

用Spring更好地处理Struts动作三种整合 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/144/2021_2022__E7_94_A8Spring_E6_c104_144612.htm 与 Struts 相似，Spring 可以作为一个 MVC 实现。这两种框架都具有自己的优点和缺点，尽管大部分人同意 Struts 在 MVC 方面仍然是最好的。很多开发团队已经学会在时间紧迫的时候利用 Struts 作为构造高品质软件的基础。Struts 具有如此大的推动力，以至于开发团队宁愿整合 Spring 框架的特性，而不愿意转换成 Spring MVC。没必要进行转换对您来说是一个好消息。Spring 架构允许您将 Struts 作为 Web 框架连接到基于 Spring 的业务和持久层。最后的结果就是现在一切条件都具备了。在接下来的小窍门中，您将会了解到三种将 Struts MVC 整合到 Spring 框架的方法。我将揭示每种方法的缺陷并且对比它们的优点。一旦您了解到所有三种方法的作用，我将会向您展示一个令人兴奋的应用程序，这个程序使用的是这三种方法中我最喜欢的一种。三个小窍门接下来的每种整合技术（或者窍门）都有自己的优点和特点。我偏爱其中的一种，但是我知道这三种都能够加深您对 Struts 和 Spring 的理解。在处理各种不同情况的时候，这将给您提供一个广阔的选择范围。方法如下：使用 Spring 的 ActionSupport 类整合 Struts 使用 Spring 的 DelegatingRequestProcessor 覆盖 Struts 的 RequestProcessor 将 Struts Action 管理委托给 Spring 框架 装载应用程序环境 无论您使用哪种技术，都需要使用 Spring 的 ContextLoaderPlugin 为 Struts 的 ActionServlet 装载 Spring 应用程序环境。就像添加任何其他插件一样，简单地向您的 struts-config.xml 文件添加

该插件，如下所示：前面已经提到过，在下载部分，您能够找到这三个完全可使用的例子的完整源代码。每个例子都为 一个书籍搜索应用程序提供一种不同的 Struts 和 Spring 的整合方法。您可以在这里看到例子的要点，但是您也可以下载应用程序以查看所有的细节。窍门 1. 使用 Spring 的 ActionSupport 手动创建一个 Spring 环境是一种整合 Struts 和 Spring 的最直观的方式。为了使它变得更简单，Spring 提供了一些帮助。为了方便地获得 Spring 环境

，org.springframework.web.struts.ActionSupport 类提供了一个 getWebApplicationContext() 方法。您所做的只是从 Spring 的 ActionSupport 而不是 Struts Action 类扩展您的动作，如清单 1 所示：清单 1. 使用 ActionSupport 整合 Struts package

```
ca.nexcel.books.actions.import java.io.IOException.import
```

```
javax.servlet.ServletException.import
```

```
javax.servlet.http.HttpServletRequest.import
```

```
javax.servlet.http.HttpServletResponse.import
```

```
org.apache.struts.action.ActionError.import
```

```
org.apache.struts.action.ActionErrors.import
```

```
org.apache.struts.action.ActionForm.import
```

```
org.apache.struts.action.ActionForward.import
```

```
org.apache.struts.action.ActionMapping.import
```

```
org.apache.struts.action.DynaActionForm.import
```

```
org.springframework.context.ApplicationContext.import
```

```
org.springframework.web.struts.ActionSupport.import
```

```
ca.nexcel.books.beans.Book.import
```

```
ca.nexcel.books.business.BookService.public class SearchSubmit
```

```
extends ActionSupport { |(1) public ActionForward execute(
ActionMapping mapping, ActionForm form, HttpServletRequest
request, HttpServletResponse response) throws IOException,
ServletException { DynaActionForm searchForm =
(DynaActionForm) form. String isbn = (String)
searchForm.get("isbn"). //the old fashion way //BookService
bookService = new BookServiceImpl(). ApplicationContext ctx =
getWebApplicationContext(). |(2) BookService bookService =
(BookService) ctx.getBean("bookService"). |(3) Book book =
bookService.read(isbn.trim()). if (null == book) { ActionErrors
errors = new ActionErrors().
errors.add(ActionErrors.GLOBAL_ERROR,new ActionError
("message.notfound")). saveErrors(request, errors). return
mapping.findForward("failure") . } request.setAttribute("book",
book). return mapping.findForward("success"). }} 让我们快速思
考一下这里到底发生了什么。在 (1) 处，我通过从 Spring 的
ActionSupport 类而不是 Struts 的 Action 类进行扩展，创建了一个新的 Action。在 (2) 处，我使用
getWebApplicationContext() 方法获得一个 ApplicationContext
。为了获得业务服务，我使用在 (2) 处获得的环境在 (3) 处查
找一个 Spring bean。这种技术很简单并且易于理解。不幸的
是，它将 Struts 动作与 Spring 框架耦合在一起。如果您想替换
掉 Spring，那么您必须重写代码。并且，由于 Struts 动作不在
Spring 的控制之下，所以它不能获得 Spring AOP 的优势。当
使用多重独立的 Spring 环境时，这种技术可能有用，但是在
大多数情况下，这种方法不如另外两种方法合适。窍门 2. 覆
```

