

JavaWeb应用程序：JSF技术的详细介绍 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/238/2021\\_2022\\_JavaWeb\\_E5\\_BA\\_94\\_c104\\_238840.htm](https://www.100test.com/kao_ti2020/238/2021_2022_JavaWeb_E5_BA_94_c104_238840.htm) 与微软Visual Studio.NET程序开发有可视化工具的支持相比，人们在开发基于Java的Web用户界面时仍然需要手工书写大量的标签，同时还要考虑页面状态信息的保存、客户端事件处理等问题，开发难度比较大，效率较低，重用性差。Java Server Faces (JSF) 技术正是为了解决这一问题应运而生的，其最引人注目的特性之一是它与标记语言、协议、客户端设备无关。利用JSF提供的可重用、可扩展、基于组件的用户界面框架，在快速开发工具RAD的支持下实现可视化开发。现在JSF技术已经得到了许多大厂商的支持，如SUN公司的JSF Web UI、IBM公司的JSF extension以及Oracle的 ADF Faces等，许多开源项目(典型的Apache Myfaces)也提供对JSF技术的支持。同时Oracle、Sun、Borland和IBM等公司都为JSF提供了开发环境。传统JSP开发存在的问题 实现表示层和业务层的分离，这是J2EE Web应用一直以来的理想，可惜JSP 并没有真正实现这个目标。JSP是一个基于Java的Web用户界面开发标准技术，是一种“脚本式”开发的Web技术，在JSP 页面中混淆了大量用于显示逻辑的HTML和用于业务逻辑的Java 代码，使得页面设计与程序开发无法分离。JSP另一个更大的缺陷是脚本不能重用，这常常导致开发者不得不在JSP页面之间进行复制-粘贴操作，进而导致同一段代码出现多个版本，从而使得程序的调试和设计极其错综复杂。而标签库TagLib作为JSP的补充，将Java代码从JSP中剥离，也只是有限地实现了表现与逻辑的分离，始终没有摆脱

代码和HTML页面揉和的问题。此外，JSP还存在着其他固有的缺陷，例如对应Servlet编译的出错信息很难在JSP中找到准确的出错位置，由此给调试带来很大困难。

JSF技术介绍

在Sun公司提出的J2EE(Java 2 Enterprise Edition)平台上，Java Server Faces (JSF) 是一种用于构建Web应用程序的新标准Java框架。它提供了一种以组件为中心来开发Java Web用户界面的方法，从而简化了开发。“企业开发人员”和Web设计人员将发现JSF开发可以简单到只需将用户界面 (UI) 组件拖放到页面上，而“系统开发人员”将发现丰富而强健的JSF API为他们提供了无与伦比的功能和编程灵活性。JSF还通过将良好构建的模型-视图-控制器 (MVC)设计模式集成到它的体系结构中，确保了应用程序具有更高的可维护性。JSF是由Java Community Process (JCP)制定的一个Web应用框架标准。JSF具有良好定义的请求处理生命周期和丰富的组件层次结构，旨在推动基于Java的Web用户界面开发的简易性。利用JSF提供的可重用、可扩展、基于组件的用户界面框架，在快速开发工具RAD的支持下，可以通过拖放组件的方式对Web用户界面进行可视化编辑，将用户界面上的组件与一个数据源绑定，并将客户端用户界面产生的事件交给服务器端处理，从而大大降低基于Java的Web用户界面的开发难度，提高开发效率。

Java Server Faces技术包括两个主要部分：(1) 一组用于表示用户界面组件并管理其状态、处理事件和验证输入的JavaAPI，这些API支持国际化和可访问性(accessibility). (2) 一个用于在JSP页面中表示JSF 组件的自定义标签库。简而言之，JSF的主要部分是一个GUI组件框架和一个用于跨不同标记语言或客户端设备描述组件的灵活模型。JSF GUI组件框架使开发人

员能够创建JSF应用程序的用户界面。JSF GUI组件包括标准的HTML表单控件(如按钮)、布局组件，以及更复杂的组件，如数据表。此外，第三方还可以扩展规范中定义的基本类，来开发额外的GUI组件。JSF组件的体系结构是这样设计的：组件的功能由组件类定义，组件的呈现由一个单独的呈现器(renderer)定义。呈现器定义了组件类如何映射为适合特定客户的组件标签。JSF参考实现中包含了一个标准的RenderKit，用于生成基于HTML 4.01的标记，实现将组件类呈现给HTML客户。JSF的主要优势之一就是它既是Java Web用户界面标准又是严格遵循模型-视图-控制器(MVC)设计模式的框架。用户界面代码(视图)与应用程序数据和逻辑(模型)的清晰分离使JSF应用程序更易于管理。为了准备提供页面对应用程序数据访问的JSF上下文和防止对页面未授权或不正确的访问，所有与应用程序的用户交互均由一个前端“Faces” servlet(控制器)来处理。JSF技术做到了应用程序逻辑和表示的完全分离，是真正彻底的MVC模式。JSF技术现在的版本是1.1，下一个版本将是1.2，它将和J2EE 5.0一起发布。JSF是构建Web应用程序的新标准Java框架，可以依据JSF的框架标准手工书写Web应用程序，以文本方式像写HTML或JSP程序一样实现Web用户界面的设计。同时在开发、运行以及调试前还需要搭建系统运行平台。这样做的缺点是不能充分体现可视化开发的便捷和高效，一般是在分析Web应用程序的结构时使用。使用快速开发工具RAD，是实现JSF可视化开发的先决条件。在快速开发工具的支持下，符合JSF标准的Web应用程序开发、运行以及调试均在IDE环境中实现，大大提高了开发的效率，整个程序开发过程快捷高效。现在获得大厂

