

即jsp最终需要被转化为.class文件才能被浏览器访问，同时这个jsp或者jsp的这个.class文件本质上其实就是个servlet，原因是只有servlet才能被外界访问到（即一个java类要想被外界访问到，必须是一个servlet）

JSP：入门学习

1. 概念：

- * **Java Server Pages:** java服务器端页面

- * 可以理解为：一个特殊的页面，其中既可以指定定义html标签，又可以定义java代码

- * 用于简化书写！！

2. 原理

- * **JSP本质上就是一个Servlet**

3. JSP的脚本：JSP定义Java代码的方式

1. **<% 代码 %>**：定义的java代码，在jsp这个servlet的service方法中。service方法中可以定义什么，该脚本中就可以定义什么。（注：<% 代码 %>是可以与下一个<% 代码 %>进行代码连续的，原因是有时候为了省略写response.getWriter().write，用html来写，在<% 代码 %>是可以与下一个<% 代码 %>之间进行插入）

2. **<%! 代码 %>**：这种形式用的非常少，定义的java代码，定义的是jsp这个servlet这个类的成员变量或者成员方法或者静态将代码快

3. `<%= 代码 %>`: 定义的java代码, 它相当于一个输出语句, 等效于 `System.out.println()`; 会输出到页面上, 输出语句中可以定义什么, 该脚本中就可以定义什么, 同时也需要遵守输出语句里的语句格式

4. JSP的内置对象:

- * 在jsp页面中不需要获取和创建 (即不需要import导入或者继承), 可以直接使用的对象

- * jsp一共有9个内置对象。

- * 今天学习3个:

- * request

- * response

- * out: 字符输出流对象。可以将数据输出到页面上。和 `response.getWriter()` 类似

- * `response.getWriter()` 和 `out.write()` 的区别:

- * 在tomcat服务器真正给客户端做出响应之前, 会先找response缓冲区数据, 再找out缓冲区数据。

- * `response.getWriter()` 数据输出永远在 `out.write()` 之前

5. 案例:改造Cookie案例

创建 `home.jsp`, 把 `cookieTest` 里的文件都给复制进来 `<body><%%></body>` 里, 并且进行导包, 然后进行以下修改:

差别:

- * 进行微小修改, 不用再设置响应的消息体格式与编码, 因为在jsp里已经默认设置好了

原来的:

```
response.setContentType("text/html;charset=utf-8");
```

现在的:

```
<%@ page contentType="text/html;charset=UTF-8"
language="java" %>
```

*

原来的:

