



Http 协议和 Tomcat 服务器

教学导航

教学目标	
教学方法	

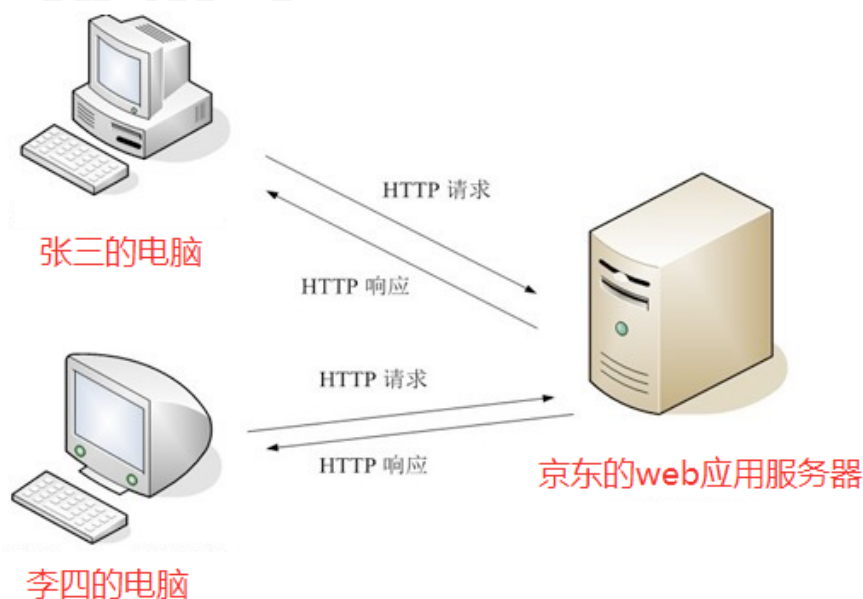
一、Http 协议

1. 什么是 Http 协议

HTTP，超文本传输协议（HyperText Transfer Protocol）是互联网上应用最为广泛的一种网络协议。所有的 WWW 文件都必须遵守这个标准。设计 HTTP 最初的目的是为了提供一种发布和接收 HTML 页面的方法

2. Http 协议的组成

Http 协议由 Http 请求和 Http 响应组成，当在浏览器中输入网址访问某个网站时，你的浏览器会将你的请求封装成一个 Http 请求发送给服务器站点，服务器接收到请求后会组织响应数据封装成一个 Http 响应返回给浏览器。即没有请求就没有响应。





3. Http 请求

编辑一个 form.html 的表单页面，如下：

```
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
    <form action="#" method="post">
        <input type="text" name="username"><br>
        <input type="password" name="password"><br>
        <input type="submit" value="测试http请求"><br>
    </form>
</body>
</html>
```

点击提交按钮，抓包如下：

Http请求

POST /DemoEE/form.html HTTP/1.1	http请求行
Accept: text/html, application/xhtml+xml, */* Referer: http://localhost:8080/DemoEE/form.html Accept-Language: zh-CN User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0) Content-Type: application/x-www-form-urlencoded Accept-Encoding: gzip, deflate Host: localhost:8080 Content-Length: 30 Connection: Keep-Alive Cache-Control: no-cache	
username=zhangsan&password=123	http请求体

1) 请求行

请求方式：POST、GET

请求的资源：/DemoEE/form.html

协议版本：HTTP/1.1

HTTP/1.0，发送请求，创建一次连接，获得一个 web 资源，连接断开。

HTTP/1.1，发送请求，创建一次连接，获得多个 web 资源，保持连接。



2) 请求头

请求头是客户端发送给服务器端的一些信息，使用键值对表示 key : value

常见请求头	描述（红色掌握，其他了解）
Referer	浏览器通知服务器，当前请求来自何处。如果是直接访问，则不会有这个头。常用于：防盗链
If-Modified-Since	浏览器通知服务器，本地缓存的最后变更时间。与另一个响应头组合控制浏览器页面的缓存。
Cookie	与会话有关技术，用于存放浏览器缓存的 cookie 信息。
User-Agent	浏览器通知服务器，客户端浏览器与操作系统相关信息
Connection	保持连接状态。Keep-Alive 连接中，close 已关闭
Host	请求的服务器主机名
Content-Length	请求体的长度
Content-Type	如果是 POST 请求，会有这个头，默认值为 application/x-www-form-urlencoded，表示请求体内容使用 url 编码
Accept :	浏览器可支持的 MIME 类型。文件类型的一种描述方式。 MIME 格式：大类型/小类型[;参数] 例如： text/html , html 文件 text/css , css 文件 text/javascript , js 文件 image/* , 所有图片文件
Accept-Encoding	浏览器通知服务器，浏览器支持的数据压缩格式。如：GZIP 压缩
Accept-Language	浏览器通知服务器，浏览器支持的语言。各国语言（国际化 i18n）

