



JAVA

JSP技术

Allen Long

Email: allen@huihoo.com

<http://www.huihoo.com>

2004-04

内容安排



- JSP概述
- JSP脚本元素
- JSP编译技术
- JSP内置对象
- MVC模式



What Is JSP



JavaServer Pages (JSP) is a Sun Microsystems specification for combining Java with HTML to provide dynamic content for Web pages. When you create dynamic content, JSPs are more convenient to write than HTTP servlets because they allow you to embed Java code directly into your HTML pages, in contrast with HTTP servlets, in which you embed HTML inside Java code.





将内容的生成和显示进行分离

使用JSP技术，Web页面开发人员可以使用HTML或者XML标识来设计和格式化最终页面生成内容的逻辑被封装在标识和JavaBeans组件中。

强调可重用的组件

绝大多数JSP页面依赖于可重用的，跨平台的组件（JavaBeans或者Enterprise

JavaBeans™组件）来执行应用程序所要求的更为复杂的处理。开发人员能够共享和交换执行普通操作的组件，

采用标识简化页面开发

一个简单的JSP



```
<%@ page contentType="text/html;charset=gb2312" %>
```

```
<html>
```

```
<head> <title>sample</title> </head>
```

```
<body>
```

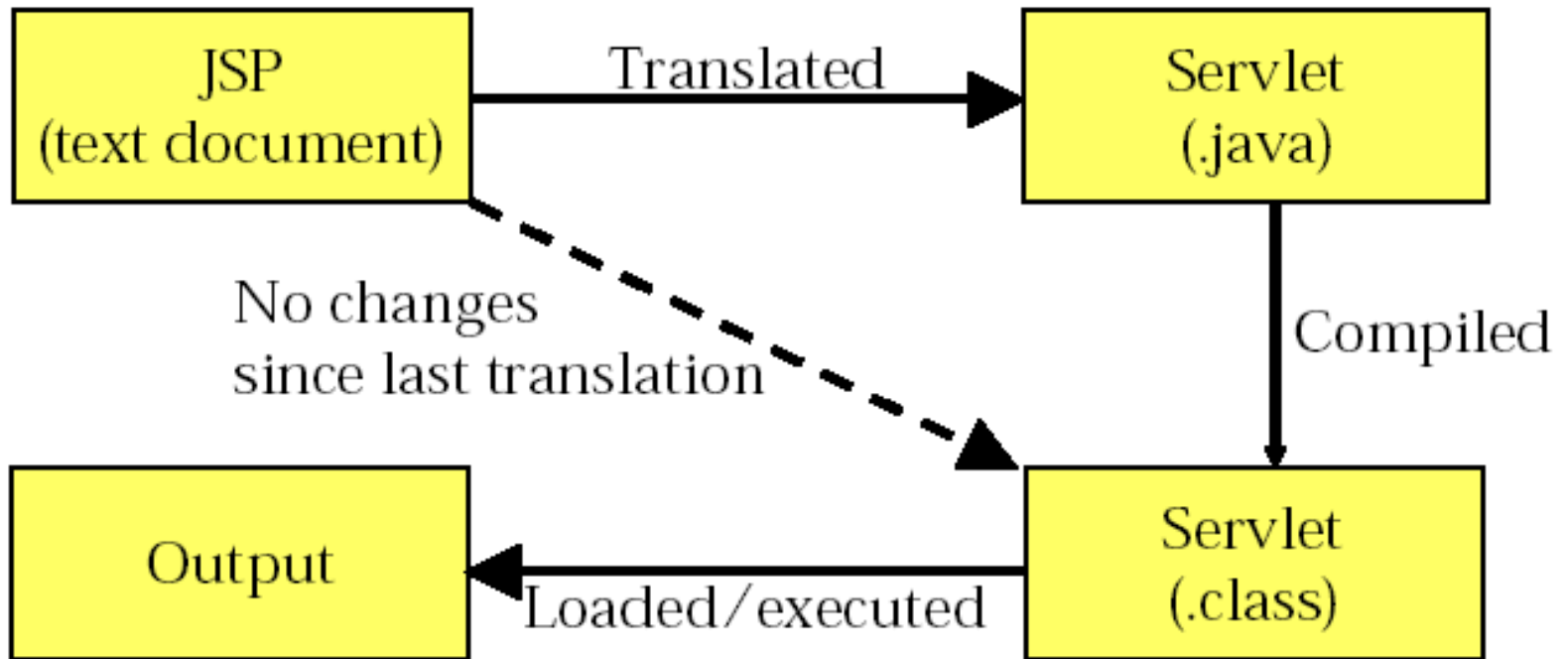
```
<%out.println("hello,huihoo"); %>
```

```
</body>
```

