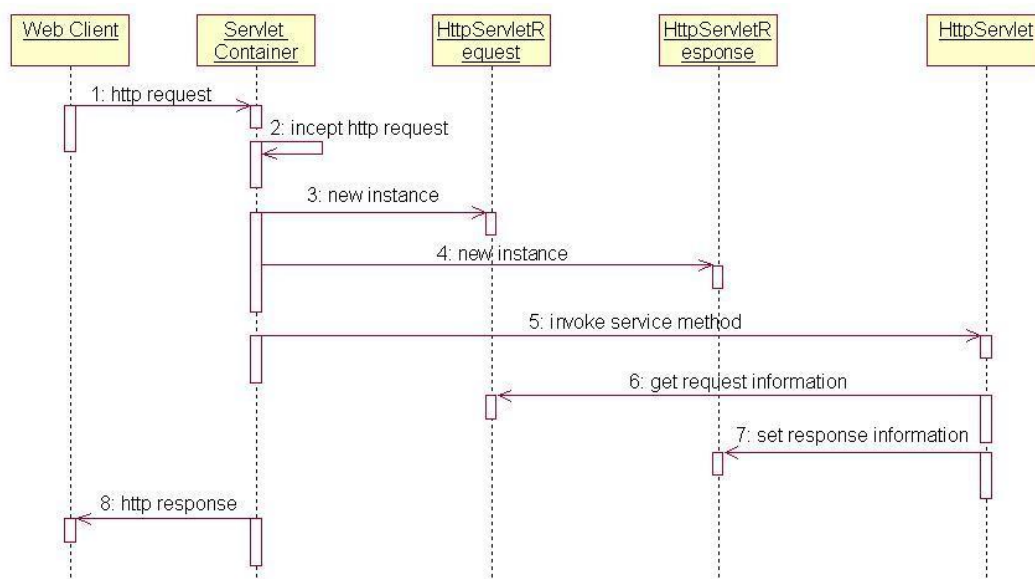
4. 结论

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）

附加提问回答：

5、

当 Web Container 接收到一个 HTTP 请求(request)，会返回一个 HTTP 响应(response)；



Servlet 工作原理时序图

- ①Web Client 向 Web Container（Tomcat）发出 Http 请求；
- ②Web Container 接收 Web Client 的请求；
- ③Web Container 创建一个 HttpRequest 对象，将 Web Client 请求的信息封装到这个对象中；
- ④Web Container 创建一个 HttpResponse 对象；
- ⑤Web Container 调用 HttpServlet 对象的 service 方法，把 HttpRequest 对象与 HttpResponse 对象作为参数传给 HttpServlet 对象；
- ⑥HttpServlet 调用 HttpRequest 对象的有关方法，获取 Http 请求信息；
- ⑦HttpServlet 调用 HttpResponse 对象的有关方法，生成响应数据；
- ⑧Web Container 把 HttpServlet 的响应结果传给 Web Client；

6、

JSP 的 Model1 和 Model2 是 sun 公司为了更好地指导 Web 开发人员进行 JavaWeb 开发提出的两种架构模式，用专业术语表达就是 JSP Model 1 和 JSP Model 2，它们的本质区别在于处理批量请求的位置不同。

一、Model1 架构模式

1、Model1 架构模式简介

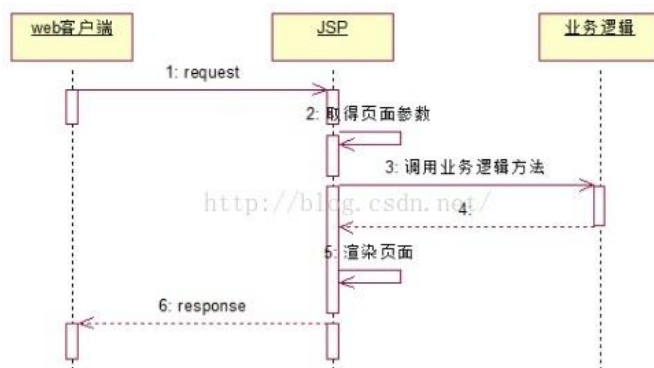
Model1 应用了 JavaWeb 开发当中的两种技术，即 JSP 和 JavaBean 技术，所以 Model1

就是 JSP+JavaBean 模式。其中 JSP 需要独自完成响应用户的请求并将处理结果返回给用户，同时还需要完成流程的控制处理；而 JavaBean 进行辅助，如保存从数据库中查询出来的数据等。

JavaBean 是一种使用 Java 语言编写的可重用组件。为写成 JavaBean，类必须是具体的和公共的，并且具有无参数的构造器。JavaBean 通过提供符合一致性设计模式的公共方法将内部域的成员属性暴露给用户。用户可以使用 JavaBean 将功能、处理、值、数据库访问和其他任何可以用 Java 代码创造的对象进行打包，并且其他的开发者可以通过内部的 JSP 页面、Servlet、其他 JavaBean、Applet 程序或者应用来使用这些对象。

JavaBean 可分为两种：一种是有用户界面（UI，User Interface）的 JavaBean；还有一种是没有用户界面，主要负责处理事务（如数据运算、操纵数据库）的 JavaBean。JSP 通常访问的是后一种 JavaBean。

2、Model1 的体系结构图



下面对上图的 Model1 体系结构图进行分析。

- (1) 首先 Web 客户端向 JSP 页面发出 request 请求；
- (2) 在 JSP 页面取得请求所需的页面参数；
- (3) 通过 JSP 调用业务逻辑层的方法并返回结果；
- (4) 在 JSP 中完成网页页面的渲染；
- (5) 通过 JSP 向 Web 客户端返回处理结果。

