

author: Zilin Xu

date : 2023.9.20

link: 【JavaWeb视频教程（JSP/Servlet/上传/下载/分页/MVC/三层架构/Ajax）】

https://www.bilibili.com/video/BV18s411u7EH?vd_source=c13765adc65e1127b9523161e34f9a99

第一阶段：入门

0.

JSP: 动态网页

静态/动态：

- 不要和是否有动感混为一谈
- 是否 随着 时间，地点，用户操作 的 改变而改变

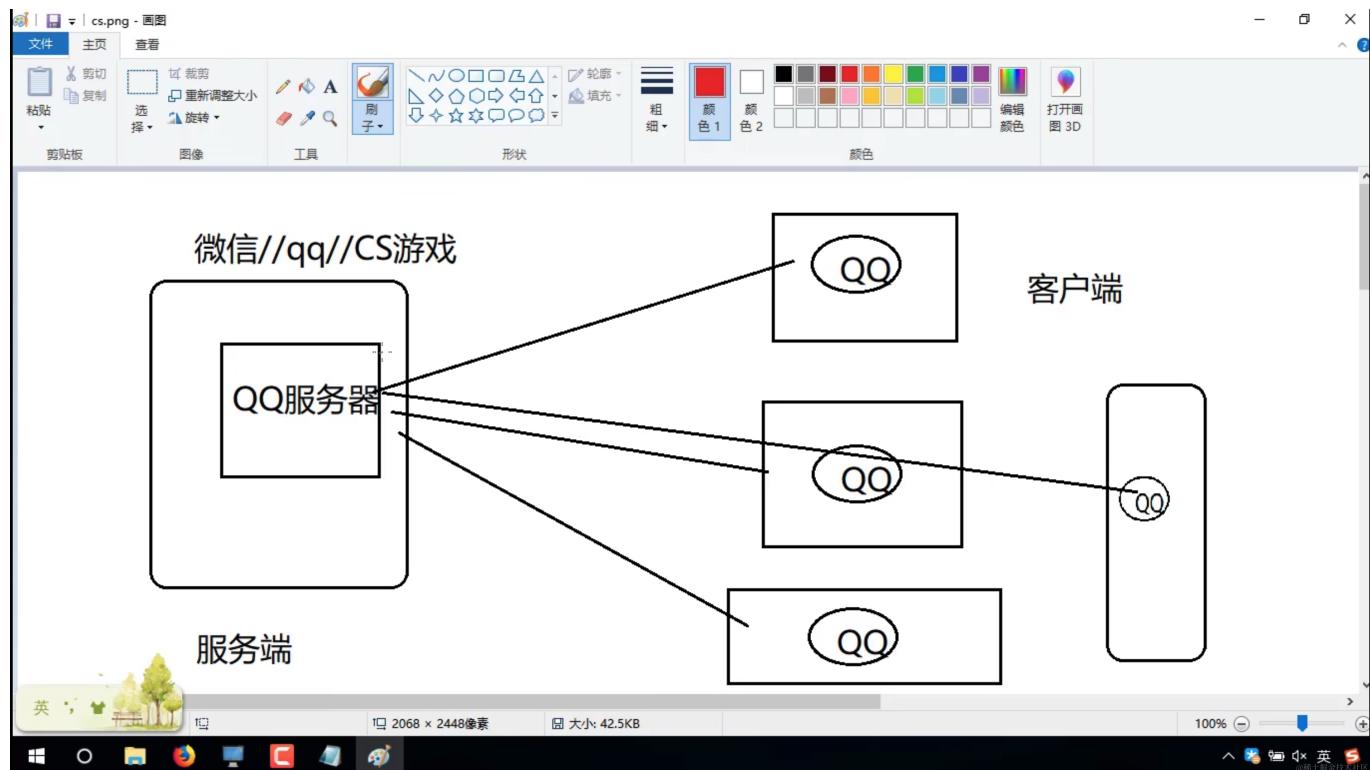
e.g.

静态： 一篇公众号

动态： 查看天气的网站

1. 架构

CS: Client Server



不足

1. 如果软件要升级，那么所有设备都要升级
2. 维护麻烦：需要维护每一台 客户端软件

3. 每一台客户端都需要安装客户端软件

我们玩QQ，需要安装

优势

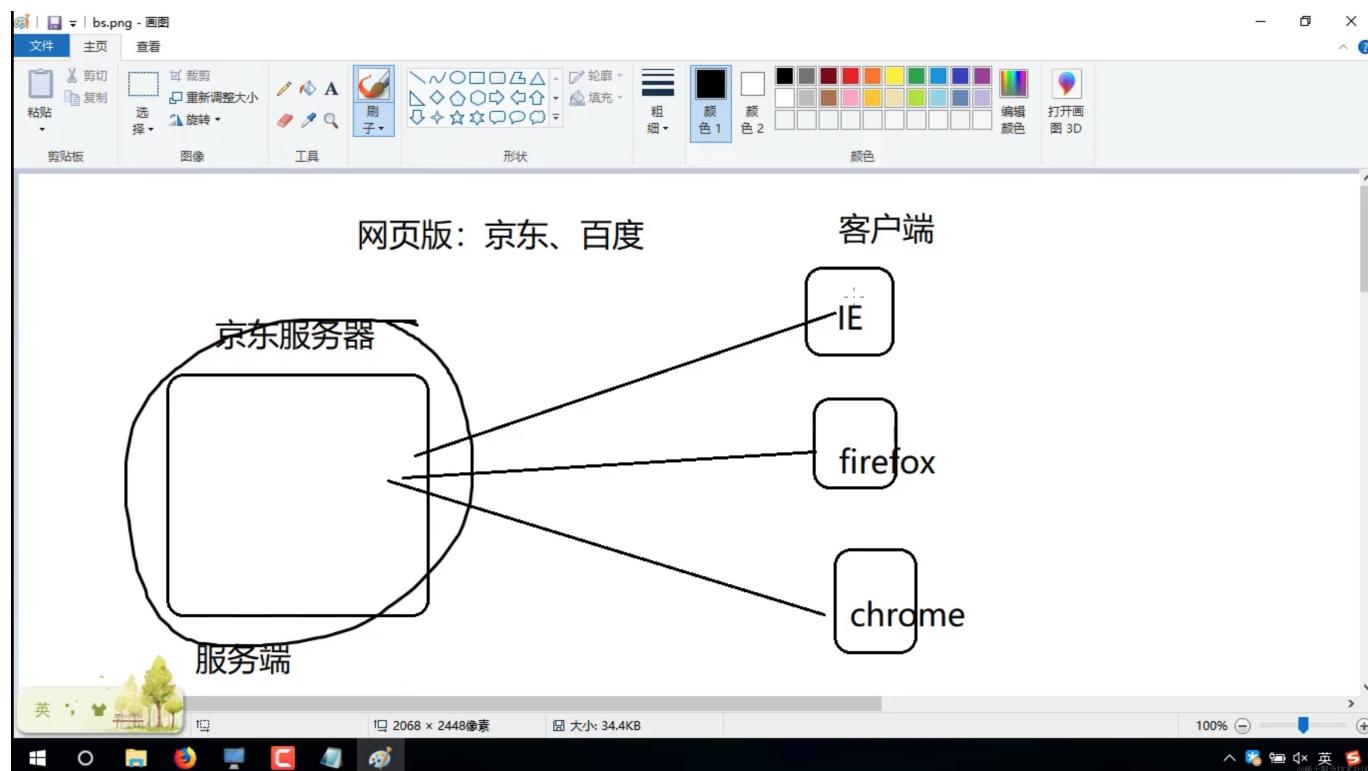
美观

BS: Browser Server

e.g.

网页版：京东，百度

客户端通过浏览器直接访问服务端



注意： BS和CS各有优势

2.tomcat解压后目录

bin: 可执行文件 (startup.bat shutdown.bat)

conf: 配置文件 (server.xml)

lib: tomcat依赖的jar文件

log: 日志文件 (记录出错等信息)

temp: 临时文件

webapps: tomcat的可执行项目 (我们开发完的项目放入该目录)

work: 存放由jsr翻译成的java, 以及编译成的class文件

jsp -> Java -> class

3.配置

1. 配置jdk (必须配置JAVA_HOME) java_home classPath path

2. 配置catalina_home

之后打开 terminal 输入

```
cd /Users/xzl/Desktop/conf/apache-tomcat-8.5.93/bin
```

进入到tomcat的bin文件夹

再输入

```
./catalina.sh start
```

会看到

```
bin - xzl@ersandeMBP - ..at-8.5.93/bin -- zsh - 96x28
bin git:(main) ✘ ./catalina.sh start

Using CATALINA_BASE:      /Users/xzl/Desktop/conf/apache-tomcat-8.5.93
Using CATALINA_HOME:      /Users/xzl/Desktop/conf/apache-tomcat-8.5.93
Using CATALINA_TMPDIR:    /Users/xzl/Desktop/conf/apache-tomcat-8.5.93/temp
Using JRE_HOME:           /Users/xzl/Library/Java/JavaVirtualMachines/openjdk-18.0.2.1/Contents/Home
Using CLASSPATH:          /Users/xzl/Desktop/conf/apache-tomcat-8.5.93/bin/bootstrap.jar:/Users/xzl/Desktop/conf/apache-tomcat-8.5.93/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.

bin git:(main) ✘ |
```