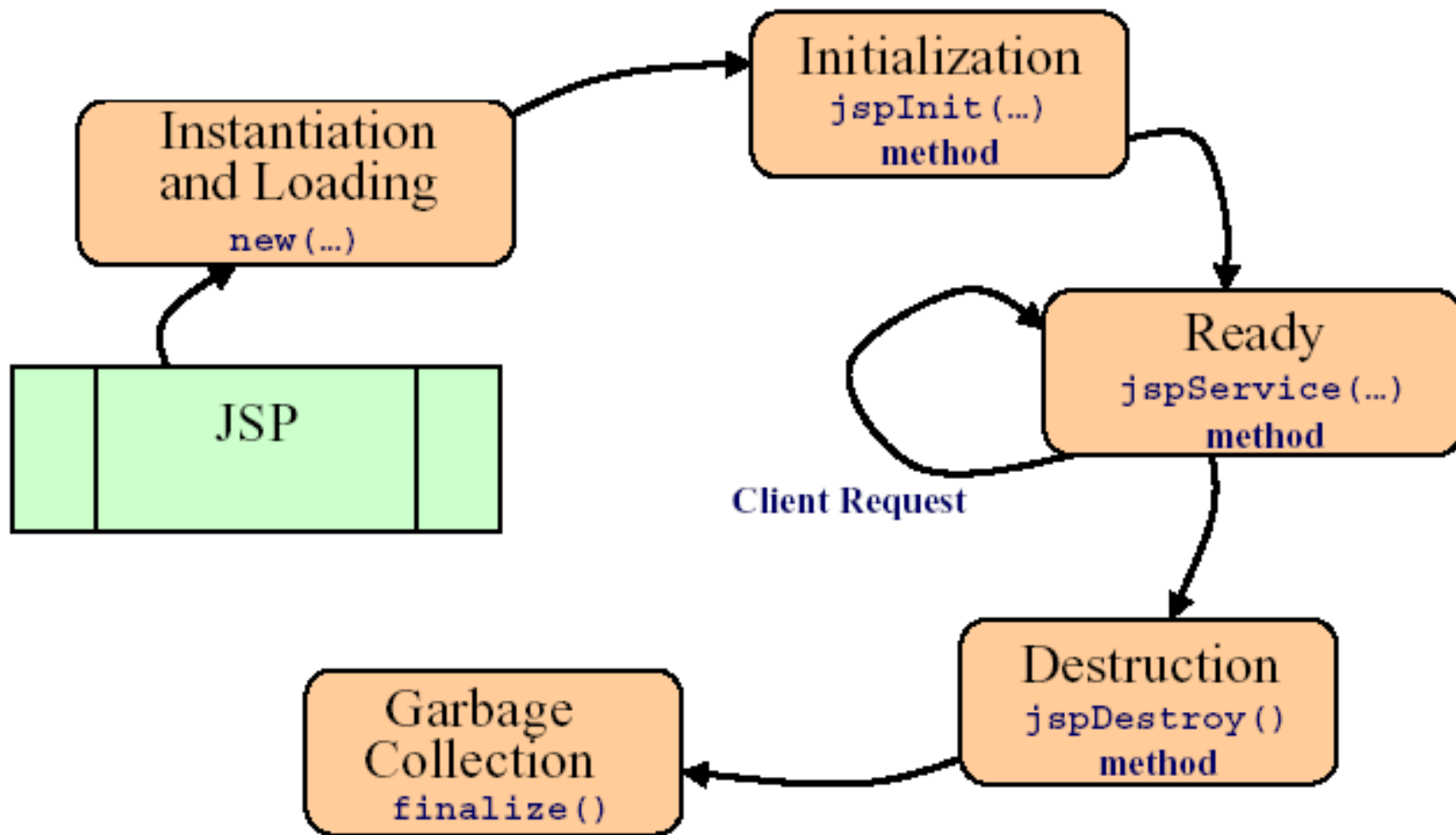
```
</html>
```



JSP处理过程



JSP生命周期



JSP编译



在JSP第一次获得请求时，不管请求来自于客户端浏览器还是服务器上的servlet, JSP文件将被JSP引擎 (JSP engine)转换成为一个servlet。在编译时候如果发现jsp文件有任何语法错误，转换过程将中断，并向客户端发出出错信息；而如果编译成功了，则所转换产生的servlet代码被编译，然后该servlet被JSP引擎加载到内存中.如果.jsp文件被修改了，服务器将自动地对文件重新编译，并将结果取代内存中的servlet，并继续上述处理过程。



JSP脚本元素

JSP Tag Syntax Summary



Scriptlet Syntax:

```
<% java code %>
```

Declaration Syntax:

```
<%! declaration %>
```

Expression Syntax:

```
<%= expression %>
```

Directive Syntax:

```
<%@ dir_type dir_attr %>
```

Action Syntax:

```
<jsp:useBean ...>  
<jsp:include ...>  
<jsp:getProperty ...>  
and more...
```

JSP脚本元素



- Scriptlet 元素
- Expression元素
- Declaration元素



Scriptlet元素



它是一段java代码，被封闭在启动和终止JSP标记中。



Scriptlet Syntax:

```
<% Java Code %>
```

An example of a simple Scriptlet:

```
<% for (int j=0; j < 5; j++) {  
    out.print("Simple Scriptlet Example<BR>");  
}  
%>
```

An example of mixing HTML and Scriptlet:

```
<% for (int j=0; j < 5; j++) { %>  
    Simple Scriptlet Example<BR>  
<% } %>
```

Expression元素



输出java执行结果，具有out.println()功能，开始<%= >
如:<%= new java.util.Date() %>



Expression Syntax:

<%= **expr** %>

Long Hand Notation for Expression:

<% out.print(**expr**); %>

Expression Example:

