

J和JSF双剑合并打造完美Web应用 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/252/2021\\_2022\\_J\\_\\_E5\\_92\\_8C](https://www.100test.com/kao_ti2020/252/2021_2022_J__E5_92_8C)

JSF\_E5\_8F\_c104\_252488.htm Java在最近几年逐渐升温，随着Java SE 5和Java SE 6的推出，Java的未来更显得无比辉煌。但以Java为基础的JSP在Java SE 5推出之前却一直抬不起头来，这最重要的原因就是JSP虽然功能十分强大，但最大的优点也是它的最大缺点，功能强大就意味着复杂，尤其是设计前端界面的可视化工具不多，也不够强大。因此，设计JSP页面就变得十分复杂和繁琐。不过，在Java SE 5推出的同时，Sun为了简化JSP的开发难度，推出了新的JavaServer Faces(简称JSF)规范。从而使JSP走上了康庄大道。

### 一、什么是JSF

JSF和JSP是一对新的搭档。JSP是用于后台的逻辑处理的技术，而JSF恰恰相反，是使开发人员能够快速的发展基于Java的Web应用程序的技术，是一种表现层技术。目前，JSF1.2已经正式作为一个标准加入了Java EE 5中。作为一种高度组件化的技术，开发人员可以在一些开发工具的支持下，实现拖拉式编辑操作，用户只需要简单的将JSF组件拖到页面上，就可以很容易的进行Web开发了。这是其作为一种组件化的技术所具有的最大好处，我们能用的组件不光是一些比较简单的输入框之类，还有更多复杂的组件可以使用的，比如DataTable这样的表格组件，Tree这样的树形组件等等。作为一种标准的技术，JSF还得到了相当多工具提供商的支持。同时我们也会有很多很好的免费开发工具可以使用，前不久Sun Java Studio Creator 2和Oracle JDeveloper 10g作为免费的支持JSF的开发工具发布，给JSF带来了不小的生气。另外我们也有一些很

优秀的商业开发工具可共选择，BEA Workshop (原 M7 NitroX)，Exadel，MyEclipse 这样的基于 Eclipse 的插件开发工具，为现在广大的 Eclipse 用户带来了不小的便利，IBM 的 Rational Application Developer 和 Borland 的 JBuilder 也是很不错的支持 JSF 可视化开发的商业开发工具。JSF 和传统的 Web 技术有着本质上的差别，在传统的 Web 技术需要用户自己对浏览器请求进行捕捉，保存客户端状态，并且手工控制着页面的转向，等等。而 JSF 的出现，无疑给我们带来了巨大的便利，JSF 提供了事件驱动的页面导航模型，该模型使应用程序开发人员能够设计应用程序的页面流。与 Struts 的方式向类似的是，所有的页面流信息都定义在 JSF 配置 XML 文件 (faces-config.XML) 中，而非硬编码在应用程序中。这很大程度简化了开发人员开发难度，简化了应用程序的开发。同时 JSF 也是一种遵循模型-视图 - 控制器 (MVC) 模式的框架。实现了视图代码 (View) 与应用逻辑 (Model) 的完全分离，使得使用 JSF 技术的应用程序能够很好的实现页面与代码的分离。所有对 JSF 页面的请求都会通过一个前端控制器 (FacesServlet) 处理，系统自动处理用户的请求，并将结果返回给用户。这和传统的 MVC 框架并没有太大的区别。在 JSF 中不仅使用了 POJO 技术，而且还使用了类似 Spring 的控制反转 (IoC) (或称为依赖注入-DI) 技术，在 JSF 的 Backing Bean 中，我们可以把视图所需要的数据和操作放进一个 Backing Bean 中。同时得益于 JSF 使用的 DI 技术，我们可以在配置文件中初始化 Managed Bean，同时我们也可以通过这样的技术很方便的和使用类似技术的 Spring 进行整合。

## 二、如何在 JSP 中使用 JSF

JSF 只有通过和 JSP 相结合，才能充分发挥它的功效

。JSF是通过标签库和JSP进行集成的。标签库就相当于ASP.NET的服务端组件。JSF提供了非常丰富的标签库，通过这些标签库，可以生成各种客户端模型，如HTML、WML、XML以及JavaScript等。通过这些标签，你可以很容易建立大规模的客户端模型，并由这些标签自动处理客户端请求。接下来让我们来看一个如何使JSF和JSP在一起工作的例子。在JSF中有两个库。第一个叫做内核库，在这个库中包含了各种主要的标签，如配置组件、管理事件、验证输入信息等。第二个库的主要功能是将HTML和JSF的各种标签相对应。每一个JSF标签都会对应一个HTML组件。如UIInput标签对应了HTML中的文本框或密码框。在JSF标签中文本输入框叫做inputText，而密码输入库叫inputSecret。下面是一个简单的JSF和JSP结合的用户接口程序。

```
<%@ taglib
uri="http://Java.sun.com/jsf/html" prefix="h" %> <%@ taglib
uri="http://Java.sun.com/jsf/core" prefix="f" %> <html > <head
> <meta HTTP-EQUIV="Content-Type"
CONTENT="text/html.CHARSET=GB2312" > <title > 第一
个JSF程序 </title > </head > <body > <f:view > <h:form
id="simpleForm" > <h:outputText id="favoriteLabel" value="请输
入一个数字："/> <h:inputText id="favoriteValue"
value="#{simple.longValue}" > <f:validateLongrange
maximum="30" minimum="0"/> </h:inputText > <p/>
<h:commandButton id="submit" value="提交"
action="#{simple.simpleActionMethod}"/> </h:form > </f:view
> </body > </html >
```

在上面的代码中，我们可以了解到JSF是如何同JSP集成的。我们首先可以看到一个内核标签：view

。然后是几个JSF组件。如form、outputText、inputText以及commandButton。这几个组件被放到form中从而开成了form中的一部分。在程序的最开始，必须使用import导入两个标签库。代码如下。 <%@ taglib uri="http://Java.sun.com/jsf/html" prefix="h" %> <%@ taglib uri="http://Java.sun.com/jsf/core" prefix="f" %> 上面2行代码声明了JSP中要使用哪一个JSF标签库。内核库使用前缀（prefix）f声明，而HTML库使用前缀(prefix)h声明。这两个前缀并不是必须要使用，而只是一个建议。在程序中，内核库必须要使用，因为view在所有的JSF页中必须使用。而HTML标签在运行时将JSF标签转化为HTML组件，这个h前缀并不是必须的，而是JSF规范推荐使用的，这样，我们使我们的JSF程序更易读。 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](http://www.100test.com)