可以在浏览器中输入 <http://localhost:8080> 访问 Tomcat 的欢迎页面。如果成功打开了页面，那么 Tomcat 就正常运行了。

关闭Tomcat

```
./catalina.sh stop
```

常见错误：可能与其他服务的端口号冲突

Tomcat端口号默认8080，太常见了，建议修改

修改端口号：

到conf目录下的server.xml的第69行

```
62 <!-- A "Connector" represents an endpoint by which requests are received
63     and responses are returned. Documentation at :
64     Java HTTP Connector: /docs/config/http.html
65     Java AJP Connector: /docs/config/ajp.html
66     APR (HTTP/AJP) Connector: /docs/apr.html
67     Define a non-SSL/TLS HTTP/1.1 Connector on port 8080
68 -->
69 <Connector port="8888" protocol="HTTP/1.1"
70   connectionTimeout="20000"
71   redirectPort="8443"
72   maxParameterCount="1000"
73   />
74 <!-- A "Connector" using the shared thread pool-->
75 <!--
76 <Connector executor="tomcatThreadPool"
77   port="8080" protocol="HTTP/1.1"
78   connectionTimeout="20000"
79   redirectPort="8443"
80   maxParameterCount="1000"
81   />
82 -->
83 <!-- Define an SSL/TLS HTTP/1.1 Connector on port 8443
84   This connector uses the NIO implementation. The default
```

@稀土掘金技术社区

4. 访问

<http://localhost:8888/>

The screenshot shows the Apache Tomcat 8.5.93 homepage. At the top, there's a navigation bar with links like Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. Below the navigation is the Apache logo. A green banner at the top says "If you're seeing this, you've successfully installed Tomcat. Congratulations!". To the left is a cartoon cat logo. In the center, there's a section titled "Recommended Reading" with links to Security Considerations How-To, Manager Application How-To, and Clustering/Session Replication How-To. To the right are three buttons: Server Status, Manager App, and Host Manager. Below this, there's a "Developer Quick Start" section with links to Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, Servlet Specifications, and Tomcat Versions. The main content area is divided into three columns: "Managing Tomcat", "Documentation", and "Getting Help". The "Managing Tomcat" column has links to Release Notes, Changelog, and Deployment Guide. The "Documentation" column has links to Tomcat 8.5 Documentation, Tomcat 8.5 Configuration, and Tomcat Wiki. The "Getting Help" column has links to FAQ and Mailing Lists, and a list of mailing lists: tomcat-announce, tomcat-users, taglibs-user, and tomcat-dev.

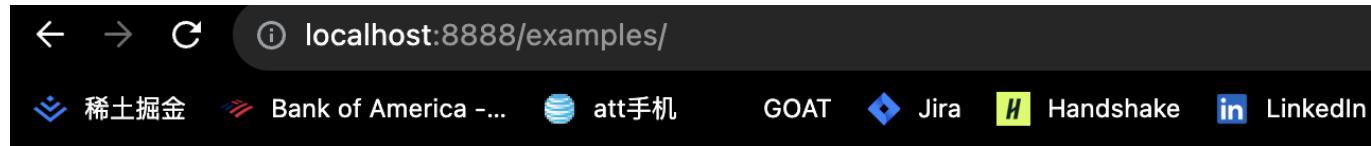
这个界面是从哪里来的呢

答：是 [/apache-tomcat-8.5.93/webapps](#) 这个目录下的 ROOT，这是默认的

The screenshot shows a file explorer window displaying the contents of the `/apache-tomcat-8.5.93/webapps` directory. The window title is "webapps". Inside, there are five subfolders: "docs", "examples", "host-manager", "manager", and "ROOT". The "ROOT" folder is highlighted. The file explorer interface includes standard toolbar buttons for back/forward, search, and file operations.

如果要访问其他的，直接在 [http://localhost:8888/](#) 后面加东西就好，比如：

<http://localhost:8888/examples/>



Apache Tomcat Examples

- [Servlets examples](#)
- [JSP Examples](#)
- [WebSocket Examples](#)

@稀土掘金技术社区

那么问题来了，这个又是哪里来的呢？

答：examples里面的WEB-INF里面的web.xml中，第406行

```
399     <async-supported>true</async-supported>
400   </servlet>
401   <servlet-mapping>
402     <servlet-name>numberwriter</servlet-name>
403     <url-pattern>/servlets/nonblocking/numberwriter</url-pattern>
404   </servlet-mapping>
405
406   <welcome-file-list>
407     <welcome-file>index.html</welcome-file>
408     <welcome-file>index.xhtml</welcome-file>
409     <welcome-file>index.htm</welcome-file>
410     <welcome-file>index.jsp</welcome-file>
411   </welcome-file-list>
412
413   <!-- Websocket examples -->
414   <listener>
415     <listener-class>websocket.drawboard.DrawboardContextListener</listener-class>
416   </listener>
417
418   <error-page>
419     <error-code>403</error-code>
420     <location>WEB-INF/samples/403.jsp</location>
```

@稀土掘金技术社区

如果没有第一个文件就第二个，没有第二个就第三个

要是把index.html删了就会404

复习八股：

200:一切正常

300/301:页面重定向（跳转）

403: 权限不足（如果访问a目录，但是a目录设置不可见）

404: 资源不存在

500:服务器内部错误（代码有误）

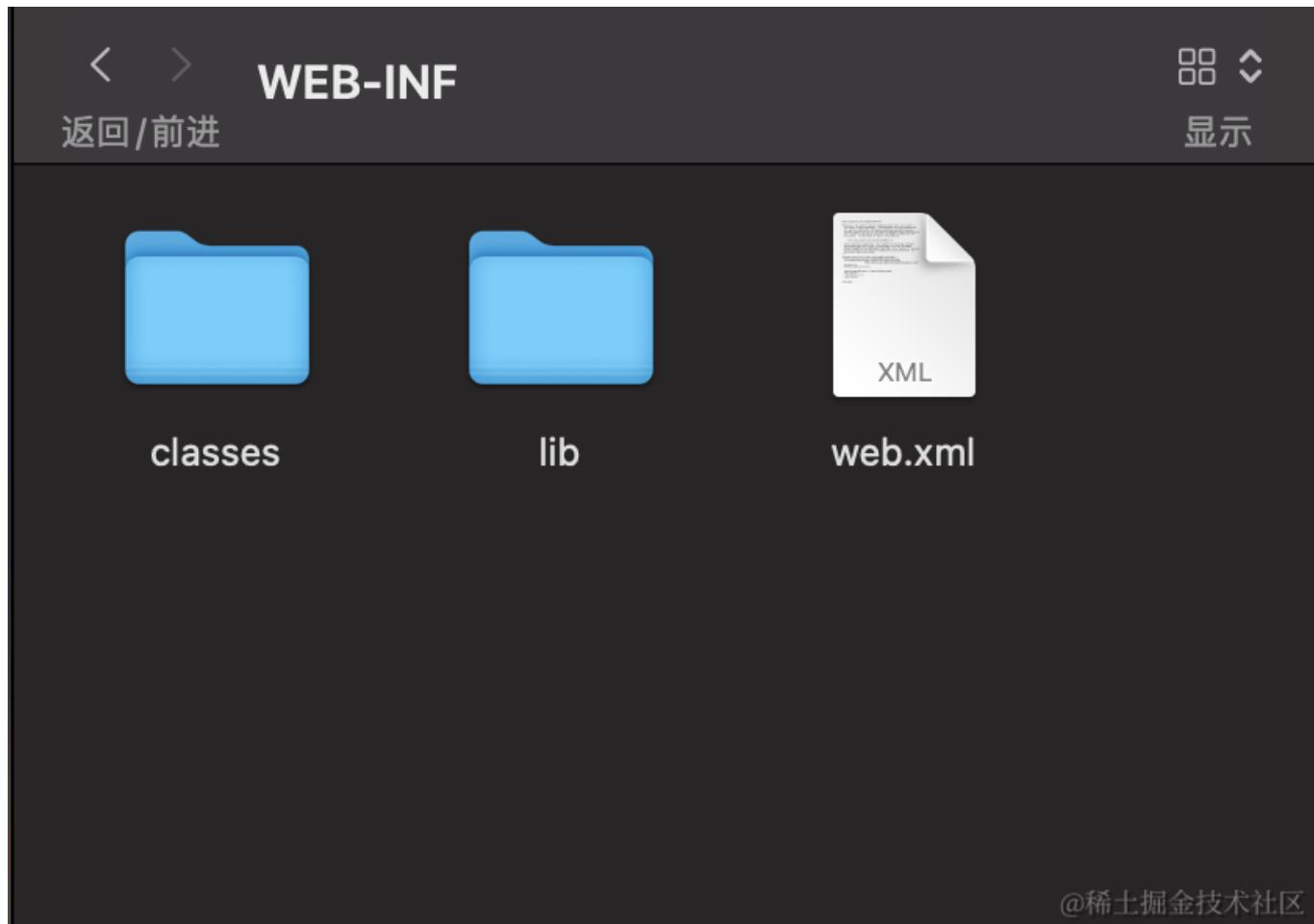
其他编码：积累

5. 创建自己的项目

在webapps目录下创建JspProject

要遵循一些规则：

- WEB-INF文件夹一定要有！从ROOT里面cv过来，这个里面要有.xml文件，以及：
 - 一个classes文件夹
 - 上面提到的字节码文件
 - 一个lib文件夹
 - 三方依赖库，比如一些引用的jar包

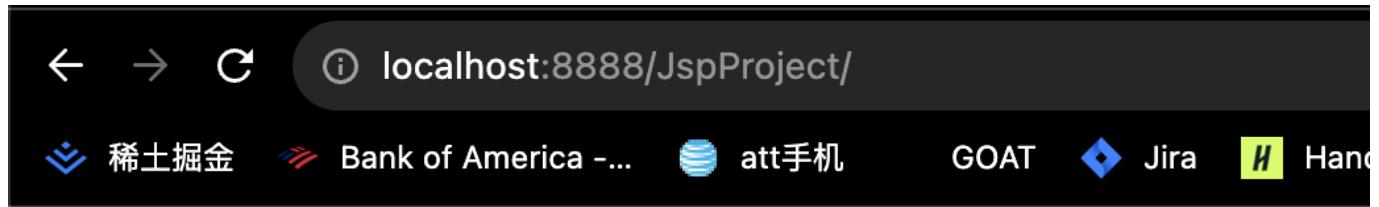


回到JspProject， 创建index.jsp

```
<html>
    <head>
        <title>my first jsp project</title>
    </head>

    <body>
        hello jsp ...
        <%
            out.print("hello");
        %>
    </body>
</html>
```