3) 请求体

当请求方式是 post 的时，请求体会有请求的参数，格式如下：

username=zhangsan&password=123

如果请求方式为 get，那么请求参数不会出现在请求体中，会拼接在 url 地址后面

<http://localhost:8080...?username=zhangsan&password=123>



4. Http 响应

Http响应

HTTP/1.1 200 OK **Http响应行**

```
Server: Apache-Coyote/1.1
Accept-Ranges: bytes
ETag: W/"312-1467289802502"
Last-Modified: Thu, 30 Jun 2016 12:30:02 GMT
Content-Type: text/html
Content-Length: 312
Date: Thu, 30 Jun 2016 12:31:12 GMT
```

Http响应头

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
  <form action="#" method="post">
    <input type="text" name="username"><br>
    <input type="password" name="password"><br>
    <input type="submit" value="嫵嫵嫵http璇锋脛"><br>
  </form>
</body>
</html>
```

Http响应体

1) 响应行

Http 协议

状态码：

常用的状态码如下：

200：请求成功。

302：请求重定向。

304：请求资源没有改变，访问本地缓存。

404：请求资源不存在。通常是用户路径编写错误，也可能是服务器资源已删除。

500：服务器内部错误。通常程序抛异常。

状态信息：状态信息是根据状态码变化而变化的

2) 响应头

响应也都是键值对形式，服务器端将信息以键值对的形式返回给客户端

常见请求头	描述
Location	指定响应的路径，需要与状态码 302 配合使用，完成跳转。
Content-Type	响应正文的类型（MIME 类型）



	取值：text/html;charset=UTF-8
Content-Disposition	通过浏览器以下载方式解析正文 取值：attachment;filename=xx.zip
Set-Cookie	与会话相关技术。服务器向浏览器写入 cookie