支持的RAD工具很多，比如IBM公司的WebSphere Studio，Oracle公司的JDeveloper，和Sun公司的Java Studio Creator等都提供对JSF开发的支持。如果不使用IDE开发环境，那么在开发应用程序前，必须先搭建系统运行平台。比如安装TOMCAT和Sun的Java Web Services Developer Pack (JWSDP)

### 1.2. 使用JSF时需要JSTL与JSF的标签函数库，在Web应用程序的WEB-INF\lib目录下放入所需的JAR文件。如果使用IDE开发环境，开发前期的准备工作比较简单，只需安装好IDE环境即可，不需要其他的配置，一般IDE中都有内嵌的应用服务器，足以支持程序的开发、运行和调试。不管是否使用IDE开发环境，JSF开发Web应用程序通常分为以下三个步骤：

(1)视图设计方便、快捷地开发基于JSF的Web应用程序，是在JSF提供了一组丰富的、可重用的服务器端用户界面组件的条件下实现的。在开发工具的支持下，用户可以很容易地在可视化环境中利用这些组件构建Web用户界面，处理组件的数据校验、事件处理等用户界面管理问题。设计每个需要的JSP页面，在页面中放置JSF的内置组件，并将组件与应用层的JavaBean绑定。利用RAD工具能通过拖放组件的方式轻松实现页面文件，不需要手写大量代码。开发工具自动生成页面对应的java文件，在文件中定义好了页面的JSF组件及其getter、setter方法以及JSF组件的“action”属性对应的方法等。程序员可以方便地在已有程序的基础上修改、调试。

(2)模型设计 JSF技术做到了应用程序逻辑和表示的完全分离。在模型设计部分，程序员只需要考虑程序的逻辑功能，不需要考虑数据的表现形式。JSF的模型是通过JavaBean程序来实现的。JSF的模型是根据页面中组件需要处理的业务逻辑而设计实现的

。在JSF中使用JavaBean可以直接在JSF页面中声明或者在配置文件faces-config.xml中声明。如果手工部署应用，则编译完成的JavaBean文件应放到Web应用程序的WEB-INF\classes目录下(TOMCAT环境下).如果借助于RAD开发工具，设计人员就不需要考虑部署的细节，开发工具会自动把实现了业务逻辑的JavaBean文件(未编译的java文件和编译成功后的jar包)成功地部署在相关位置。(3)控制器设计 JSF的控制器设计是在配置文件中完成的，相关的主要有两个文件：web.xml文件和faces-config.xml文件(均在WEB-INF目录下)。web.xml文件主要用于控制JSF的生命周期，实现部署描述符

。faces-config.xml文件实现导航，在文件中控制页面之间的跳转流程。配置步骤是首先配置Web应用程序的web.xml文件，然后是配置JSF的控制文件faces-config.xml文件。这两个文件格式固定，在IDE环境中这个过程由工具自动完成，不需要设计人员的干预，即使手工配置也比较简单。JSF的技术重点在View部分，它实现了Web应用程序设计角色的完全分离

。JSF网页设计者只需要专注于页面的设计.应用程序开发者主要关心Model部分的JavaBean的开发.程序的流程控制则

由faces-config.xml专门配置。总结 JSF是基于Java的Web应用开发领域里提供了一个可重用、可扩展、基于组件、工具友好的服务器端UI框架。在支持JSF的RAD开发工具中，人们可以像使用Visual Studio.NET一样方便快捷地构建Web用户界面，大大降低了利用Java技术实现Web用户界面的难度，提高了开发效率。JSF具有强大的组件体系和事件处理系统，完全实现了MVC模式的应用架构，使得基于Java的Web用户界面程序开发难度大大降低，提高了开发效率，非常适用

于Web页面的开发。JSF是一个开放的标准，具有很好的可扩展性。依照JSF的规范，用户完全可以根据需求定制自己的用户界面组件、事件处理器、数据校验和转换组件等，这些组件和标准的JSF组件一样是可重用的。JSF API是直接架构在Servlet API之上的，因此JSF用户界面组件的呈现并不局限于特定的脚本技术或标记语言。表示层完全可以采用JSP之外的技术。在众多的J2EE表现层框架技术中，JSF表现出其旺盛的生命力。尽管JSF技术还有不少问题，但随着JSF技术的不断成熟和版本更新，其必将获得越来越多的应用。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)