启动tomcat后把网址改成JspProject



hello jsp ... hello

@稀土掘金技术社区

jsp:在html中嵌套Java代码<% %>

如何指定打开哪个文件

在项目/WEB-INF/web.xml中设置 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>
```

什么时候会报错 500

Java代码写错了

比如：把上面的.jsp里面的out.print("hello"); 分号去掉

类型 异常报告

消息 无法为JSP编译类:

描述 服务器遇到一个意外的情况，阻止它完成请求。

例外情况

```
org.apache.jasper.JasperException: 无法为JSP编译类:

JSP文件: [/index.jsp] 的第 [9] 行发生了一个错误
Syntax error, insert ";" to complete Statement
6:      <body>
7:          hello jsp ...
8:          <%
9:              out.print("hello")
10:             %>
11:         </body>
12:     </html>

Stacktrace:
    org.apache.jasper.compiler.DefaultErrorHandler.javacError(DefaultErrorHandler.java:102)
    org.apache.jasper.compiler.ErrorDispatcher.javacError(ErrorDispatcher.java:213)
    org.apache.jasper.compiler.JDTCompiler.generateClass(JDTCompiler.java:600)
    org.apache.jasper.compiler.Compiler.compile(Compiler.java:381)
    org.apache.jasper.compiler.Compiler.compile(Compiler.java:351)
    org.apache.jasper.compiler.Compiler.compile(Compiler.java:335)
    org.apache.jasper.JspCompilationContext.compile(JspCompilationContext.java:597)
    org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:398)
    org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:383)
    org.apache.jasper.servlet.JspServlet.service(JspServlet.java:331)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:583)
```

@稀土掘金技术社区

6. 虚拟路径

Tomcat怎么知道你默认要访问webapps里面的内容呢

答：虚拟路径

方式1：

缺点：需要重启

在conf/server.xml中161行

```

157     <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
158         resourceName="UserDatabase"/>
159     </Realm>
160
161     <Host name="localhost" appBase="webapps"
162         unpackWARs="true" autoDeploy="true">
163
164     <!-- SingleSignOn valve, share authentication between web applications
165         Documentation at: /docs/config/valve.html -->
166     <!--
167     <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
168     -->
169
170     <!-- Access log processes all example.
171         Documentation at: /docs/config/valve.html
172         Note: The pattern used is equivalent to using pattern="common" -->
173     <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
174         prefix="localhost_access_log" suffix=".txt"
175         pattern="%h %l %u "%r" %s %b" />
176
177     </Host>
178 </Engine>

```

@稀土掘金技术社区

最常用用法

将web项目放到webapps以外的目录下

docBase: 实际路径

path: 虚拟路径 (绝对路径, 相对路径相对于webapps)

```

<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">

    <!-- SingleSignOn valve, share authentication between web
        applications
        Documentation at: /docs/config/valve.html -->
    <!--
    <Valve className="org.apache.catalina.authenticator.SingleSignOn"
/>
    -->

    <!-- Access log processes all example.
        Documentation at: /docs/config/valve.html
        Note: The pattern used is equivalent to using
        pattern="common" -->
    <Valve className="org.apache.catalina.valves.AccessLogValve"
        directory="logs"
            prefix="localhost_access_log" suffix=".txt"
            pattern="%h %l %u "%r" %s %b" />

    <Context docBase="" path="/JspProject" />

</Host>

```

上面 /JspProject 是相对路径，相对于 webapps，等同写法：

/Users/xzl/Desktop/conf/apache-tomcat-8.5.93/webapps/JspProject

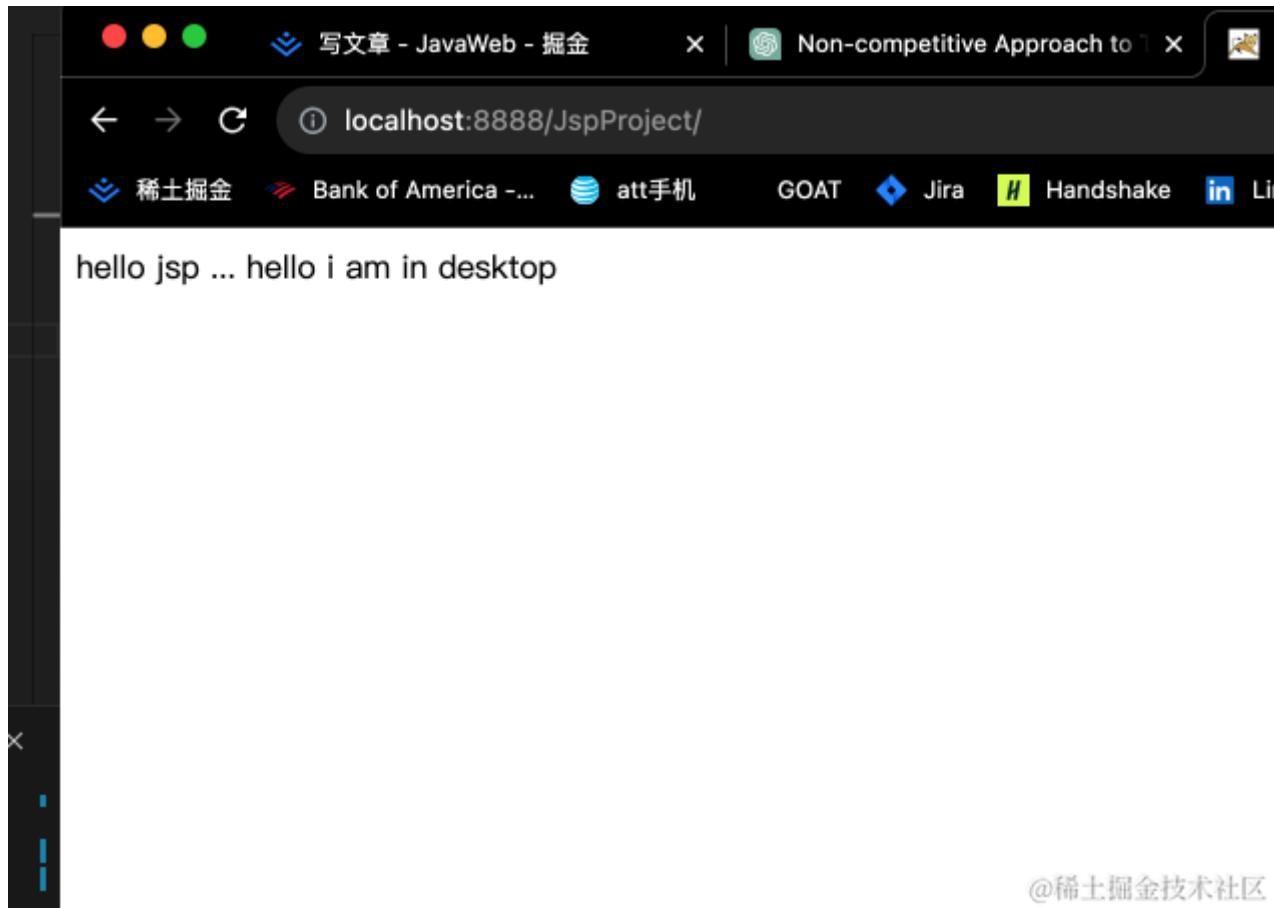
一般来说用相对路径

把 webapps/JspProject 搬到桌面，上面 17 行改成

```
<Context docBase="/Users/xzl/Desktop/JspProject" path="/JspProject" />
```

这一行的含义是，当我们访问 path 的时候 就访问 docBase

就改好了，能正常运行

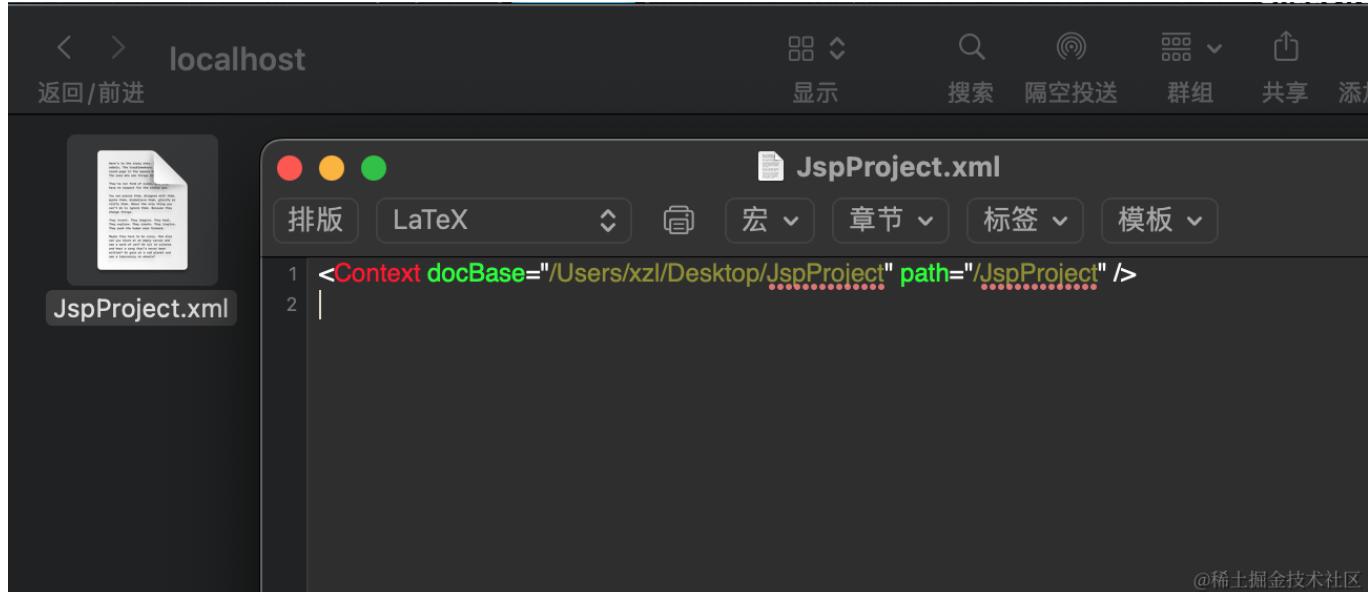


方式2：

在 /conf/Catalina/localhost 中创建 项目名.xml