<%= **i** + 5 %>



Declaration元素



用来进行变量和方法的初始化，并使其他脚本元素使用。它声明的变量和方法是非本地的，即是全局的。<%! >.



Declaration Syntax:

```
<%! Declaration(s) %>
```

Variable Declaration Examples:

```
<%! int i = 0; %>
```

```
<%! String name, phone, address;  
      double balance;
```

```
%>
```



控制语句



- 判断语句
- 循环语句
- 异常语句



动作和指令



- 标准动作
 - javaBean useBean setProperty getProperty
 - Resource
- 指令
 - Page指令
 - Include指令
 - Taglib指令



Action



Action	描述
useBean	Loads and instantiates java class if not already loaded
setProperty	Sets value of variable of JavaBean
getProperty	Gets value of variable of JavaBean
include	Includes another page or servlet in this jsp
forward	Forwards request to another jsp or servlet
plugin	Generates necessary code to load and initialize java plugin

JavaBean



JavaBean封装了页面中使用的表示逻辑或初步业务逻辑。通过业务的封装，提高可维护，易重用。

它主要由3个标准动作实现

<JSP:UseBean>

<JSP:getProperty>

<JSP:setProperty >



<JSP:UseBean>



```
<jsp:useBean id="OurBean" scope="request"  
  class="com.learnweblogic.examples.ch4.UserDataBean"/>
```

<JSP:getProperty>



```
<jsp:getProperty name= name property=  
    propertyName />
```

The attributes are:

- **name** The name of the object instance from which the property is obtained.
- **property** Names the property to get.



