

主要内容

Java Web 技术介绍

什么是 JSP

JSP示例程序的编译与运行

JSP基本语法

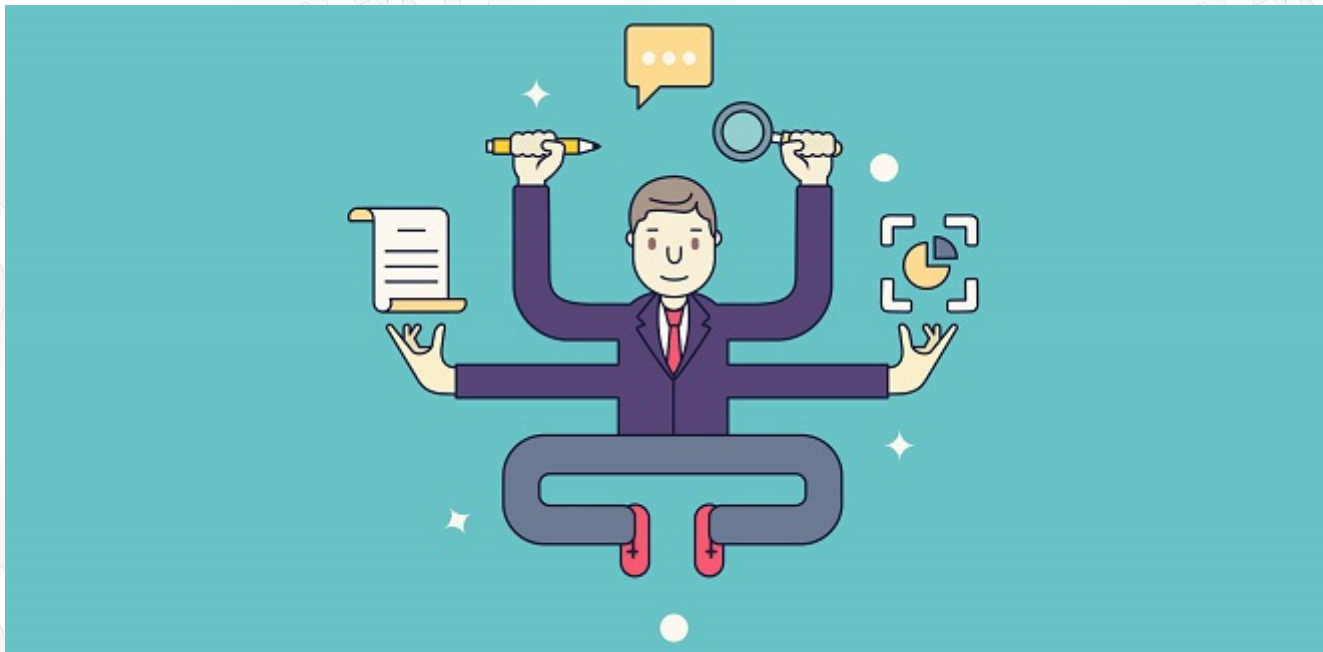
Java Web

Java Web是用Java技术来解决相关web互联网领域的技术总称。

需要在特定的web服务器上运行，分为web服务器和web客户端两部分。

跨平台，能够在多个不同平台下部署与运行。

需要会哪些技术



需要会哪些技术

基于页面的前端技术，如HTML,CSS,JavaScript,JQuery等。

动态语言技术，如Java,JSP等。

数据库的技术，如Oracle,MySQL,SqlServer等。

其他工具与组件，如服务器，SSM，SSH框架等。

Java Web应用场景一：淘宝

2004年，淘宝由PHP语言转换为Java语言。



Java Web应用场景二：12306



1：高并发

2：数据量大

3：软件与硬件成本高

JSP简介

JSP全名为Java Server Pages，中文名叫Java服务器页面，其根本是一个简化的Servlet设计。

JSP是在传统的网页HTML文件中插入Java程序段(Scriptlet)和JSP标记(tag)，从而形成JSP文件，后缀名为(*.jsp)。

用JSP开发的Web应用是跨平台的，既能在Linux下运行，也能在其他操作系统上运行。

JSP与Servlet的不同

JSP在本质上就是SERVLET,但JSP是Servlet的一种简化, JSP由HTML代码和JSP标签构成, 可以方便地编写动态网页。

Servlet完全是由JAVA程序代码构成流程控制和事务处理。

Servlet的应用逻辑是在Java文件中, 并且完全从表示层中的HTML里分离开来。JSP侧重于视图, Servlet主要用于控制逻辑。

开发环境准备

Eclipse IDE for Java EE(32位)

JDK_1.8.0_31(32位)

Apache-tomcat-8.0.35(32位)

第一个JSP程序

完整的JSP程序

JSP程序的执行过程

JSP程序生命周期

创建动态项目

New Dynamic Web Project

Dynamic Web Project

Create a standalone Dynamic Web project or add it to a new or existing Enterprise Application.