```
<Context docBase="/Users/xzl/Desktop/JspProject" path="/JspProject" />
```

把这一行放过去就行



两种方式都很常见

第二种方法还有一招：把 `.xml` 文件名字改成 `ROOT.xml`

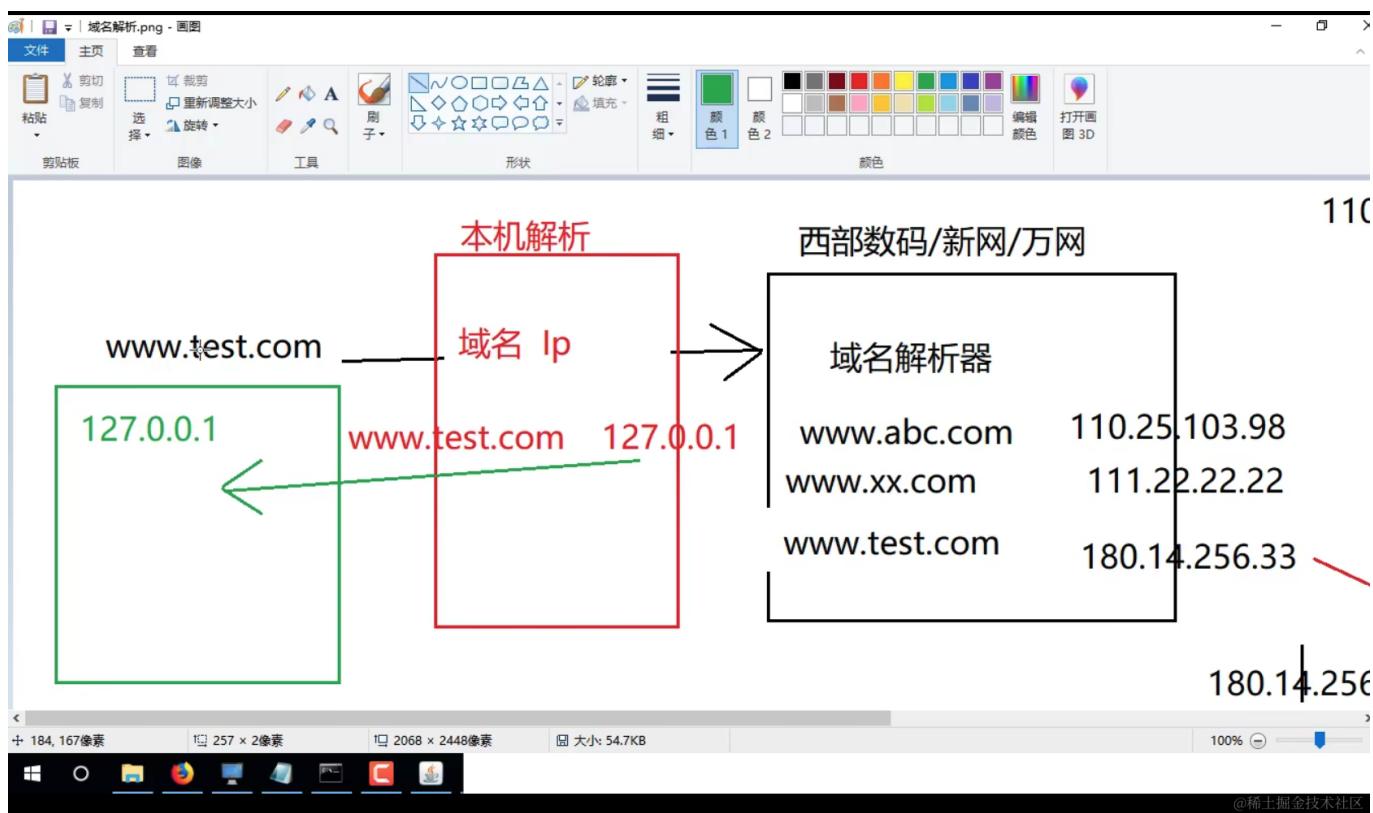
这样的话我们访问 `http://localhost:8888/` 就行，因为启动 Tomcat 的时候它默认去 `ROOT.xml` 中找映射

一般不这样干

7. 虚拟主机

通过 `www.test.com` 访问本机

补充八股 again！



在访问一个网址后，会先在本机解析，如果本机有那么就直接加载本机的ip地址。如果没有，比如 www.jd.com，那么就回去域名解析器中找对应的ip地址。

127.0.0.1 是localhost的IP地址

怎么在mac中查看localhost的IP地址？

ifconfig lo0 | grep "inet "

步骤

0. 先在server.xml中配置

```

conf > server.xml
140      -->
141      <Engine name="Catalina" defaultHost="www.test.com">
142
143          <!--For clustering, please take a look at documentation at:
144              /docs/cluster-howto.html (simple how to)
145              /docs/config/cluster.html (reference documentation) -->
146          <!--
147          <Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster"/>
148          -->
149
150          <!-- Use the LockOutRealm to prevent attempts to guess user passwords
151              | via a brute-force attack -->
152          <Realm className="org.apache.catalina.realm.LockOutRealm">
153              <!-- This Realm uses the UserDatabase configured in the global JNDI
154                  resources under the key "UserDatabase". Any edits
155                  that are performed against this UserDatabase are immediately
156                  available for use by the Realm. -->
157              <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
158                  resourceName="UserDatabase"/>
159          </Realm>
160
161          <Host appBase="/Users/xzl/Desktop/JspProject" name="www.test.com">
162              <Context docBase="/Users/xzl/Desktop/JspProject" path="/" />
163
164          </Host>
165
166
167          <Host name="localhost" appBase="webapps"
168              | unpackWARs="true" autoDeploy="true">
169
170              <!-- SingleSignOn valve, share authentication between web applications
171                  | Documentation at: /docs/config/valve.html -->
172              <!--
173              <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
174              -->
175
176              <!-- Access log processes all example.
177                  | Documentation at: /docs/config/valve.html
178                  | Note: The pattern used is equivalent to using pattern="common" --
179              <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
@博士掘金技术社区
```

先把141行的defaultHost改成我们想要的www.test.com

再把下面的Host依样画葫芦的写，值得注意的是Context中的path 我们写的是"/"而不是 "JspProject" 这是因为我们这样的话就可以简化我们的网址。如果还是写 "JspProject" 的话那我们输入的网址就要是www.test.com/JspProject了。所以这里我们简单一点。

1. 添加127.0.0.1到www.test.com的映射，方法如下：

要将 127.0.0.1 映射到 www.test.com，你可以通过修改你的系统的 hosts 文件来实现。

以下是具体步骤：

1. 打开终端应用程序。
2. 输入以下命令以编辑 hosts 文件：

```
sudo nano /etc/hosts
```

你可能需要输入管理员密码以确认。

3. 在打开的文本编辑器中，找到一个空白行（或在文件末尾添加一个新行），然后输入：

```
127.0.0.1 www.test.com
```

这将把 `www.test.com` 映射到本地主机的 IP 地址 `127.0.0.1`。

4. 保存并退出文本编辑器。如果你使用的是 nano 编辑器，可以按下 `Ctrl + X`，然后按 `Y` 确认保存，最后按 `Enter` 退出。
5. 刷新 DNS 缓存：

在终端中运行以下命令以刷新 DNS 缓存：

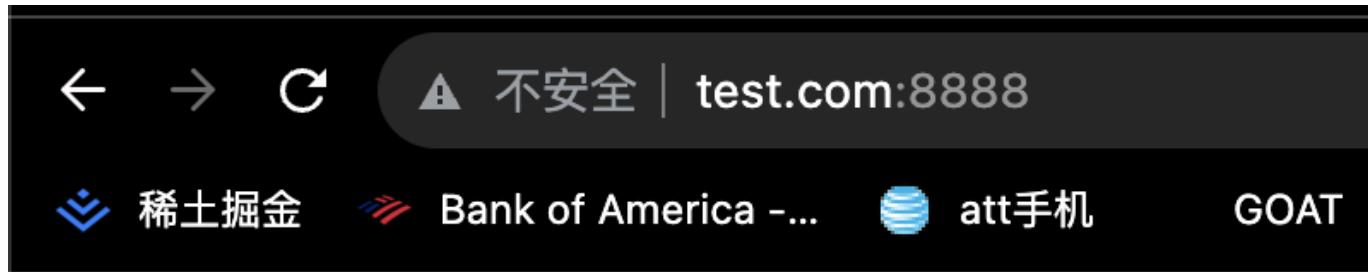
```
sudo dscacheutil -flushcache
```

或者在较新版本的 macOS 中，你可以使用以下命令：

