**JSP**

### 一、初识jsp

1. 动态网页：随着时间、地点和用户操作的改变而改变。

动态网页需要使用到服务端脚本语言（JSP），静态网页不需要。

1. 架构：
2. CS:Client Server

缺陷：如果软件升级，则所有软件都需要升级。

维护不方便，需要维护每一个客户端软件。

每一台客户端，都需要安装客户端软件。

1. BS:Broswer Server
2. tomcat服务器

bin：可执行文件（startup.bat/shutdowm.bat）

conf：配置文件（server.xml）

lib：tomcat依赖的jar文件

logs：日志文件（记录出错等信息）

temp：临时文件

webapps：可执行的项目（将我们开发的项目放入该目录）

work：存放由jsp翻译成的java以及编译成的class文件。（jsp→java→class）

1. 配置tomcat
2. 配置jdk（必须配置JAVA\_HOME）
3. 配置CATALINA\_HOME：**D:\apache-tomcat-8.5.38**
4. tomcat端口号默认为8080，较容易冲突。（改为8888）
5. 常见状态码
6. 404：资源不存在
7. 200：一切正常
8. 403：权限不足
9. 300/301：页面重定向（跳转）
10. 500：服务器内部错误（常为代码错误）
11. 新建项目
12. 新建项目中必须有WEB-INF/web.xml、WEB-INF/classes、WEB-INF/lib

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee  http://xmlns.jcp.org/xml/ns/javaee/web-app\_3\_1.xsd"  version="3.1"  metadata-complete="true">  <display-name>Welcome to Tomcat</display-name>  <description>  Welcome to Tomcat  </description>  </web-app> |

1. jsp：在html中嵌套java代码（<%%>）

3）在项目/WEB-INF/web.xml设置默认的初始页面

|  |
| --- |
| **<welcome-file-list>**  **<welcome-file>index.html</welcome-file>**  **<welcome-file>index.xhtml</welcome-file>**  **<welcome-file>index.htm</welcome-file>**  **<welcome-file>index.jsp</welcome-file>**  **</welcome-file-list>** |

1. 配置虚拟路径

* 方式一：需要重启tomcat

server.xml中<Host>标签中

|  |
| --- |
| <Context docBase="E:\我的办公\java\workspace\ajax\WebContent" path="/JspProject" /> |

docBase：实际路径；path：虚拟路径

* 方拾二：

conf\Catalina\localhost文件夹中新建 项目名.xml,写入如方法一。

1. 虚拟主机

实现通过[www.baidu.com](http://www.baidu.com)访问自己项目

1. conf/server.xml中在<Engine>中配置Host

|  |
| --- |
| <Host appBase="E:\我的办公\java\workspace\JspProject" name="www.baidu.com">  <Context docBase="E:\我的办公\java\workspace\JspProject" path="/"></Context>  </Host> |

1. 更改默认引擎

<Engine name="Catalina" defaultHost="www.baidu.com">

1. 更改本机域名解析器C:\Windows\System32\drivers\etc\hosts

添加：**127.0.0.1** www.baidu.com

1. 更改端口号为网站默认端口号80，否则需要在网址后添加端口号。

注：网站加载流程：[www.baidu.com→hosts](http://www.baidu.com→hosts)下找映射→server.xml下找Engine的defaultHost→通过虚拟路径映射到自己项目