Project name: HelloWeb

Project location

☒ Use default location

Location: D:\workspace\HelloWeb Browse...

Target runtime

<None> New Runtime...

Dynamic web module version

3.0

Configuration

Default Configuration Modify...

The default configuration provides a good starting point. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: EAR New Project...

Working sets

☐ Add project to working sets

Working sets: Select...

? < Back Next > Finish Cancel

创建动态项目

New Dynamic Web Project

Web Module

Configure web module settings.

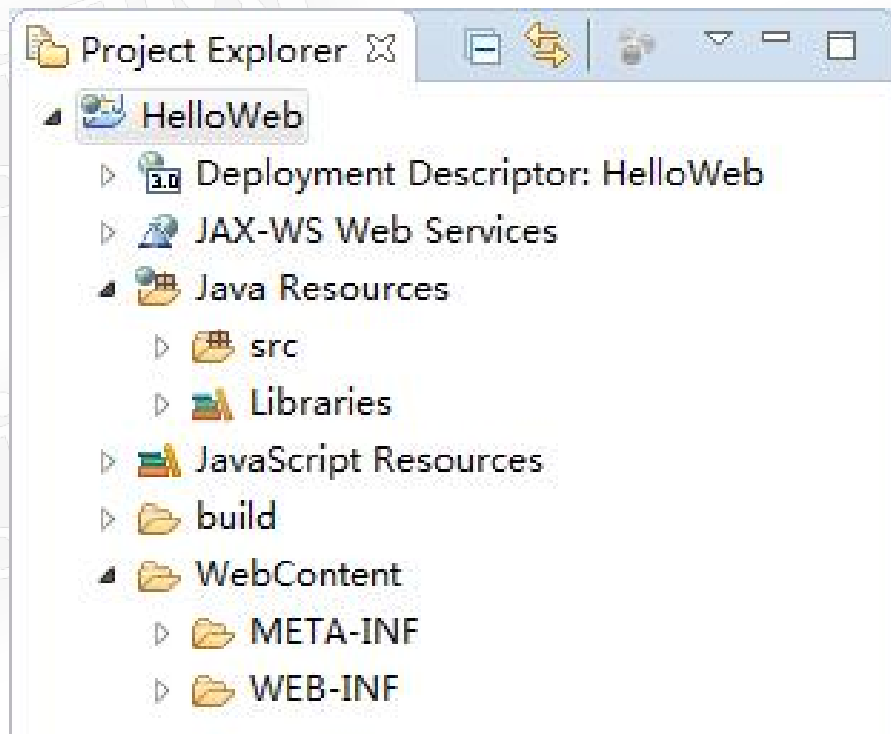
Context root: HelloWeb

Content directory: WebContent

☐ Generate web.xml deployment descriptor

? < Back Next > Finish Cancel

动态项目的结构



动态项目的结构

Java Resources : 包含项目的Java源代码。

WebContent 文件夹 : 所有 Web 资源（包括 HTML、JSP 和图形文件等等）的指定位置。

WEB-INF : 对于 Web 应用程序，此目录包含支持的 Web 资源，包含 web.xml 文件以及 classes 和 lib 目录。

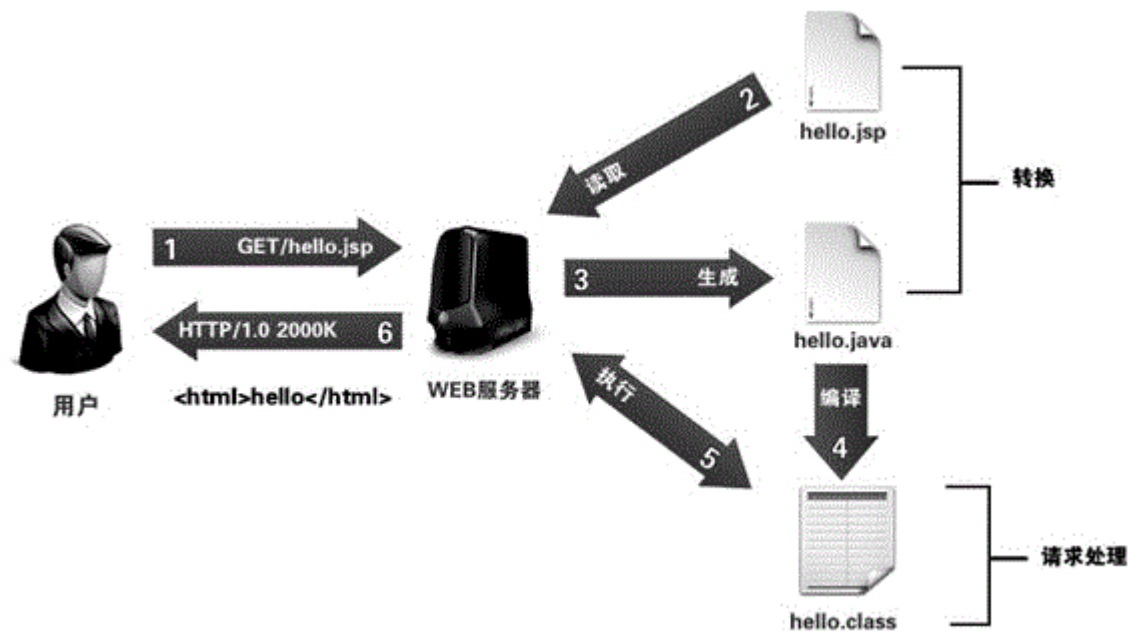
动态项目的结构

/classes : 此目录用于 Servlet、实用程序类以及 Java 编译器输出目录。应用程序类装入器使用此目录中的类来装入类。

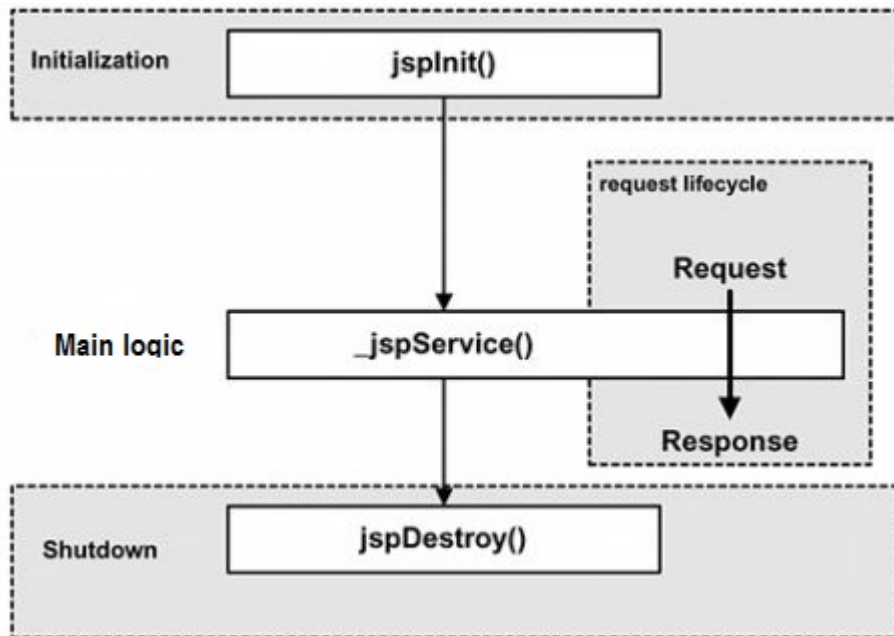
/lib : Web 应用程序引用的支持 JAR 文件。放在此目录中的 .jar 文件中的所有类都可用于 Web 应用程序。

Web 部署描述符 : 标准 Web 应用程序部署描述符 (web.xml 文件)。

JSP程序的执行过程



JSP程序的生命周期



JSP基本语法

JSP声明语法

JSP程序脚本

JSP脚本注释