Content-Encoding	服务器使用的压缩格式 取值：gzip
Content-length	响应正文的长度
Refresh	定时刷新，格式：秒数;url=路径。url 可省略，默认值为当前页。 取值：3;url=www.itcast.cn //三秒刷新页面到 www.itcast.cn
Server	指的是服务器名称，默认值：Apache-Coyote/1.1。可以通过 conf/server.xml 配置进行修改。<Connector port="8080" ... server="itcast"/>
Last-Modified	服务器通知浏览器，文件的最后修改时间。与 If-Modified-Since 一起使用。

3) 响应体

响应体是服务器回写给客户端的页面正文，浏览器将正文加载到内存，然后解析渲染显示页面内容



二、Tomcat 服务器

1. Web 开发中的常见概念

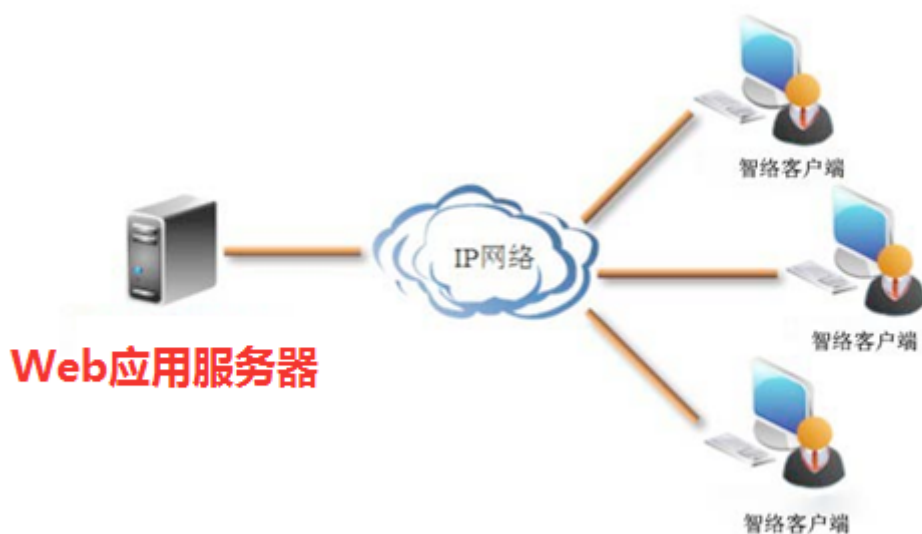
(1) B/S 系统和 C/S 系统

Brower/Server : 浏览器 服务器 系统 ----- 网站

Client/Server : 客户端 服务器 系统 ----- QQ、飞秋、大型游戏

(2) web 应用服务器

供向外部发布 web 资源的服务器软件



(3) web 资源

存在于 web 应用服务器可供外界访问的资源就是 web 资源

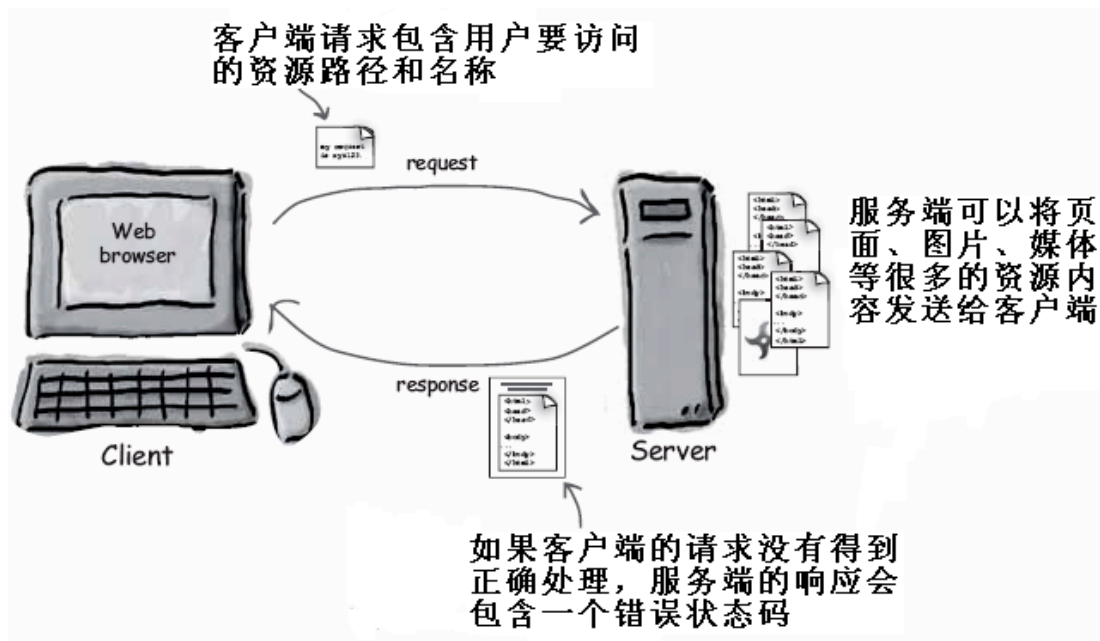
例如：存在于 web 应用服务器内部的 Html、Css、js、图片、视频等

1) 静态资源：指 web 页面中供人们浏览的数据始终是不变。比如：HTML、CSS、JS、图片、多媒体。

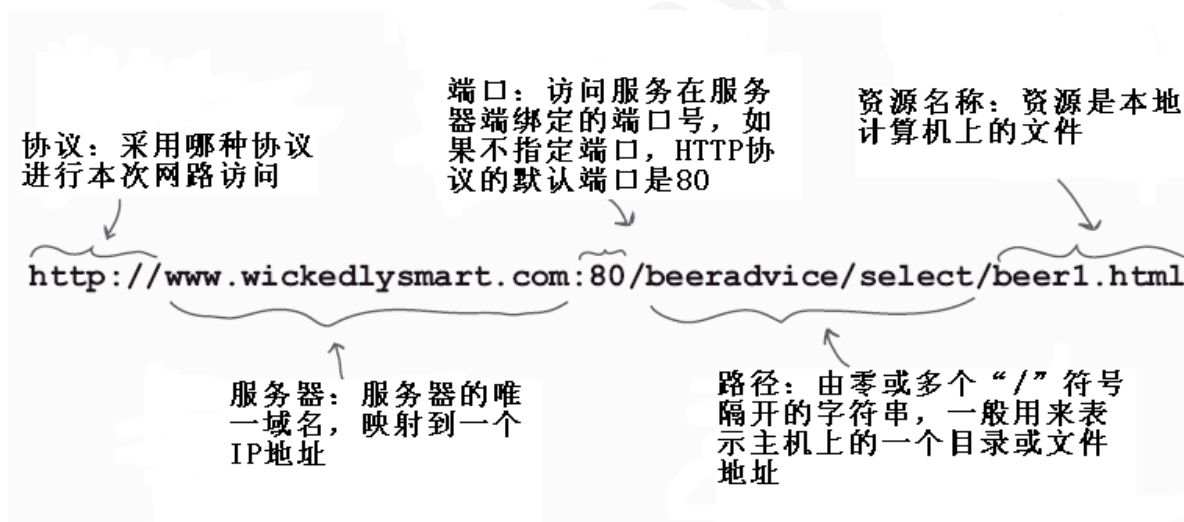
2) 动态资源：指 web 页面中供人们浏览的数据是由程序产生的，不同时间点访问 web 页面看到的内容各不相同。比如：JSP/Servlet、ASP、PHP

javaWEB 领域：动态资源认为通过 java 代码去动态生成 html

(4) 请求和响应



(5) 请求的 URL 地址



2. Web 开发中常用的 web 应用服务器

- 1) weblogic : oracle 公司的大型收费 web 服务器 支持全部 javaEE 规范
- 2) websphere : IBM 公司的大型收费 web 服务器 支持全部的 javaEE 规范
