

# JSP基础编程

报告人:包亮

邮箱:rslab@lzb.ac.cn

# HTTP基础

HTTP is a simple, stateless protocol. A client, such as a web browser, makes a request, the web server responds, and the transaction is done.

- **Client**发出一个请求:包括**HTTP**命令和一些附加信息如**C**端用的是**什么软件**和**什么类型的内容**.
- **服务器**响应:版本号+**状态码(200,404)**和**服务器的附加信息**.
- **例子**:见下页

GET /intro.html HTTP/1.0

User-Agent: Mozilla/4.0 (compatible; MSIE 4.0; Windows 95) Accept: image/gif, image/jpeg, text/\*, \*/\*

HTTP/1.0 200 OK

Date: Saturday, 23-May-00 03:25:12 GMT Server: Tomcat Web Server/3.2 MIME-version: 1.0 Content-type: text/html Content-length: 1029 Last-modified: Thursday, 7-May-00 12:15:35 GMT

# 与服务器交互的方法

- 把客户端连接服务器并发送的http请求叫作方法.
- 最常见的是Post和Get方法,POST是张贴而GET是读

# Jsp服务器的搭建

**Web服务器+应用程序服务器+数据库**

**Web服务器+独立的servlet容器+数据库**

大多数应用程序服务器和若干独立servlet容器都内置了web服务。

# Apache+tomcat+mysql

- 安装j2sdk-1.4.06
- 安装apache
- 安装tomcat(4.x-5.0.x)
- apache和tomcat的整合  
<http://linux.chinaunix.net/doc/netconf/2005-02-02/1058.shtml>

# “一切都是对象”

- **Java**是纯的面向对象程序设计语言.**Jsp**继承了**java**的语法和特性.
- **JSP**从这个层次上面可以理解为:用**java**的方法来解决**http**协议需求.



# Jsp中的隐含对象

Jsp的设计开发人员造出了9个内置对象,用于Http协议的需求.

内置:说明本身jsp页面里面就有了.不需要创建对象.如:`Request request = new Request();`//不需要.

# Hello.jsp

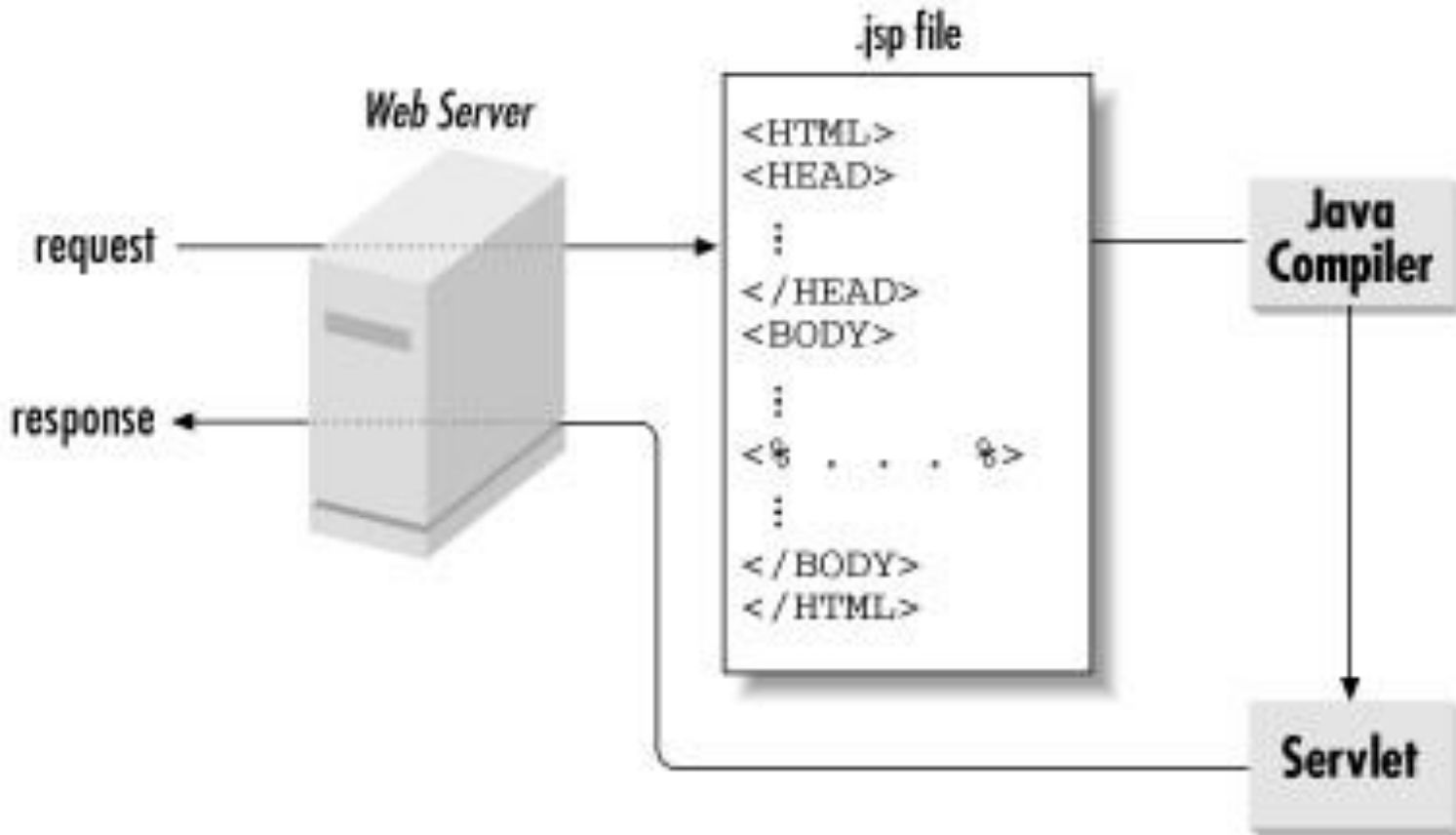
- 主要向大家展示了request内置对象使用方法和页面之间的参数传递.
- `request.getParameter();`