<JSP:setProperty >



```
<jsp:setProperty name=" beanName" prop_expr />  
prop_expr ::= property="*" |  
property= propertyName |  
property= propertyName  
param=" parameterName" |  
property= propertyName  
value= propertyValue  
propertyValue ::= string
```

<JSP:include>



在当前的JSP页面中加入静态或动态资源。在JSP容器把包括近来的资源放到输出流中，处理继续进行。

```
<jsp:include page="date.jsp" flush="true" />
```



<JSP:forward>



把当前的请求转化到另一个资源，其中的Page属性指定应转发的资源。

```
<jsp:forward page="date.jsp" />
```



<JSP:plugin>



指示客户端下载Java插件，并在客户端执行。

指令



通过指令，让JSP容器采取必要动作。

指令	描述
Page	for controlling page attributes like class imports
Include	for including other JSPs or HTML documents in this jsp
taglib	for loading tag libraries that extend functionality of jsp

语法：<%@ dir_type dir_attr %>



Page指令



Page指令<%page开始，包括指令类型和值。

<%@ page page_directive_attr_list %>

- { language=" scriptingLanguage" }
- { extends=" className" }
- { import=" importList" }
- { session="true|false" }
- { buffer="none| sizekb" }
- { autoFlush="true| false" }
- { isThreadSafe="true|false" }
- { info=" info_text" }
- { errorPage=" error_url" }
- { isErrorPage="true|false" }
- { contentType="ctinfo" }



Page指令



属性	描述
Language	Defines scripting language used within the tags
Extends	Defines superclass that JSP page inherits from
Import	Defines packages visible to the scripting environment, can appear more than once
Session	Declares whether page participates in http session
Buffer	Specifies the size of the output buffer
autoflush	Declares whether buffer will be flushed or an exception will be raised when the buffer is full
isThreadSafe	Defines how the jsp container should dispatch multiple outstanding client requests
Info	A string that gets incorporated into the translated page
errorPage	Specifies URL that will process all uncaught exceptions
isErrorPage	Declares whether current jsp is another jsp's error page
contentType	Defines the character encoding for the jsp page

