

JSP学习笔记

认识JSP

文件:

```
C:\Users\wulele\AppData\Local\JetBrains\IntelliJ IDEA 2020.3\tomcat\*\work\Catalina\localhost\servlet_jsp_war_exploded\org\apache\jsp\*.java
```

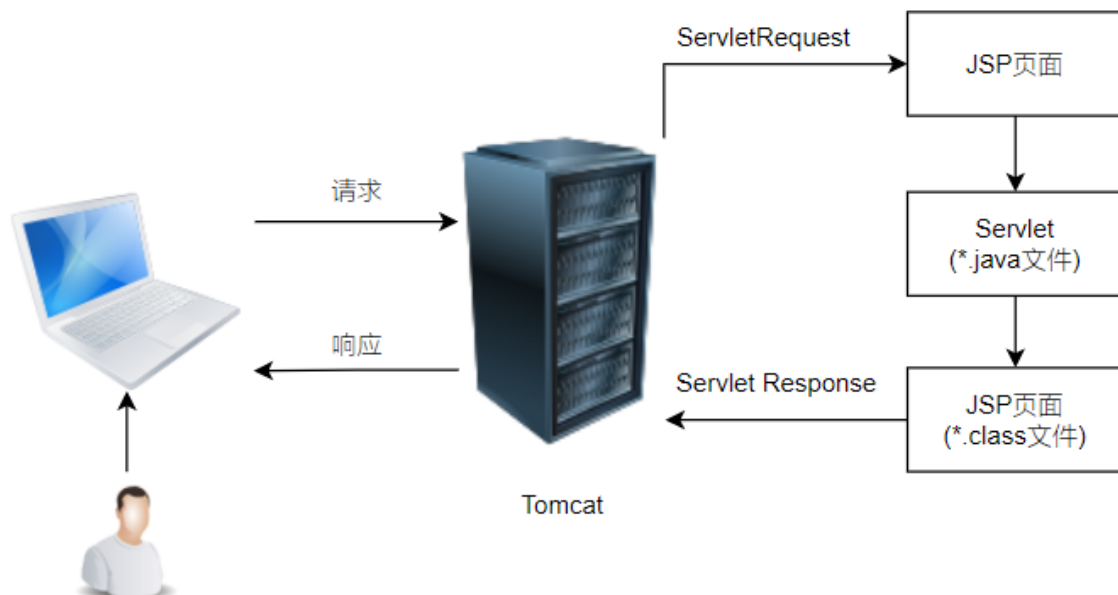
JSP本质上是servlet

Class HttpJspBase

```
java.lang.Object
├── javax.servlet.GenericServlet
│   └── javax.servlet.http.HttpServlet
│       └── org.apache.jasper.runtime.HttpJspBase
```

```
public abstract class HttpJspBase extends HttpServlet implements HttpJspPage {
    private static final long serialVersionUID = 1L;
```

JSP工作原理



JSP基础语法

```
<dependencies>
  <!-- servlet 依赖 -->
  <dependency>
```

```

        <groupId>javax.servlet</groupId>
        <artifactId>javax.servlet-api</artifactId>
        <version>4.0.1</version>
    </dependency>
    <!-- jsp 依赖 -->
    <dependency>
        <groupId>javax.servlet.jsp</groupId>
        <artifactId>javax.servlet.jsp-api</artifactId>
        <version>2.3.3</version>
    </dependency>
    <!-- JSTL 表达式的依赖 -->
    <dependency>
        <groupId>javax.servlet.jsp.jstl</groupId>
        <artifactId>jstl-api</artifactId>
        <version>1.2</version>
    </dependency>
    <!-- standard 标签库 -->
    <dependency>
        <groupId>taglibs</groupId>
        <artifactId>standard</artifactId>
        <version>1.1.2</version>
    </dependency>
</dependencies>

```

JSP表达式

<%-- jsp表达式，作用：用来将程序输出到客户端

格式：<%= 变量或表达式%> --%>

<%= new java.util.Date()%>

JSP脚本片段

```

<%
    int sum = 0;
    for (int i = 0; i < 100; i++) {
        sum += i;
    }
    out.print("<br/>Sum=" + sum);
%>

```

JSP内嵌HTML

```

<%
    for (int i = 0; i < 5; i++) {
%>
    <h2>Hello world <%=i%></h2>
<%
    }
%>

```

JSP声明

```

<%!
    static {
        System.out.println("This is a static block");
    }

    private int global = 0;

    public class Test {
        Test() {
        }
    }
%>

```

声明会被生成到*.java文件的类中，其他的（jsp表达式等）会被生成在_jspService()方法中，详见文件

JSP指令

```

<%@ page contentType="text/html; charset=UTF-8" language="java" %>
//设置500页面
<%@ page errorPage="error-page/500.jsp" %>
//取公共页面
<%@ include file="" %>
<%--合成页面，本质上是一个页面--%>
<%@include file="commom/header.jsp"%>
<h1>Hello</h1>
<%@include file="commom/footer.jsp"%>
<hr/>
<%--jsp标签--%>
<%--拼接页面，本质上是三个页面--%>
<jsp:include page="commom/header.jsp"/>
<h1>world</h1>
<jsp:include page="commom/footer.jsp"/>