```
sudo killall -HUP mDNSResponder
```

现在，当你在浏览器中访问 `www.test.com` 时，它将会映射到本地主机 `127.0.0.1`。请注意，这只会在你的计算机上有效，而不会影响到其他设备。

这个时候我们访问 `www.test.com:8888` 就会出现



hello jsp ... hello i am in desktop

@稀土掘金技术社区

如果需要省略端口号的话，就去把8888改成80（虽然我没成功）

回顾一下整个流程

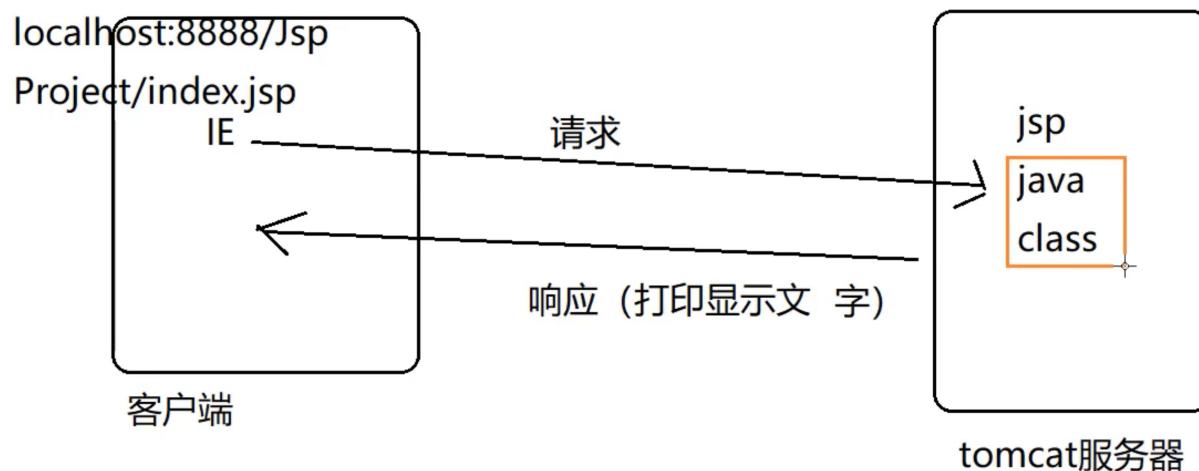
www.test.com -> host找映射关系 -> server.xml找Engine的defaultHost -> 通过"/"(可以省略)映射到/Users/xzl/Desktop/JspProject

这个只是学习一下，一般开发不会用，将上面复原！

8.JSP执行流程

第一次访问：

JSP -> Java -> class



.java 和 .class文件都放在

/Users/xzl/Desktop/offer comes!!!/notes/javaweb 2023.9.20/apache-tomcat-8.5.93/work/Catalina/localhost/JspProject/org/apache/jsp中

第二次访问：直接访问.class文件

注意

如果服务端代码修改了，将会再访问时重新翻译，编译

对比一下：

The screenshot shows a Java IDE interface with two code editors side-by-side. On the left, the 'index.jsp' file is displayed, containing JSP code with scriptlets and comments. On the right, the corresponding generated 'index_jsp.java' file is shown, which is a standard Java servlet. The Java code includes imports for javax.servlet, javax.servlet.http, and javax.servlet.jsp, and it implements the HttpServlet class. It handles the logic from the JSP scriptlets, such as printing 'hello jsp ...' and 'offer comes!!!'. The Java code also includes error handling for ServletException.

```

<html>
<head>
    <title>my first jsp project</title>
</head>
<body>
    hello jsp ...
    <%>
    out.print("hello i am in folder offer comes!!!");
</body>
</html>

```