<%@include file=""%>



用于加进任何文本和JSP页面。如页首,页尾



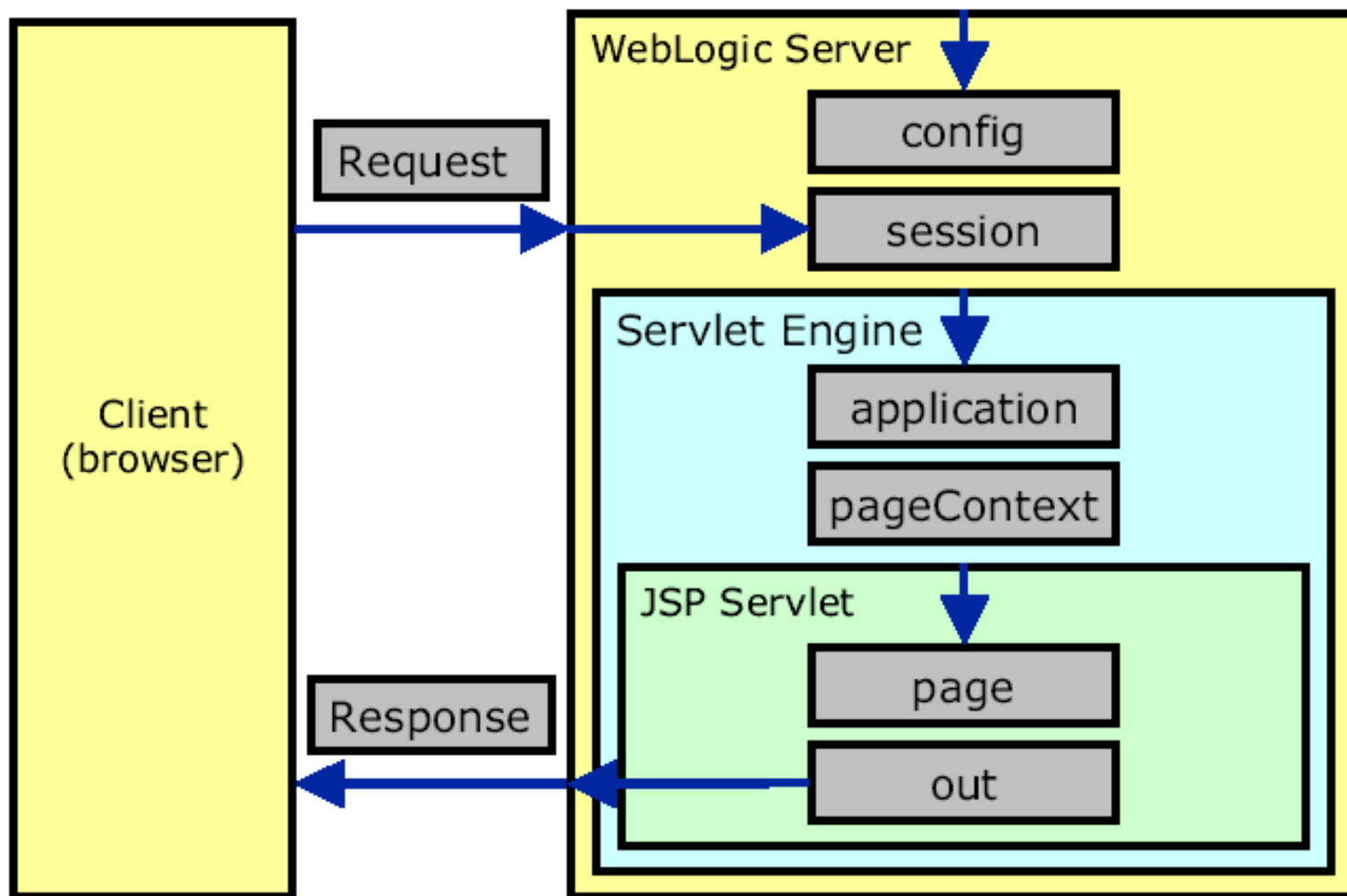
JSP内置对象



- Out
- Response
- pageContext
- Session
- Application
- Config
- Page
- exception



JSP内置对象



Built-in Objects Summary



JSP provides 以下8个 built-in objects:



Object	Type	Scope
request	<code>javax.servlet.HttpServletRequest</code>	request
response	<code>javax.servlet.HttpServletResponse</code>	page
out	<code>javax.servlet.jsp.JspWriter</code>	page
session	<code>javax.servlet.http.HttpSession</code>	session
config	<code>javax.servlet.ServletConfig</code>	page
page	The generated servlet class	page
pageContext	<code>javax.servlet.jsp.PageContext</code>	page
application	<code>javax.servlet.ServletContext</code>	application

