

适用于初学者的struts简单案例分析Java认证考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/644/2021_2022__E9_80_82_E7_94_A8_E4_BA_8E_E5_c104_644486.htm

前言：虽然现在ssh开发满天飞，但是对于初学者来说，未必是好事，从基础学起，一步一个脚印，也许对他们来说更好。接下来的时间里，讲解一个struts的简单案例，从基础学起，有兴趣的朋友可以慢慢看来。

概要：Front Controller(模式) 说明：讲解struts具体例子之前，先说明一下Front Controller模式(前端控制器模式)。struts就是它的具体实现，得以扩展，并且是Model2结构的完整实现。Model2结构通过一个控制组件来接受请求，这个控制组件本身可以为请求提供服务，也可以把这项任务分配给其他一些组件。生成响应信息的任务接着被分配给适当的视图组件。实现这种要求的方式很多，Front Controller就是其中一种。实现前端控制器可以采取不同的策略，一种典型的策略是使用一个servlet问题：如果把所有的功能集中起来会出现一个大而臃肿的控制器组件，它将负责整个Web的相关处理。解决：办法之一是使用多个前端控制器，分别负责某一方面的处理任务。另一种解决方案是使用Command and Controller(命令与控制器)(稍候介绍) 下面是这种Front Controller模式示例程序框架实现：

```
public class FrontController
extends HttpServlet {
protected void
processRequest(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException {
RequestDispatcher
dispatcher = getServletContext().getRequestDispatcher(nextView).
dispatcher.forward(req, res). }
protected void
```

```
doGet(HttpServletRequest req, HttpServletResponse res) throws  
ServletException, IOException { processRequest(req, res). }  
protected void doPost(HttpServletRequest req, HttpServletResponse  
res) throws ServletException, IOException { processRequest(req,  
res). } }
```

在这里，它只是HttpServlet一个简单的扩充，默认实现了doGet()和doPost()方法，这两个方法负责把请求分配给另一个叫processRequest()的方法。这是为了保证不论发出什么请求，前端控制器都能为之提供服务。在processRequest()的方法中我们省去主要部分，但实际上前端控制器将先执行与请求关联的处理，然后再把它分配给一个视图组件来生成相关的响应信息。视图组件一般是JSP页面。控制器完成了自己的业务逻辑之后，它就可以通过RequestDispatcher把处理分配给一个JSP。到此你可能会提出一个问题。如果控制器要负责处理所有的请求，那么它怎么知道各请求都是什么以及如何处理呢?(Command and Controller，上面问题所提到的，现在开始讲解) Command and Controller策略 在Command and Controller策略中，处理每个具体请求的逻辑被转移到了一个单独的组件中。这些额外组件各表示一个具体的命令(或操作，既action)，并且组件中封装着执行相关操作的逻辑。前端控制器把对请求的处理分配给适当的命令组件。操作类 为了实现Command and Controller策略，首先需要定义位于控制器和操作组件之间的接口。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com