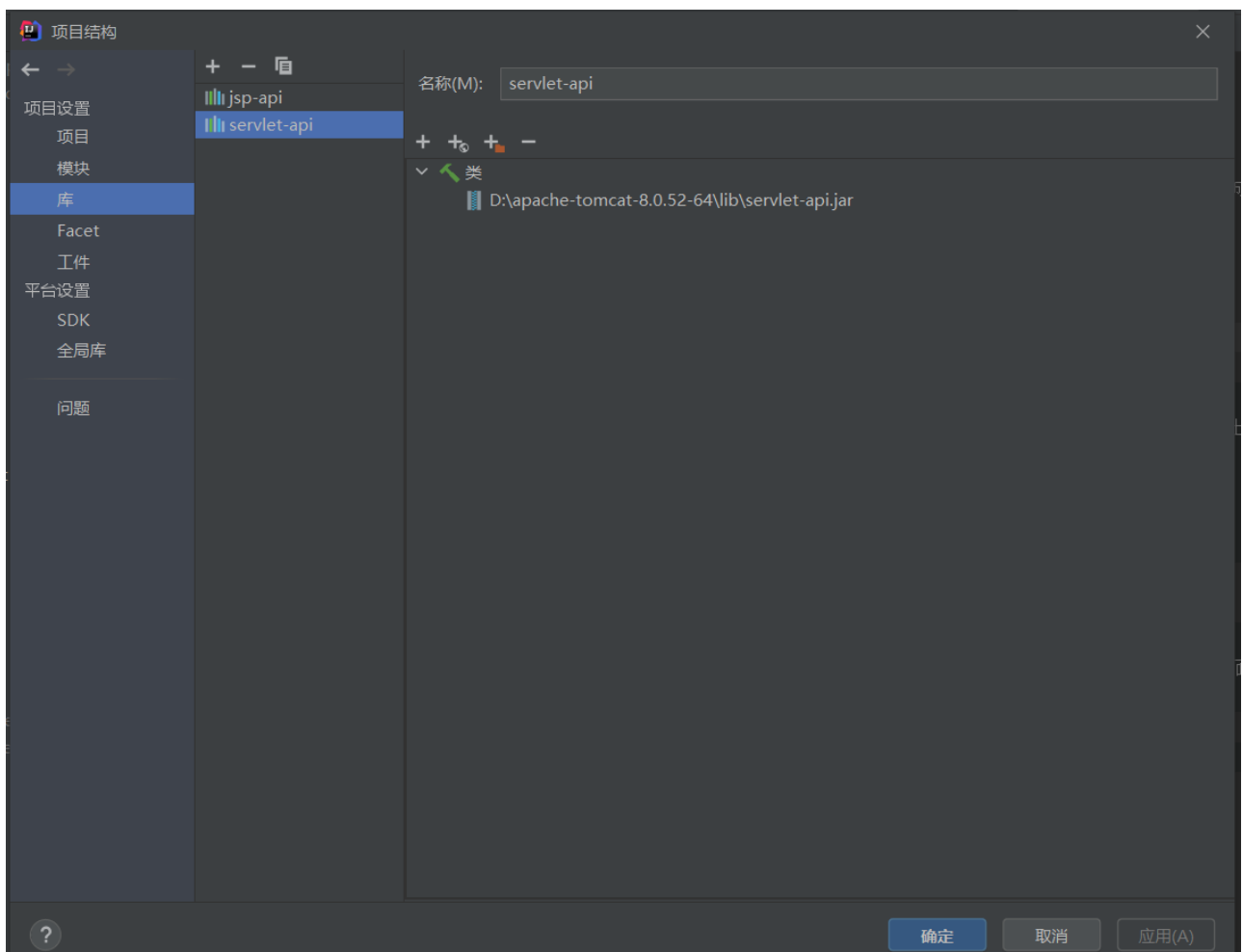
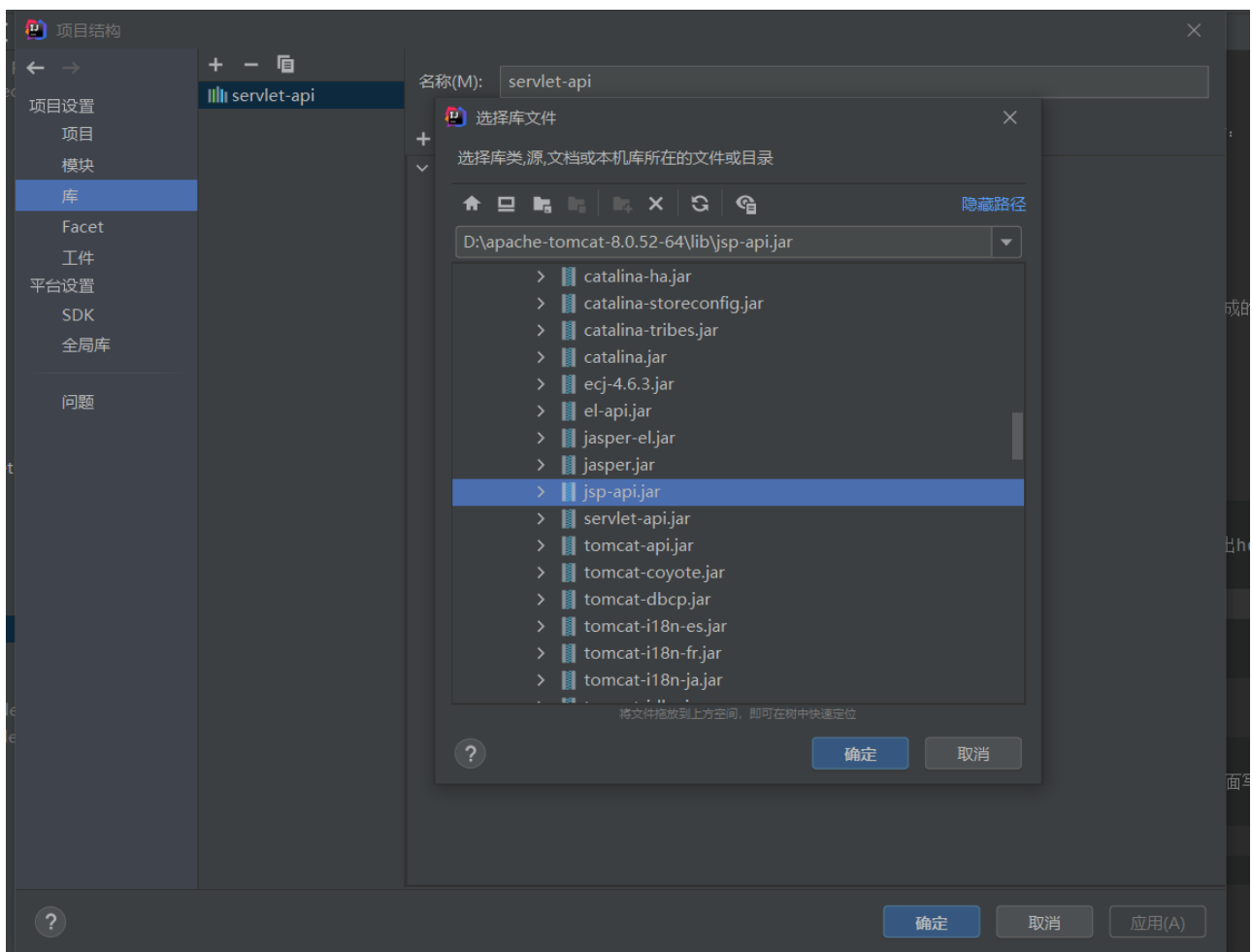
```
response.getWriter().write("<h1>欢迎回来, 您上次访问时间为:"+value+"</h1>");
```

现在的 (注: 这是放在 `<body>` 标签里的):

```
<h1>欢迎回来, 您上次访问时间为:<%=value%></h1>
```

- * 还可以在 `<body>` 标签里写入 `<input>` 标签

导入jsp包



查看home.jsp