获得数据



- 请求
- 获得参数信息
- 获得头信息
- 获得属性信息
- 获得服务器信息



发送信息



- 使用response对象
- Http状态码
- Http头信息
- 转发请求



JSP错误处理



- `ErrorPage`
- `isErrorPage`

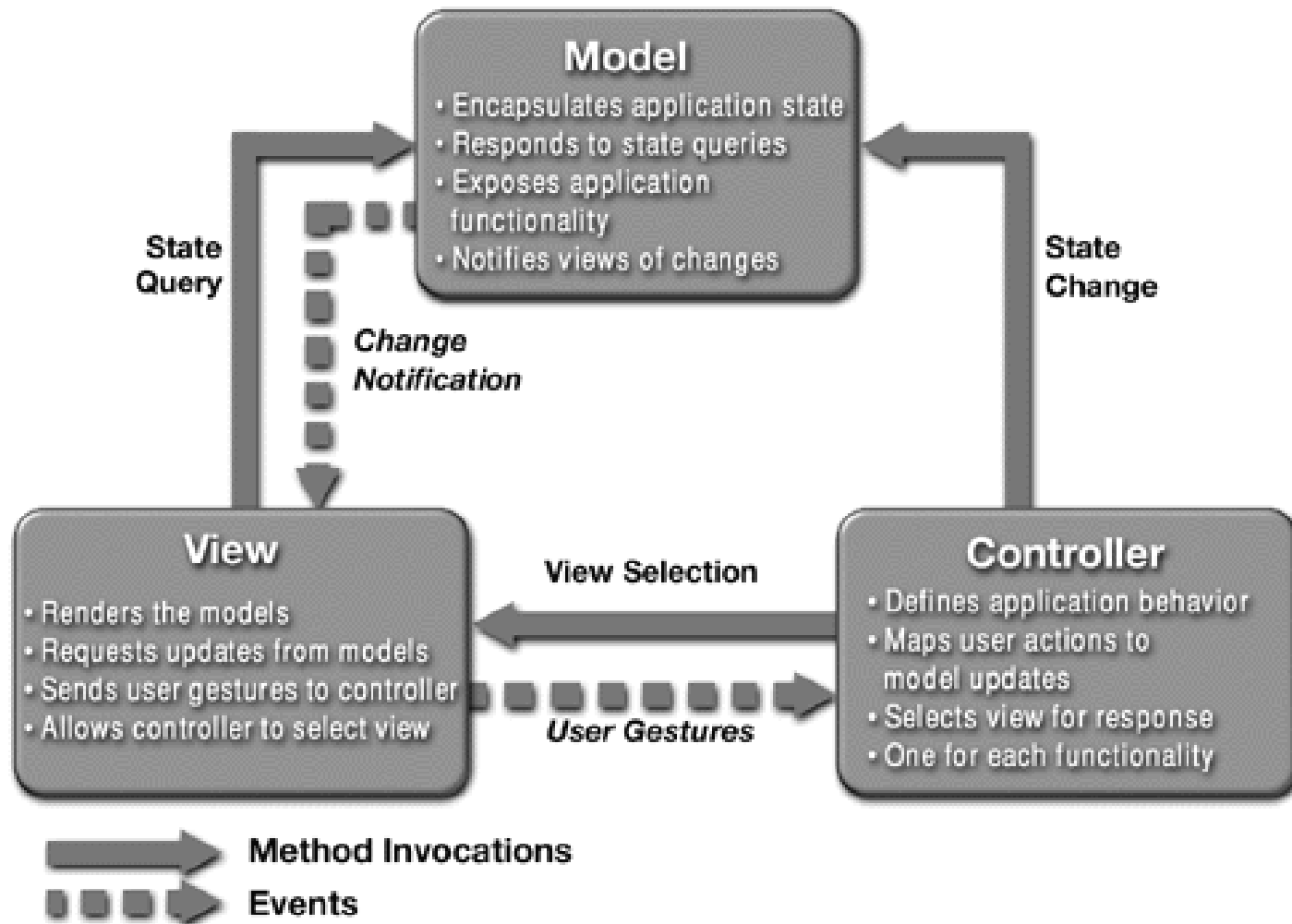




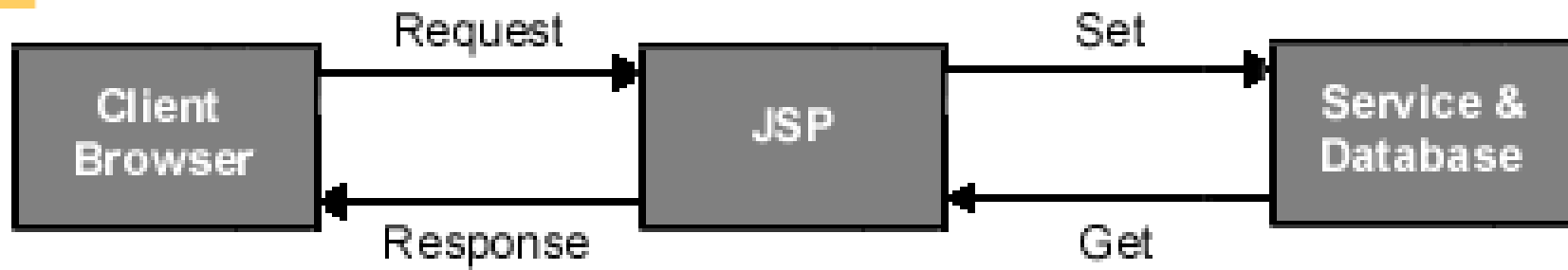
- Cookie
- HttpSession (常用)