JSP内容输出表达式

包引入语法

JSP声明语法

语法格式：

```
<%! String str = "hello world" ; %>
```

- 定义成员变量，以及成员方法
- 不能直接包含程序语句

```
<%! out.print( "hello world" ); %>
```



JSP程序脚本

语法格式：

```
<% out.print( "hello world" ); %>
```

```
<% int i = 100; %>
```

```
<%  
    if ( i > 10 ){  
        out.print( "hello world" );  
    }  
%>
```

JSP程序脚本

- 包括变量的声明、表达式和程序逻辑
- 声明的变量转换为_service方法中的变量，因而是局部变量
- 语句块可以自由地与将Java脚本与页面代码组合使用

```
<% int i = 100; %>  
<% if ( i > 10 ){ %>  
    <h1>这里是页面文本: i > 10</h1>  
<% }else{ %>  
    <h1>这里是页面文本: i <= 10</h1>  
<%}%>
```

JSP注释

语法格式：

```
<%-- Java脚本、JSP中其他代码--%>
```

- JAVA语言所固有的注释语法依然使用。

```
<%--  
    <% out.print( "JSP代码被注释掉" ); %>  
--%>
```

JSP注释

```
<%
```

```
// 这是一个单行注释
```

```
/*
```

```
    这是一个多行注释
```

```
*/
```

```
out.print( "hello world" );
```

```
%>
```

JSP内容输出表达式

语法格式：

```
<%! int i = 10; %>  
<%=i %>
```

- 输出的变量名称不需要添加分号

```
<%-- 这是一个带有返回值的方法 --%>  
<%=getInfo() %>
```


JSP包引入语法

语法格式：

```
<%@ page import = "java.util.*" %>
```

```
<%@ page import = "java.io.*" %>
```

- 不同的包引用被逗号隔开，作为一个整体字符串。

```
<%@ page import = "java.util.* , java.io.*" %>
```

总结

本节详细介绍了JSP文件的基本结构，语法，执行过程，是学习其他JSP程序设计技术的基础，对这些基本概念的深刻理解和掌握有利于读者设计出高性能的JSP Web应用程序。