- 3) Tomcat : Apache 开源组织下的 开源免费的中小型的 web 应用服务器 支持 javaEE 中的 servlet 和 jsp 规范



3. Tomcat 的下载与安装

(1) 下载 Tomcat

官网地址：<http://tomcat.apache.org/whichversion.html>

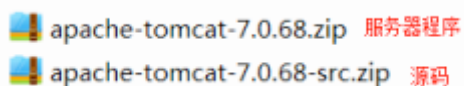
- o [zip](#) ([pgp](#), [md5](#), [sha1](#))
- o [tar.gz](#) ([pgp](#), [md5](#), [sha1](#))
- o [32-bit Windows zip](#) ([pgp](#), [md5](#), [sha1](#))
- o [64-bit Windows zip](#) ([pgp](#), [md5](#), [sha1](#))
- o [32-bit/64-bit Windows Service Installer](#) ([pgp](#), [md5](#), [sha1](#))

(2) 安装 Tomcat

Tomcat 有安装版和解压版（绿色版）

安装版以.exe 形式的安装包，双击安装到我们的电脑上，用的比较少

解压版，即绿色版，解压后直接使用，用的比较多



4. Tomcat 的目录结构



bin：脚本目录

启动脚本：startup.bat

停止脚本：shutdown.bat



conf : 配置文件目录 (config /configuration)

核心配置文件 : server.xml

用户权限配置文件 : tomcat-users.xml

所有 web 项目默认配置文件 : web.xml

lib : 依赖库, tomcat 和 web 项目中需要使用的 jar 包

logs : 日志文件.

localhost_access_log.*.txt tomcat 记录用户访问信息, 星*表示时间。

例如 : localhost_access_log.2016-02-28.txt

temp : 临时文件目录, 文件夹内内容可以任意删除。

webapps : 默认情况下发布 WEB 项目所存放的目录。

work : tomcat 处理 JSP 的工作目录。

5. Tomcate 的启动与运行

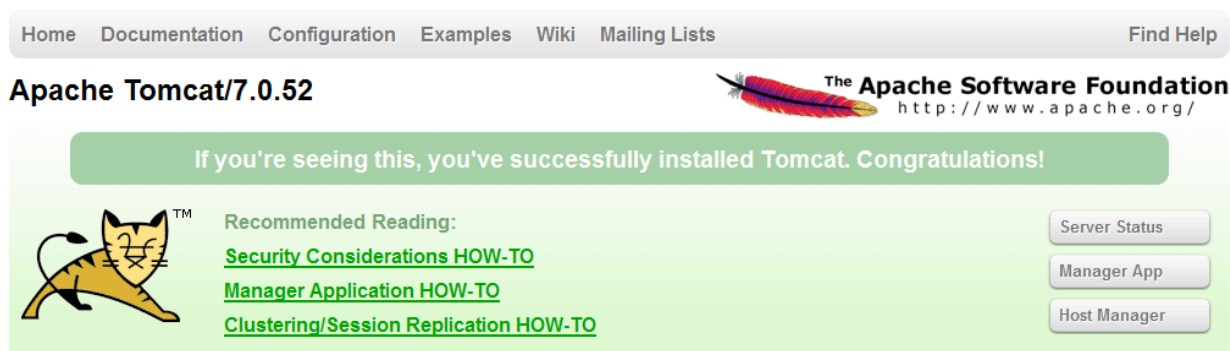
双击 Tomcat 下的 bin 下的 startup.bat 启动 Tomcat

startup.bat