- Request
- Response
- PageContext
- Session
- Application

- Out
- Config
- Page
- Exception

- `HttpServletRequest` request
  - The servlet request.
- `HttpServletResponse` response
  - The servlet response.
- `javax.servlet.jsp.JspWriter` out
  - The output writer, used like a `PrintWriter` but it has different buffering characteristics.
- `HttpSession` session
  - The user's session.
- `ServletContext` application
  - The web application.
- `javax.servlet.jsp.PageContext` pageContext
  - An object primarily used to abstract the implementation of the server but sometimes used directly to share variables between JSP pages and supporting beans and tags.

# 幕后的运行机制



- 经tomcat编译过的hello.jsp文件hello.class

# JSP语法

- Expressions and Declarations

`<%! Declaration;[declaration;]+....%>`

`<%=expression%>`

- Saying "Hello" Using a JSP Declaration –  
hello2.jsp
- 全局变量和局部变量的关系.

# Jsp语法

- Directives

```
<%@ directiveName  
    attributeName="attribValue"%>
```

- `<%@ page ... %>`



# page

- `contentType`
  - Specifies the content type of the generated page. For example:
  - `<%@ page contentType="text/plain" %>` The default content type is `text/html; charset=8859_1`.
- `import`
  - Specifies a list of classes and packages the generated servlet should import. Multiple classes can be given in a comma-separated list. For example:
  - `<%@ page import="java.io.*,java.util.Hashtable" %>` The implicit include list is `java.lang.*,javax.servlet.*,javax.servlet.http.*,javax.servlet.jsp.*`.

- **buffer**
  - Specifies the minimum required size of the response buffer in kilobytes, similar to the servlet method `setBufferSize()`. The value should be written as `##kb`. A special value of `none` indicates that content should be passed directly to the underlying `PrintWriter` in the `ServletResponse` (which may or may not pass the content directly to the client). For example:
    - `<%@ page buffer="12kb" %>` The default is 8kb.
- **autoFlush**
  - Specifies if the buffer should be flushed when it's filled or if instead an `IOException` should be thrown. A `true` indicates to flush, a `false` indicates to throw. For example:
    - `<%@ page autoFlush="true" %>` The default is `true`.
- **session**
  - Indicates the page wants to have access to the user's session. A `true` puts the session variable in scope and may set a client cookie to manage the session. A `false` disables access to the session variable. For example:
    - `<%@ page session="false" %>` The default is `true`.

- **errorPage**
  - Specifies a page to display if a Throwable is thrown from within the JSP page and is not caught before reaching the server. This proves useful because it's difficult to do try/catch blocks when writing JSP pages. For example:
    - `<%@ page errorPage="/error.jsp" %>` The default behavior is implementation dependent. The path is context relative, so you don't need to worry about prepending the current context path. The target may be a JSP but doesn't have to be. If the target is a servlet, the servlet may retrieve the Throwable as the request attribute `javax.servlet.jsp.jspException`.
- **isErrorPage**
  - Indicates the page is intended to be used as the target of an errorPage. If the value is true the page can access an implicit variable named `exception` to retrieve the Throwable.
- **language**
  - Specifies the scripting language used in the code sections of the page. The language used must interact with Java well enough to expose the necessary Java objects to the script environment. For example:
    - `<%@ page language="javascript" %>` The default is java, the only language blessed by the specification.

# JSP和JavaBean

- Javabean是可重用的类,使用<jsp:useBean>标签可以将javaBeans直接嵌入到jsp页面中或作为一个可供jsp调用的操作.

# 调用javabean

- `<jsp:useBean id="name"  
scope="page|request|session|application"  
class="className" type="typeName">`
- `</jsp:useBean>`

# 和bean的交互

- `<jsp:setProperty name="beanName" property="*" />`
- `<jsp:setProperty name="beanName" property="propertyName"/>`
- `<jsp:setProperty name="beanName" property="propertyName" param="paramName" />`

- 例:hello3.jsp调用了java文件,可以减少jsp页面的内容,把逻辑控制放在bean里面,jsp只起到一个显示作用.

# 包含和转向

- `<%@ include file="pathFile"%>`其作用范围:translation time,servlet创建期
- `<jsp:include page  
="pathToDynamicResource" flush="true" />`  
作用范围在请求时间.



- `<jsp:forward  
page="pathToDynamicResource"/>`
- `<jsp:param/>`

# 标签库

- 让jsp页面包含xml标签(类似于html的标签)
- 最广泛的标签库apache Taglibs和Apache Struts

# Jsp乱码和与数据库连接

- Java用的是unicode编码,本地要依赖于环境系统.
- Jdbc数据库联接