```

private void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.setContentType("text/html");
    response.setCharacterEncoding("UTF-8");
    PrintWriter out = response.getWriter();
    out.println("hello i am in folder offer comes!!!");
}

```

.java文件严格来说是Servlet文件

Jsp和Servlet可以相互转换

因为第一次请求服务端会有翻译和编译的过程，因此比较慢；后续访问可以直接访问class，因此速度较快。但是 如果服务端修改了代码，则再次访问时会重新的翻译和编译

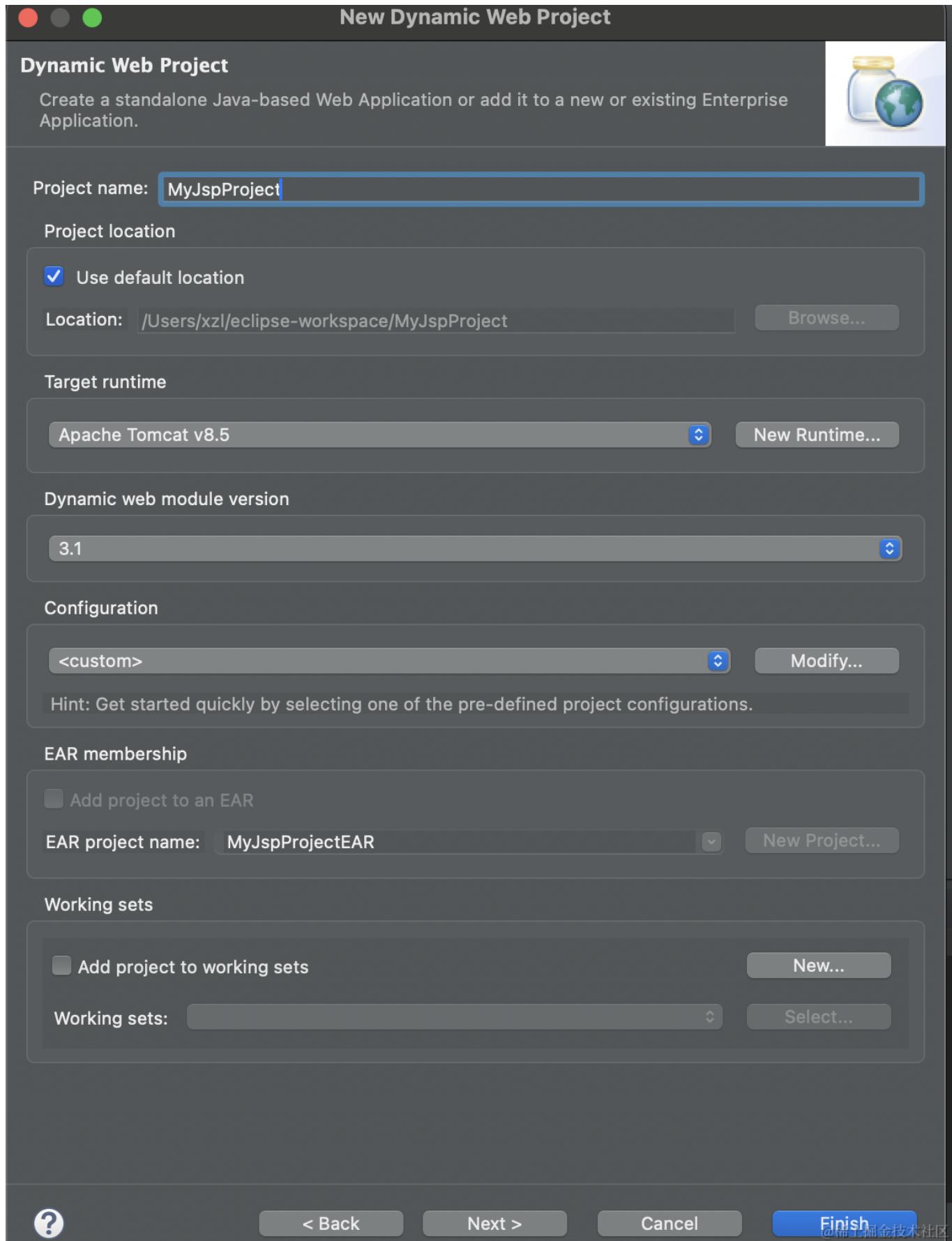
第二阶段： 使用Eclipse开发

1. 使用Eclipse 开发Web项目 (JSP)

- 用Eclipse配置tomcat
 - 打开preference，找到server，选runtime environment
- 在Sever (面板) 里配置tomcat
 - 点window，把server面板show出来
 - 在面板里添加tomcat

在eclipse里创建project

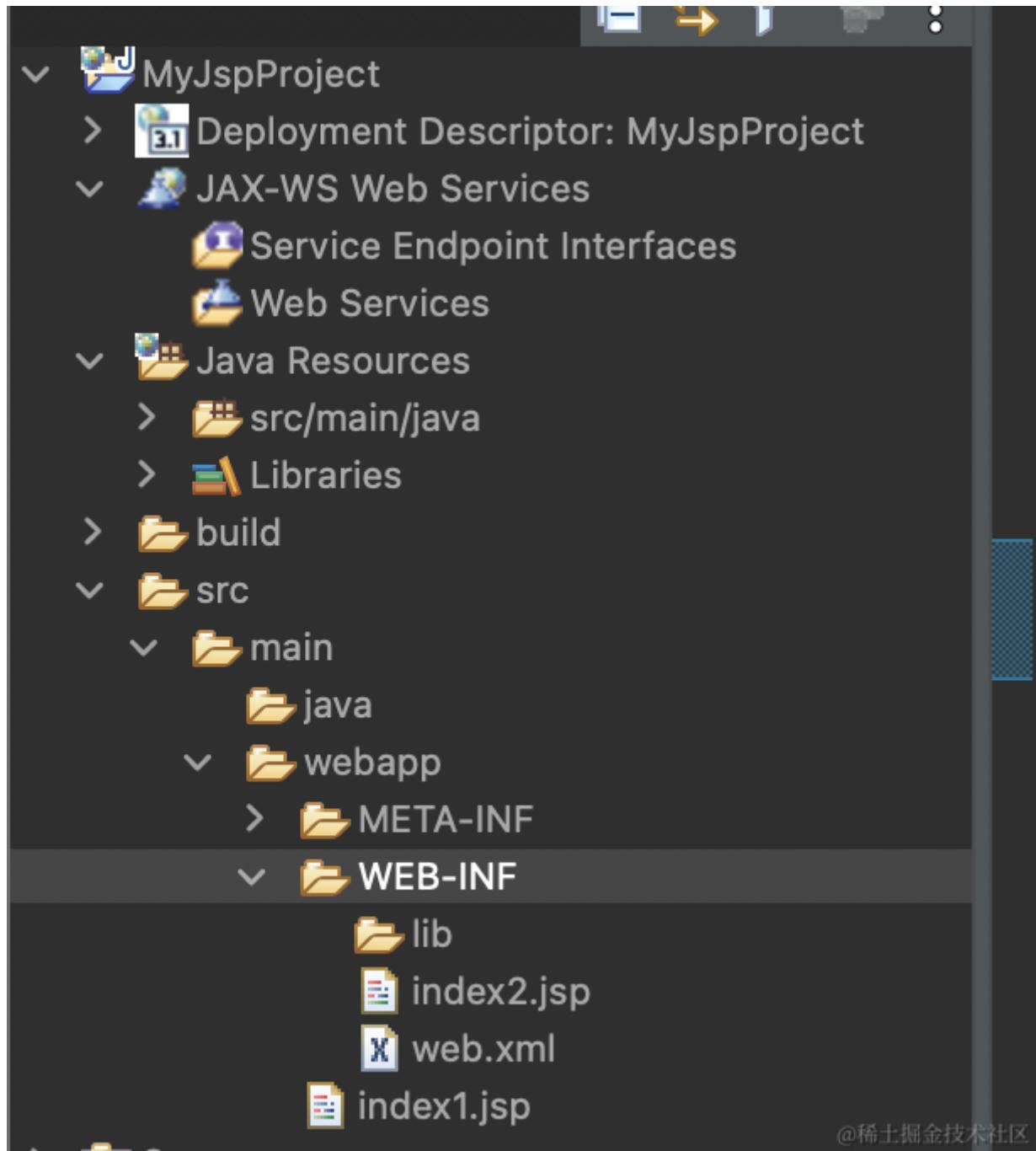
- 点击file, newproject, other, 搜web, 选择Dynamic Web Project
- 在出来的页面里，规定名字，选择tomcat刚才配置的版本，那个version3.1就是 servlet版本，后面会讲



- 然后点next, 在最后把web.xml钩上

补充一个eclipse小知识

如果你的界面混乱了, 点 window -> Perspective -> Reset Perspective



在 webapp 目录下分别创建两个.jsp 文件

注意

第一次创建的时候选 new -> other 然后搜 jsp

添加两行代码

The screenshot shows a code editor with two tabs open:

- index1.jsp**:

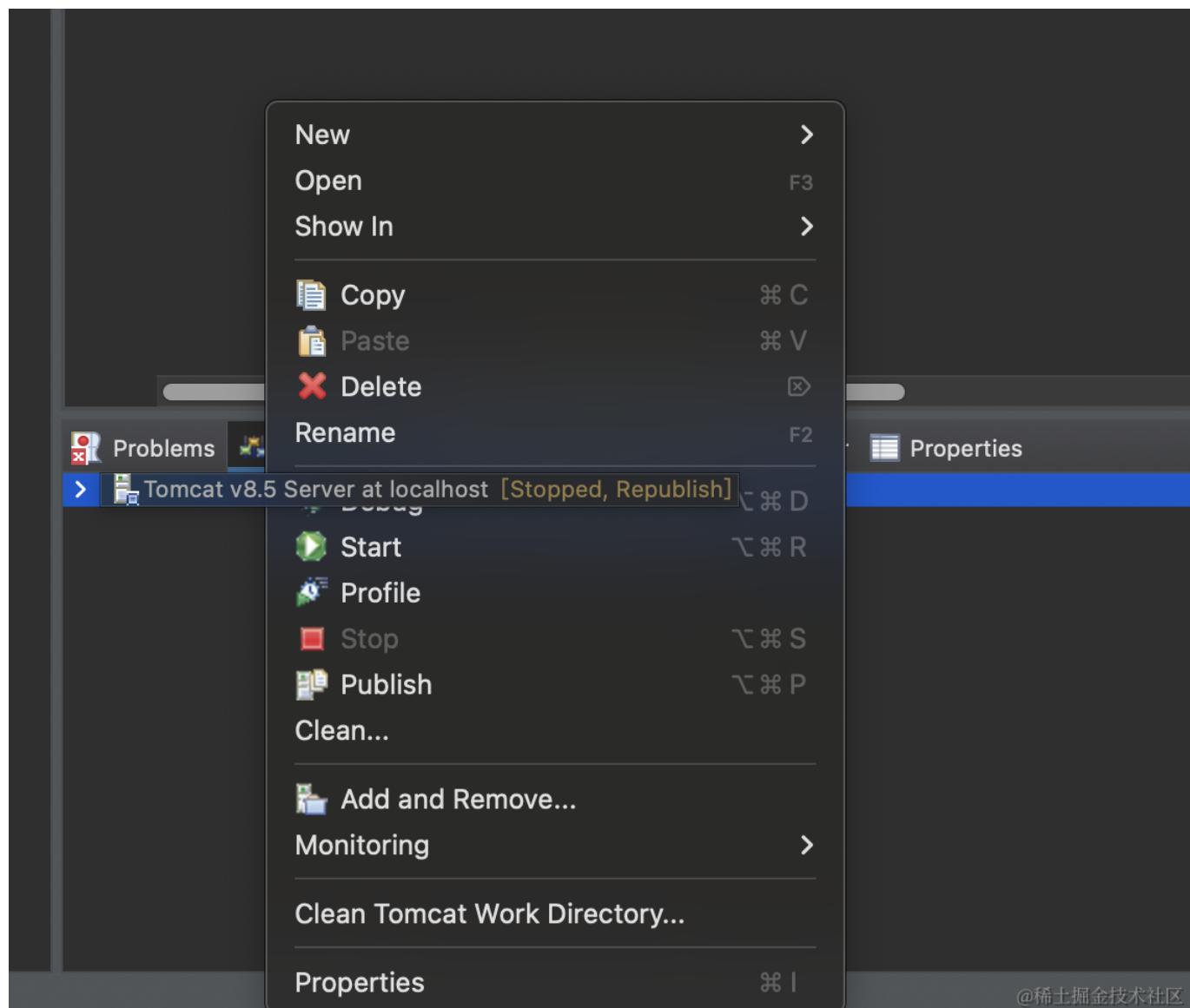
```
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10    hello index1
11 </body>
12 </html>
```
- index2.jsp**:

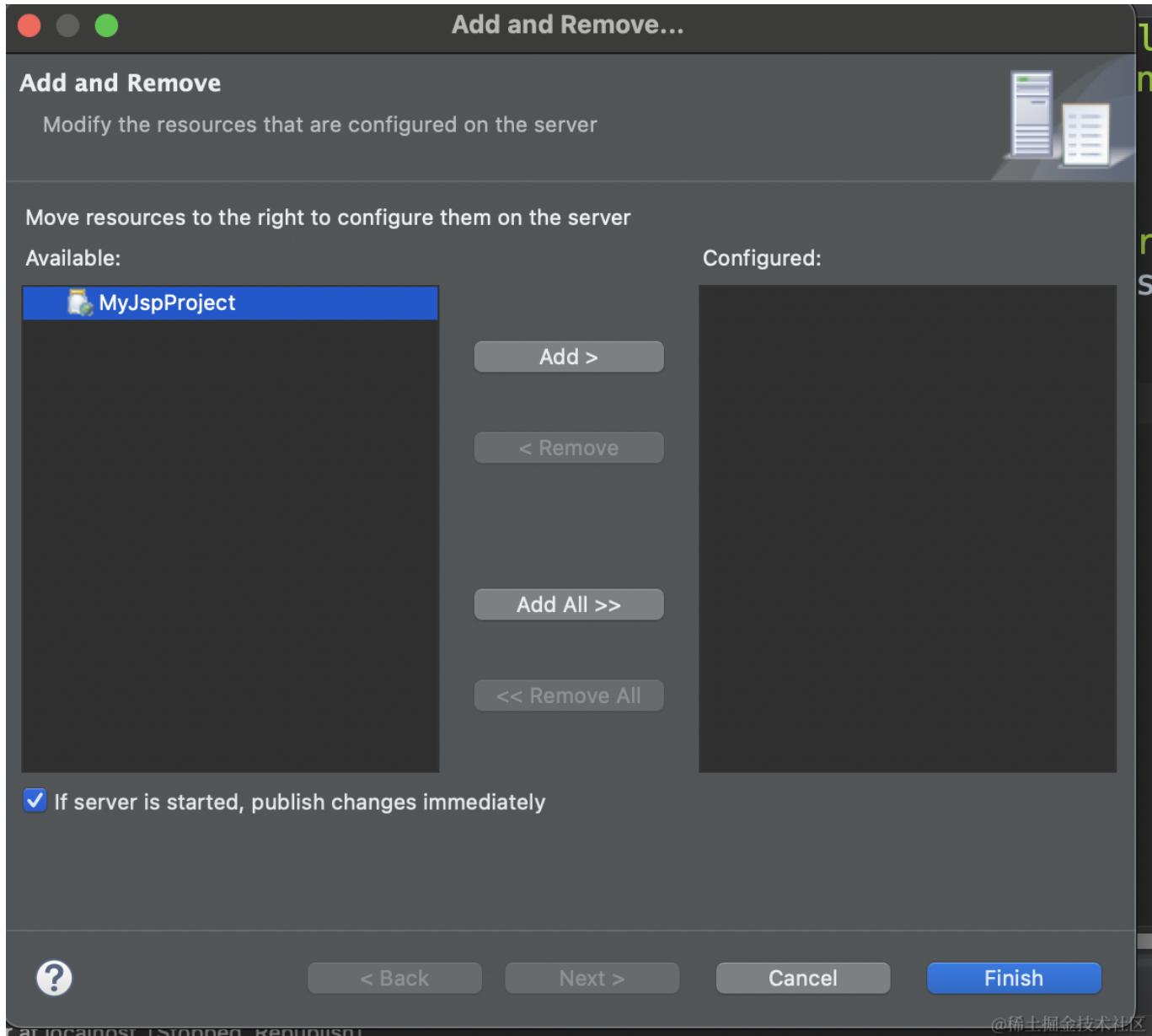
```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="UTF-8">
7 <title>Insert title here</title>
8 </head>
9 <body>
10   hello index2
```

The code editor has a dark theme. The bottom right corner of the window contains the text "@稀土掘金技术社区".

怎么启动呢

在server面板里右击，add and remove，把刚才创建的project加进去就行





在右击，start就行。输入网址<http://localhost:8888/MyJspProject/index1.jsp>

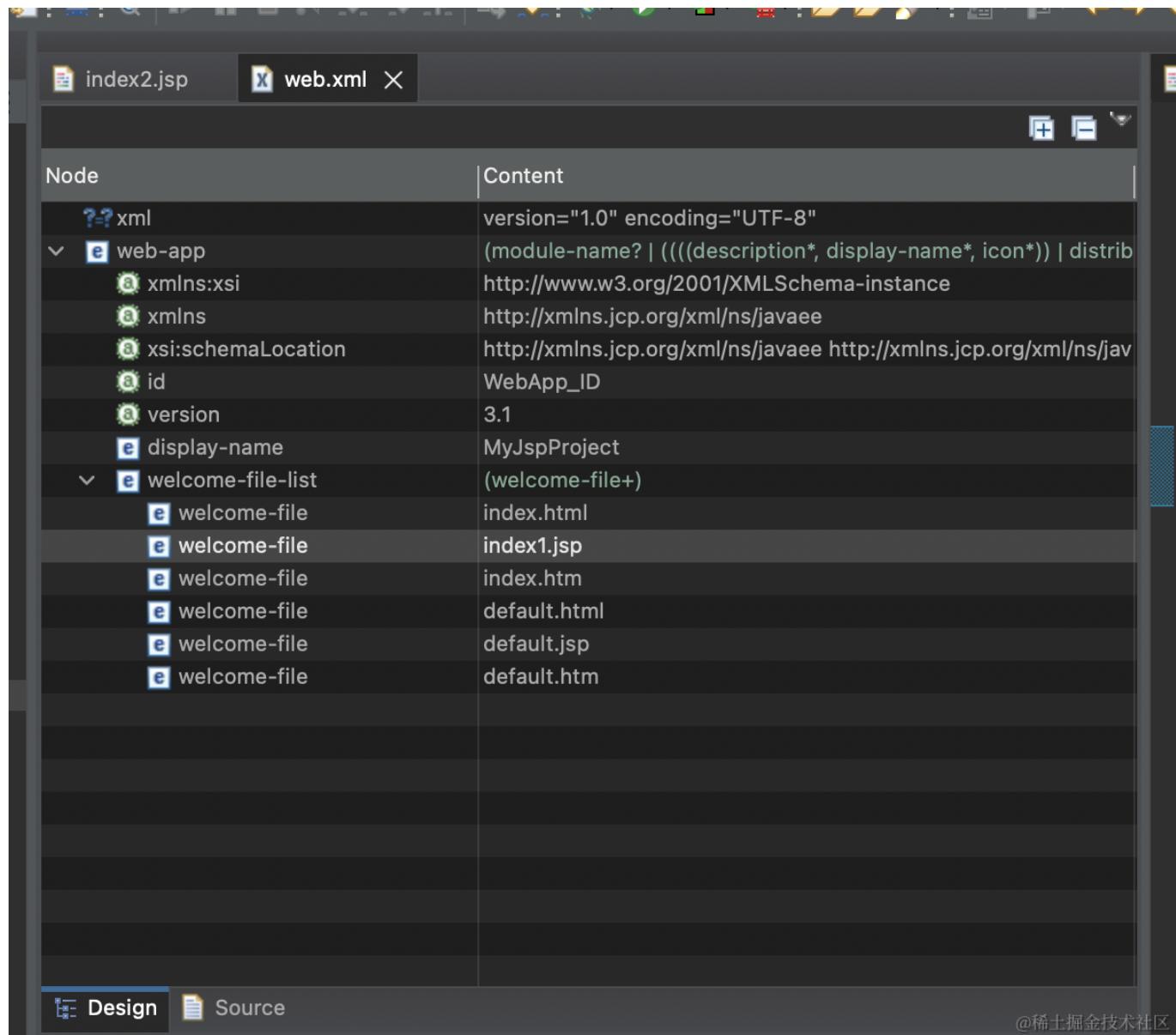


@稀土掘金技术社区

为什么输入<http://localhost:8888/MyJspProject/> 找不到呢404

因为我们没有在web.xml配置

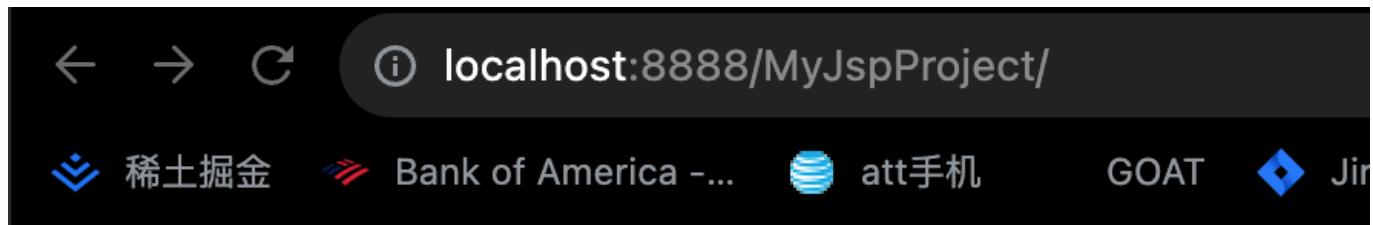
在web.xml中改成index1.jsp，然后在下面面板中restart就行。



The screenshot shows the 'Design' view of the web.xml configuration in an IDE. The 'Content' tab is selected. The configuration includes:

Node	Content
xml	version="1.0" encoding="UTF-8"
web-app	(module-name? (((description*, display-name*, icon*)) distrib
xmlns:xsi	http://www.w3.org/2001/XMLSchema-instance
xmlns	http://xmlns.jcp.org/xml/ns/javaee
xsi:schemaLocation	http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/jav
id	WebApp_ID
version	3.1
display-name	MyJspProject
welcome-file-list	(welcome-file+)
welcome-file	index.html
welcome-file	index1.jsp
welcome-file	index.htm
welcome-file	default.html
welcome-file	default.jsp
welcome-file	default.htm

At the bottom, there are tabs for 'Design' and 'Source'. A watermark '@稀土掘金技术社区' is visible at the bottom right.



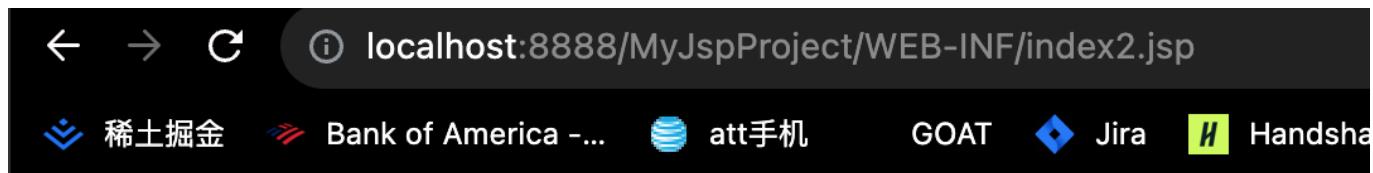
hello index1

@稀土掘金技术社区

值得一提的是

<http://localhost:8888/MyJspProject/> 这个地址是个相对路径, eclipse帮我们配置好了, 意思就是访问目录/webapp下的项目

那我们试一下/WEB-INF/index2.jsp



HTTP状态 404 - 未找到

类型 状态报告

描述 源服务器未能找到目标资源的表示或者是不愿公开一个已经存在的资源表示。

Apache Tomcat/8.5.93

@稀土掘金技术社区

因为出于安全考虑，WEB-INF这个文件夹权限比较高，客户端（浏览器）不能直接访问。只能通过请求转发访问。

注意

并不是任何内部跳转都能访问WEB-INF，原因是，跳转有两种方式

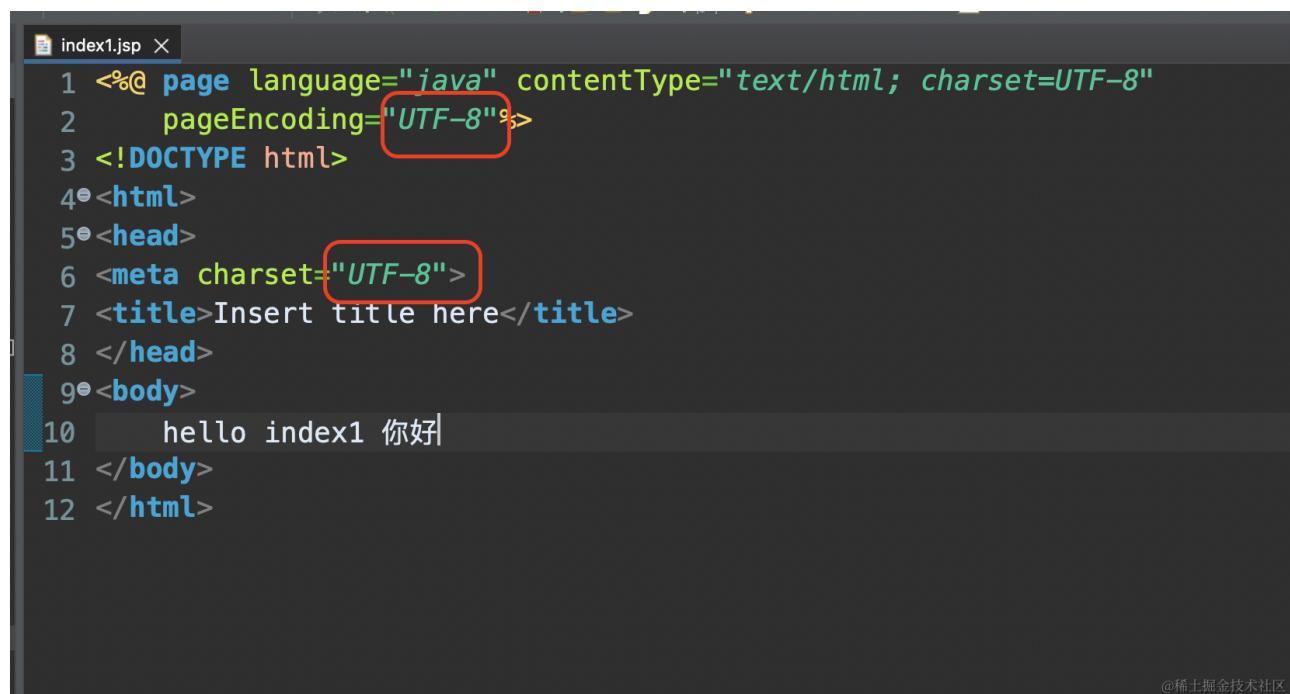
- 请求转发
- 重定向（访问不了）

统一字符集编码

0. 编码分类

1. 设置jsp文件的编码：jsp -> java，配置方法：jsp文件中的pageEncoding属性
2. 设置浏览器读取jsp文件的编码 配置方法：jsp文件中的content属性

一般将上述设置成一样的，推荐使用国际编码，UTF-8



The screenshot shows the Eclipse IDE interface with a Java file named 'index1.jsp' open. The code is as follows:

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5 <meta charset="UTF-8">
6 <title>Insert title here</title>
7 </head>
8 <body>
9 hello index1 你好
10 </body>
11 </html>
```

