E 2

朱洪军

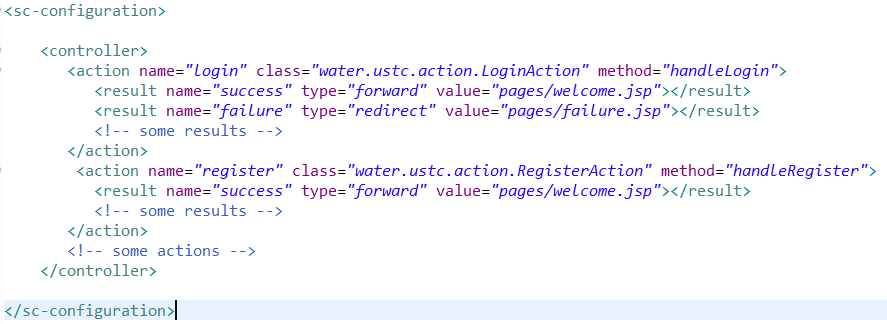
主题: A Simple Controller Based on Configuration File

内容:

1. 将 E1 中的控制器修改为基于配置文件的控制器。
   1. 将 E1 中的工程 UseSC 配置 web.xml 中定义的

servlet-mapping 修改为：可以拦截“\*.sc”类型的请求，如 Http：

//host/request\_action.sc

* 1. 在 UseSC 工程中新建源码包 water.ustc.action，在该包下声名 POJO 类 LoginAction、RegisterAction 等，每个 POJO 类可以有不同功能的方法，所有方法均返回 String 类型的结果。针对 POJO 类方法的返回值，分别自定义对应的视图 jsp 或 html
  2. 在 UseSC 工程中 src 下新建 controller.xml，在其中配置若干<action>与<result>。示例如下：
  3. 修改 SimpleController 工程的类 SimpleController 源码， 当一个 http request 请求访问 web container 资源时，由 doPost() 对请求进行处理：获取请求（action）的名称
  4. SimpleController 获取请求 action 名称后，解析使用

simple-controller.jar 库的工程（对于当前练习即为 UseSC 工程） 配置文件 controller.xml（XML 解析，SAX、Dom 或其他），查找对应 name 的 action。如果在 controller.xml 中找到，则解析该 action 的配置。如果没有找到，响应客户端信息为：不可识别的 action 请求。

* 1. SimpleController 查找到 http request 请求的 action 后， 利用其 class 属性实例化所指向的类（ Java 反射机制，

Reflection），并执行指定的 method 方法。

* 1. method 方法执行完毕后，返回字符串作为请求结果。

SimpleController 根据请求结果，查找此 action 中<result>结点的 name 属性，若找到，将 value 指向的资源按 type 所定义的方式返回到客户端。如果没有匹配的<result>，响应客户端为信息为：没有请求的资源。

* 1. 为了使得 SimpleController 类的代码简洁，建议自主添加 工 具 类 或 辅 助 类 完 成 以 上 功 能 。 重 新 打 包 导 出

SimpleController 工程为 simple-controller.jar，并将该 jar 添加为

UseSC 的 lib 库

* 1. 将 UseSC 部署在 tomcat 中测试，验证当浏览器请求对应的 action 时，是否能够响应正确的视图。如果有错，调试程序直到输出期望结果

1. 描述你对 Struts 2 控制器的理解，并参考资料，比较基于配

置的控制器和注解的控制器各自优缺点。

规则:

1. 在提交日期之前将成果提交至教学辅助系统
2. 提交文件格式为.pdf
3. 文件命名格式为：学号\_作业号。如 SA2018001\_E 2.pdf

参考资料：

* 1. Java Reflection: <https://docs.oracle.com/javase/tutorial/reflect/>
  2. XML Parser SAX: <http://www.saxproject.org/quickstart.html>
  3. XML Parser DOM: <http://www.w3schools.com/dom/dom_parser.asp>