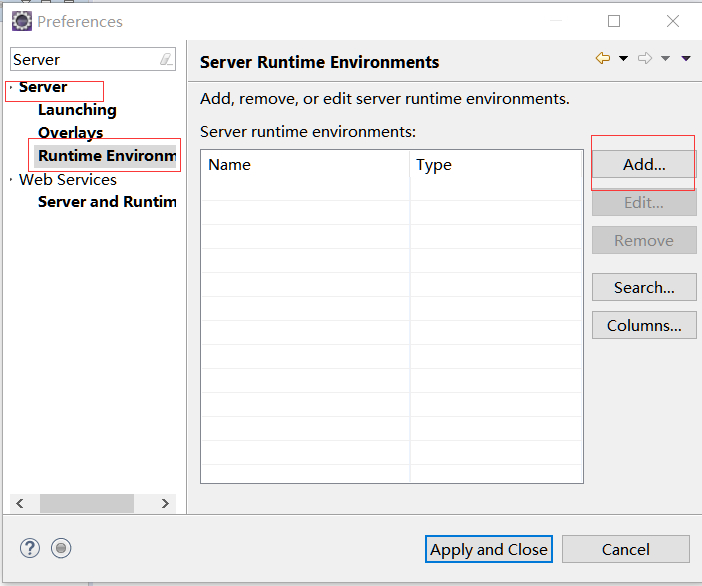
1. JSP执行流程

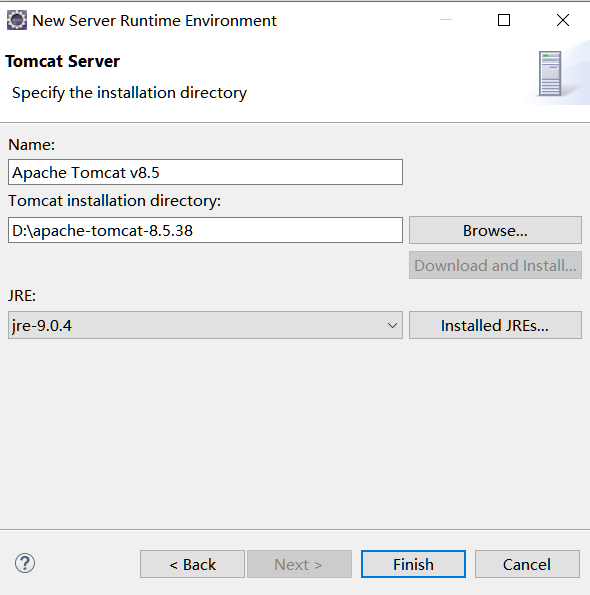
客户端第一次访问服务端：服务端将jsp翻译成java，再将java编译成class文件

第二次访问：直接访问class文件，但如果服务端代码出现了变化，会重新翻译和编译。

1. 使用Eclipse开发web项目
2. 集成tomcat：window→preferences→Server→Runtime Environment→Add(Apache Tomcat v8.5)→选择jre版本
3. 配置下面面板的Servers
4. 其他：在Eclipse中创建的Web项目，浏览器可以直接访问WebContent中的文件。但WEB-INF中的文件无法通过客户端（浏览器）直接访问，只能通过请求转发来访问。

注：并不是任何的内部跳转都能访问WEB-INF，原因是跳转有两种方式：请求转发和重定向。





1. 有的项目可能会报错，需要配置tomcat运行时环境。

方法一：

1. 找到D:\apache-tomcat-8.5.38\lib\servlet-api.jar复制到Java Resource是src中
2. 将其加入到项目的构建路径中：右键Bulid Path→Add to Build Path

方法二：

1. 项目名→Build Path→configure Build Path→Libraries→Add Library→Server Runtime
2. 统一字符集编码
3. 设置jsp文件的编码：jsp→java的转码方式（jsp文件中的pageEncoding属性）
4. 设置浏览器读取jsp文件的编码（jsp文件中的contentType属性）
5. 文本编码：

* 将整个eclipse中的文件统一设置：window→preferences→JSP Files：Encoding
* 设置某一个项目文件设置为统一编码：右键项目→preferences→Resource
* 设置单独文件：文件右键→preferences→Resource

1. 部署tomcat

在Servers面板新建一个tomcat实例，再在实例中部署项目。（右键Add）

1. 其他：

* 一般修改web.xml、配置文件、java需要重启tomcat服务；修改jsp/html/css/js不需要重启tomcat服务。
* out.print()和out.println()都不能回车，要想回车需要写html代码“</br>”

### 二、JSP

1. jsp页面元素
2. 脚本Scriptlet。

* <% 局部变量、java语句%>
* <%! 全局变量、定义方法%>
* <%= 输出表达式%>

1. html代码
2. 指令

* page指令：一般写在网页的开头

属性：

1. language：指定jsp页面使用的脚本语言（一般为java）
2. import：导入的类或者包
3. pageEncoding：jsp文件自身编码（jsp→java）
4. contentType：浏览器解析jsp所用的编码

|  |
| --- |
| <%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"* pageEncoding=*"UTF-8"* import=*"java.util.Date"* %> |

1. 注释

* html注释：<!-- --> ,可以被客户通过浏览器查看源码所观察到。
* java注释：// /\*\*/
* jsp注释：<%-- --%>

1. jsp 9大内置对象（不需要new，也能使用的对象）

* out：输出对象，向客户端输出内容
* request：请求对象，储存客户端向服务端发送的请求信息。

常见方法：

* String getParameter(String name)：根据请求的字段名key，返回字段值value
* String[] getParameterValues(String name)：根据请求的字段名key，返回多个字段值value（checkbox）
* void setCharacterEncoding(“utf-8”)：设置请求的编码（默认编码：tomcat7及以前为iso-8859-1，tomcat8之后utf-8）。
* getRequestDispatcher(“b.jsp”).forward(request,response)：请求转发（a→b）
* ServletContext getServerContext()：获取当前项目的ServletContext对象。
* response：
* pageContext
* session
* application
* config
* page
* exception