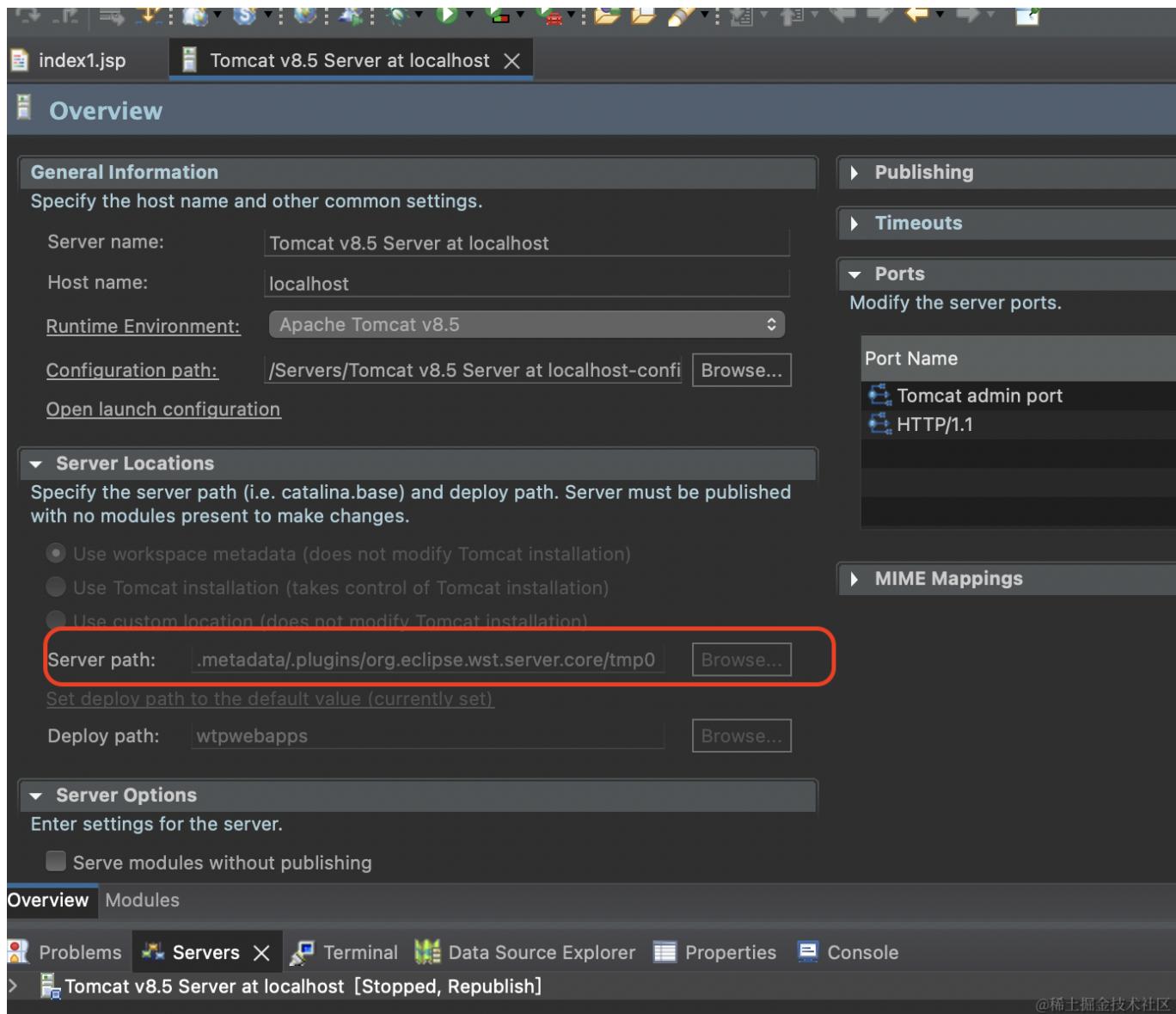
Two specific lines of code are highlighted with red boxes: the 'pageEncoding' attribute in line 1 and the 'charset' attribute in line 6. The code editor has a dark theme, and the highlighted text is in green.