盖 RequestProcessor 将 Spring 从 Struts 动作中分离是一个更巧妙的设计选择。分离的一种方法是使用 org.springframework.web.struts.DelegatingRequestProcessor 类来覆盖 Struts 的 RequestProcessor 处理程序，如清单 2 所示：清单 2. 通过 Spring 的 DelegatingRequestProcessor 进行整合 |(1) 我利用了 标记来用 DelegatingRequestProcessor 覆盖默认的 Struts RequestProcessor。下一步是在我的 Spring 配置文件中注册该动作，如清单 3 所示：清单 3. 在 Spring 配置文件中注册一个动作 |(1) 注意：在 (1) 处，我使用名称属性注册了一个 bean，以匹配 struts-config 动作映射名称。SearchSubmit 动作揭示了一个 JavaBean 属性，允许 Spring 在运行时填充属性，如清单 4 所示：清单 4. 具有 JavaBean 属性的 Struts 动作

```
package ca.nexcel.books.actions.import java.io.IOException.import
```

```
javax.servlet.ServletException.import
```

```
javax.servlet.http.HttpServletRequest.import
```

```
javax.servlet.http.HttpServletResponse.import
```

```
org.apache.struts.action.Action.import
```

```
org.apache.struts.action.ActionError.import
```

```
org.apache.struts.action.ActionErrors.import
```

```
org.apache.struts.action.ActionForm.import
```

```
org.apache.struts.action.ActionForward.import
```

```
org.apache.struts.action.ActionMapping.import
```

```
org.apache.struts.action.DynaActionForm.import
```

```
ca.nexcel.books.beans.Book.import
```

```
ca.nexcel.books.business.BookService.public class SearchSubmit
```

```
extends Action { private BookService bookService. public
```

```
BookService getBookService() { return bookService. } public void
setBookService(BookService bookService) { | (1) this.bookService =
bookService. } public ActionForward execute( ActionMapping
mapping, ActionForm form, HttpServletRequest request,
HttpServletRequest response) throws IOException,
ServletException { DynaActionForm searchForm =
(DynaActionForm) form. String isbn = (String)
searchForm.get("isbn"). Book book =
getBookService().read(isbn.trim()). |(2) if (null == book) {
ActionErrors errors = new ActionErrors().
errors.add(ActionErrors.GLOBAL_ERROR,new
ActionError("message.notfound")). saveErrors(request, errors).
return mapping.findForward("failure") . }
request.setAttribute("book", book). return
mapping.findForward("success"). }} 在清单 4 中，您可以了解到
```

如何创建 Struts 动作。在 (1) 处，我创建了一个 JavaBean 属性。DelegatingRequestProcessor 自动地配置这种属性。这种设计使 Struts 动作并不知道它正被 Spring 管理，并且使您能够利用 Spring 的动作管理框架的所有优点。由于您的 Struts 动作注意不到 Spring 的存在，所以您不需要重写您的 Struts 代码就可以使用其他控制反转容器来替换掉 Spring。

DelegatingRequestProcessor 方法的确比第一种方法好，但是仍然存在一些问题。如果您使用一个不同的 RequestProcessor，则需要手动整合 Spring 的 DelegatingRequestProcessor。添加的代码会造成维护的麻烦并且将来会降低您的应用程序的灵活性。此外，还有过一些使用一系列命令来代替 Struts

RequestProcessor 的传闻。这种改变将会对这种解决方法的使用寿命造成负面的影响。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