```

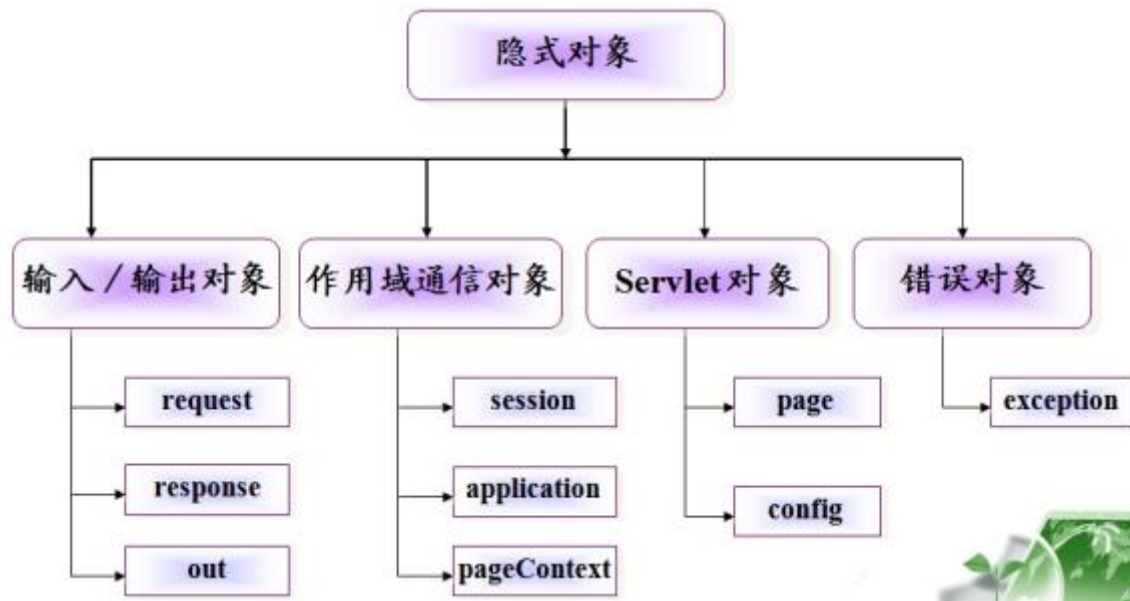
web.xml配置错误页面

```

<error-page>
    <error-code>404</error-code>
    <location>/error-page/404.jsp</location>
</error-page>
<error-page>
    <error-code>500</error-code>
    <location>/error-page/500.jsp</location>
</error-page>

```

JSP内置对象



```

<%
    pageContext.setAttribute("first","wll-01"); // 保存的数据只在一个
    页面有效
    request.setAttribute("second","wll-02"); // 保存的数据在一次请
    求中有效，转发会携带这个数据
    session.setAttribute("third","wll-03"); // 保存的数据在一次会
    话中有效，从打开浏览器到关闭浏览器
    application.setAttribute("fourth","wll-04"); // 保存的数据在服务器
    中有效，从打开服务器到关闭服务器
%>
<%
    String first = (String) pageContext.getAttribute("first");
    String second = (String) request.getAttribute("second");
    String third = (String) session.getAttribute("third");
    String fourth = (String) application.getAttribute("fourth");
%>
${first}    <br/>
${second}  <br/>
${third}   <br/>
${fourth}  <br/>

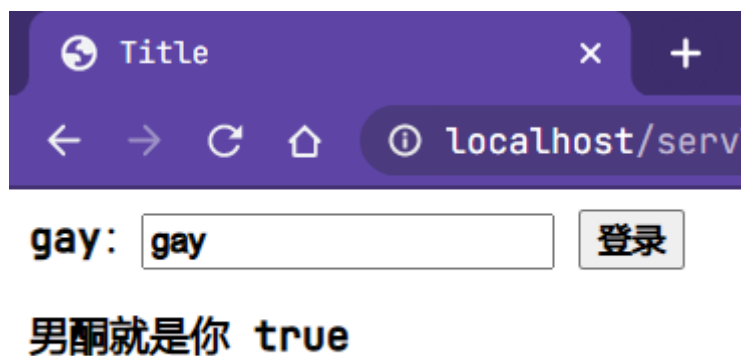
```

JSTL、EL表达式、JSP标签

```
<%-- 引用JSTL核心标签库，有时还需将jstl和standard的jar包复制一份到tomcat的lib目录下--%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
```

```
<body>
    <form action="jstl.jsp" method="get">
        <span>gay: </span><input type="text" name="gay"
value="${param.gay}">
        <input type="submit" value="登录">
    </form>
</body>
<c:if test="${param.gay=='gay'}" var="gay">
    <c:out value="男酮就是你" />
</c:if>

<c:out value="${gay}" />
```



JavaBean

特征

1. 必须为共有类
2. 必须有无参构造
3. 属性以private修饰
4. 必须有对应的set()和get()方法
5. 一定放在包内

作用

一般用来和数据库的字段做映射

ORM：对象关系映射

- 类：表
- 属性：字段

- 对象：行记录

插件lombok，省去写构造方法、setter()方法等

```
package com.wll.pojo;

import lombok.*;

/**
 * @author wulele
 */
@Setter
@Getter
@NoArgsConstructor
@ToString
@AllArgsConstructor
public class People {
    private int id;
    private String name;
    private int age;
    private String address;
}
```

```
<%
// 等同于
//     People people = new People();
//     people.setId();
//     people.getId()
%>
<%-- 获取对象 --%>
<jsp:useBean id="people" class="com.wll.pojo.People"
scope="page"/>
<%-- 设值 --%>
<jsp:setProperty name="people" property="id" value="1"/>
<jsp:setProperty name="people" property="name" value="jack"/>
<jsp:setProperty name="people" property="age" value="20"/>
<jsp:setProperty name="people" property="address" value="UK"/>
<%-- 取值 --%>
id:
<jsp:getProperty name="people" property="id"/>
<br>
姓名:
<jsp:getProperty name="people" property="name"/>
<br>
年龄:
<jsp:getProperty name="people" property="age"/>
<br>
地址:
<jsp:getProperty name="people" property="address"/>
```