3. 文本编码：jsp文件本身的编码

1. 将整个eclipse中的文件统一设置：preference -> Jsp files -> encoding
2. 设置某一个项目
3. 设置单独文件：右键 properties

来看个小问题

我们如果在我们安装tomcat的文件夹（offer comes/ notes/ javaweb）里面的server.xml里面改了端口号的话，我们发现对我们eclipse里面的项目没有任何影响。这是因为，我们在eclipse中配置tomcat的时候，会自动拷贝一份独立出来。



要选择上面的Use Tomcat installation意思是托管模式

注意，只能在第一次部署tomcat的时候修改这个选项（双击）

JSP页面元素

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

<!DOCTYPE html>

<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
```

```

</head>

<body>

hello index1 你好

<%
String name = "Bruce";
out.println("hello JSP" + name);

%>

</body>

</html>

```

0. 脚本Scriptlet (line 21 - line 25)

1. <% 局部变量、Java语句 %>
2. <%! 定义全局变量，方法 %>
3. <%= 输出表达式%> (和二基本一样，这个不用分号) <%= "hello" + name %>

一般而言修改web.xml、配置文件、java 需要重启tomcat服务器

如果修改jsp/html/css/js， 不需要重启

注意

在jsp里面，println不会回车，如果需要的话要这样

```
out.println("hello JSP" + name + "<br/>");
```

1. 指令 (第一第二行)

page指令 (用得最多，还有两种另外的指令)

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

指定的属性：

1. language: jsp页面使用的脚本语言 (Java)
2. import: 导入类
3. pageEncoding: jsp文件自身编码 jsp->java
4. contentType: 浏览器解析jsp的编码 (一般和上面保持一致)

```

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

<!DOCTYPE html>

```

```
<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

hello index1 你好

<%!

public String name;

public Date date = new Date();

public void init(){

name = "Bruce";

}

%>

<%

init();

out.println("hello JSP");

out.println(name + date);

%>

</body>

</html>
```

2. 注释

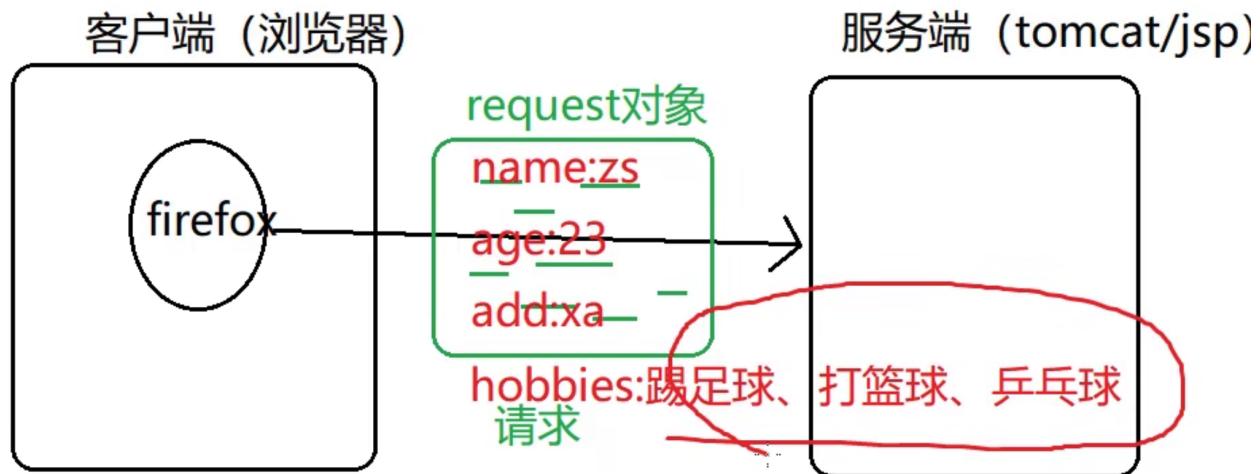
html comment: <!-- --> (可以被客户通过浏览器查看源代码所观察到, 其他两个不行)

java comment: // /* */

jsp comment: <%-- --%>

JSP九大内置对象 (自带的，不需要new也能使用的对象)

- out: 输出对象，向客户端输出
- request: 请求对象，存储客户端向服务端的请求信息



@稀土掘金技术社区

request对象常见方法

```

String getParamater(String name) : 根据请求的字段名key, 返回字段值value
String[] getParamaterValues(String name) : 根据请求的字段名key, 返回多个字段值
value (数组, 比如checkbox多选按钮)
setCharacterEncoding("编码格式utf-8") : 设置请求编码 (tomcat7以前默认iso-8859-1,tomcat8以后改成了utf-8)
getRequestDispatcher("B.jsp").forward(request,response); : 请求转发 页面A -> B
ServletContext getServerContext () : 获取项目的ServletContext对象

```

- pageContext
- response
- session
- application
- config
- page
- exception
-