3、Model1 架构模式的优缺点

- (1) 优点：架构简单，比较适合小型项目的开发。基本上使用 JSP+持久层就可以完成一个小型项目的开发。
- (2) 缺点：JSP 页面有控制页面显示的代码又有与后台业务逻辑交互的 Java 代码，职责不单一，页面负担很重，影响页面运行速度；而且不方便后期的维护。如果不加选择地随意运用 Model 1，会导致 JSP 页面内被嵌入大量的脚本片段或 Java 代码，特别是当需要处理的请求量很大时，情况会更为糟糕。

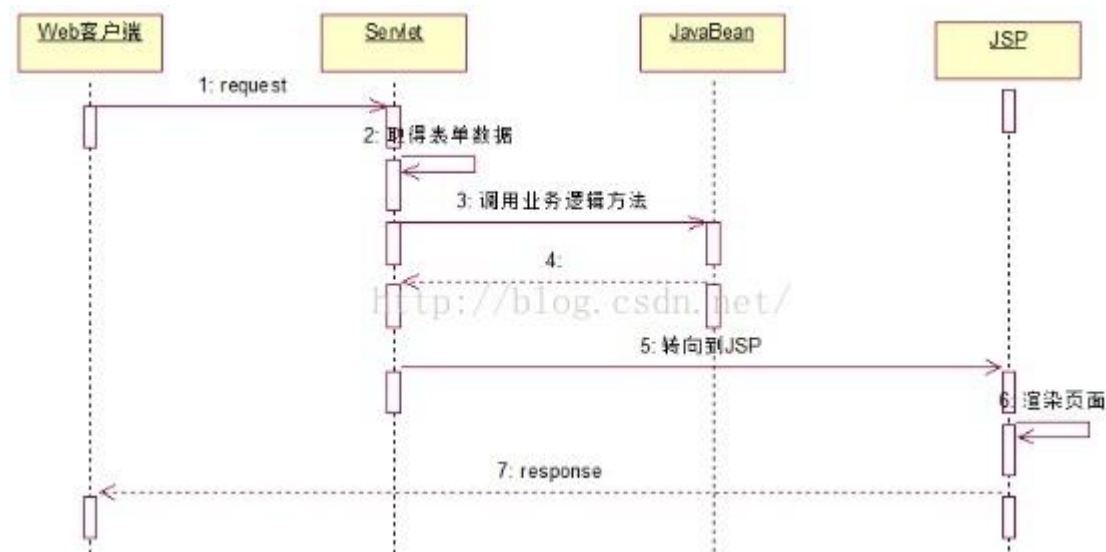
二、Model2 架构模式

1、Model2 架构模式简介

Model 2 架构模式是一种把 JSP 与 Servlet 联合使用来实现动态内容服务的方法，所以 Model2 就是 JSP+Servlet+JavaBean 模式。它吸取了两种技术各自的突出优点，使用 JSP 生成表达层的内容，使用 Servlet 完成深层次的处理任务。在 Model 2 中，Servlet 充当控制者的角色，负责管理对请求的处理，创建 JSP 页面所需的 JavaBean 和对象，同时根据用户的动作决定把哪个 JSP 页面传给用户。特别要注意，在 JSP 页面内没有处理逻辑，它仅负责检索原先由 Servlet 创建的对象或 JavaBean，然后从 Servlet 中提取动态内容插入到静态模板中进行页面显示。

所以简单来说，在 Model 2 中，JSP 进行页面显示，JavaBean 进行逻辑处理，Servlet 完成逻辑控制。

2、Model2 的体系结构图



下面对上图的 Model2 体系结构图进行分析。

- (1) 首先 Web 客户端向 Servlet 发出 request 请求；
- (2) 通过 Servlet 取得请求所需的表单数据；
- (3) Servlet 向业务逻辑层调用业务逻辑方法并返回所需的数据；
- (4) Servlet 再重定向转到 JSP 页面，在 JSP 页面完成页面渲染；
- (5) 通过 JSP 通过 response 向 Web 客户端返回处理结果。

3、Model2 架构模式的总结

- (1) JavaBean(Model)——模型层：主要职责是完成业务逻辑和保存数据的状态；
- (2) JSP(View)——视图层，主要职责是完成页面显示；
- (3) Servlet(Controller)——控制器，主要职责是获取表单数据、调用业务逻辑和完成页面的渲染。

4、Model2 架构模式的优缺点

- (1) 优点：清晰地分离了表达和内容，明确了角色的定义以及开发者与网页设计者的分工，适合大型项目的开发。
- (2) 缺点：分层过多，不适合小型项目的开发。

5、Model1 与 Model2 的比较

Model1 在一定程度上实现了 MVC 的设计理念，即 JSP 将控制层与表示层合二为一，JavaBean 则为模型层。这样的话当然还是将控制层从 JSP 中单独划分出来比较好。

Model2 则是将控制层(Servlet)单独地划分出来了，专门负责业务流程的控制，响应页面的请求，创建所需的 JavaBean 实例，并将处理后的数据再返回给 JSP。这样层次分明，不同开发人员的职责分工明确，所以项目越复杂，使用 Model 2 架构模型的好处就越大。

5.参考文献

以上内容的理论知识点或技术点如果参考了网上或印刷制品，请在这里罗列出来

<http://www.runoob.com/aspnet/mvc-intro.html> MVC 教程

<https://www.jianshu.com/p/571c474279af> 作者：猿码道

<https://blog.csdn.net/chs2065/article/details/51935005> 作者：梦-彼岸 来源：CSDN