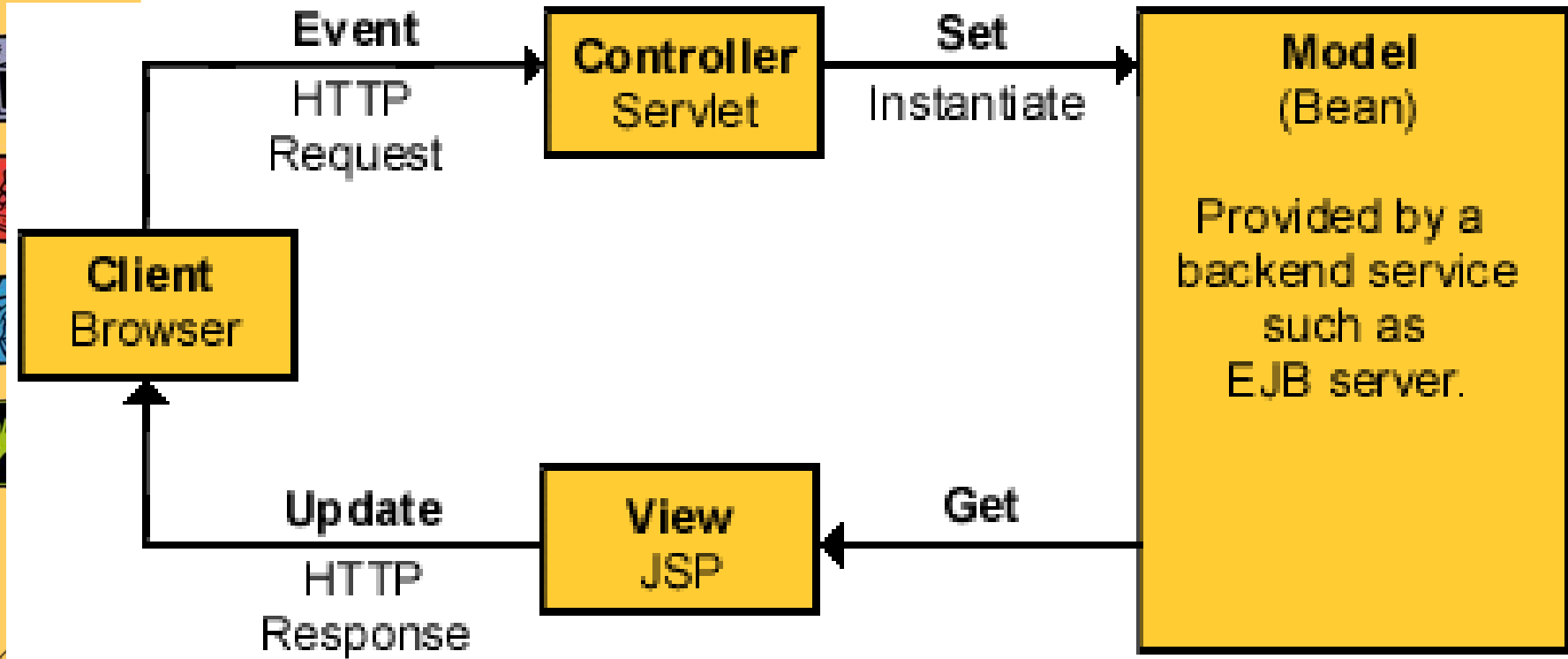
MVC模式



Model1



Model2



JSP与XML



- Jsp生成XML文件
- XML与XSL结合
- 数据与表示的分离



内容回顾



- JSP概述
- JSP脚本元素
- JSP编译技术
- JSP内置对象
- MVC模式



总结



- jsp 是Web应用开发的主流技术
- 其他的脚本语言还有：perl,php,python等, 但jsp具有JAVA语言特性，将会在Web应用开发方面占有一定优势。



参考资料



- <http://java.sun.com/products/jsp/>
sun公司的jsp站点
- <http://www.huihoo.com>
国内一个关于中间件的专业站点



结束



谢谢大家！

allen@huihoo.com

<http://www.huihoo.com>