```
Tomcat
信息: Deploying web application directory C:\Users\haohao\Desktop\apache-tomcat-7.0.52\webapps\docs
六月 30, 2016 9:25:53 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory C:\Users\haohao\Desktop\apache-tomcat-7.0.52\webapps\examples
六月 30, 2016 9:25:53 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory C:\Users\haohao\Desktop\apache-tomcat-7.0.52\webapps\host-manager
六月 30, 2016 9:25:53 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory C:\Users\haohao\Desktop\apache-tomcat-7.0.52\webapps\manager
六月 30, 2016 9:25:53 下午 org.apache.catalina.startup.HostConfig deployDirectory
信息: Deploying web application directory C:\Users\haohao\Desktop\apache-tomcat-7.0.52\webapps\ROOT
六月 30, 2016 9:25:53 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["http-bio-8080"]
六月 30, 2016 9:25:53 下午 org.apache.coyote.AbstractProtocol start
信息: Starting ProtocolHandler ["ajp-bio-8009"]
六月 30, 2016 9:25:53 下午 org.apache.catalina.startup.Catalina start
信息: Server startup in 1348 ms
```



在浏览器的地址栏中输入 `http://localhost:8080`,看到如下页面证明启动成功



Tomcat 启动不成功的原因分析：

1) 如果没有配置 `JAVA_HOME` 环境变量，在双击 “startup.bat” 文件运行 tomcat 时，将一闪立即关闭。且必须配置正确，及 `JAVA_HOME` 指向 JDK 的安装目录

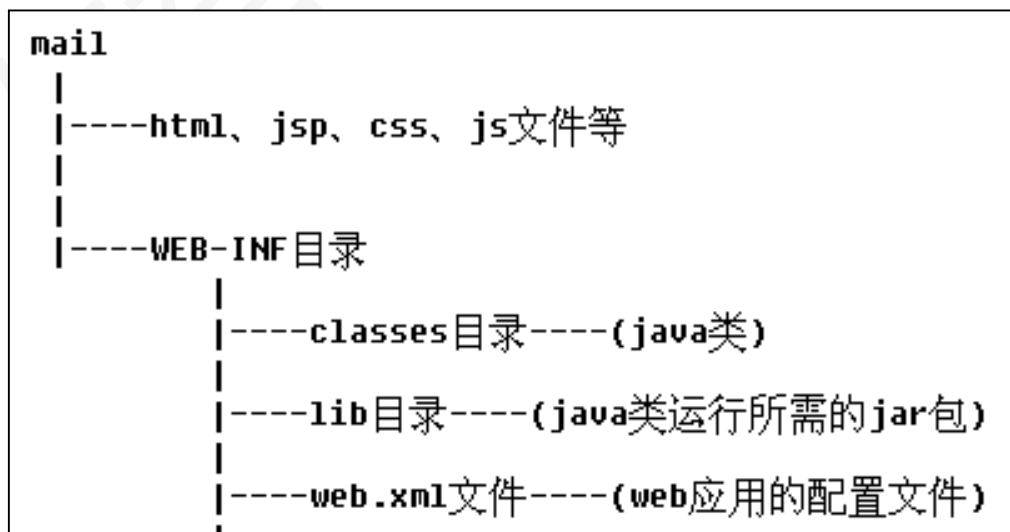
2) 端口冲突

`java.net.BindException: Address already in use: JVM_Bind <null>:8080`

修改 Tomcat/conf/server.xml

```
-->
<Connector port="9999" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" />
<!-- A "Connector" using the shared thread pool-->
```

6. Web 应用的目录结构

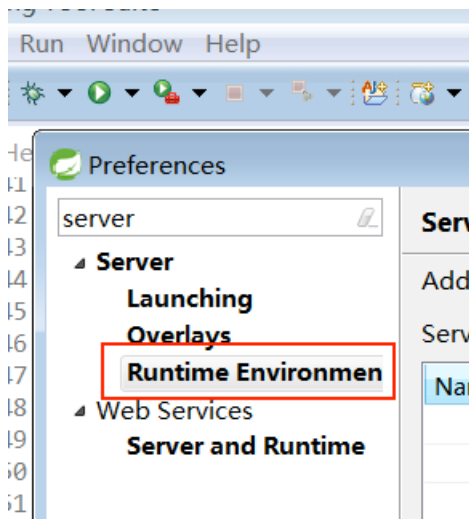




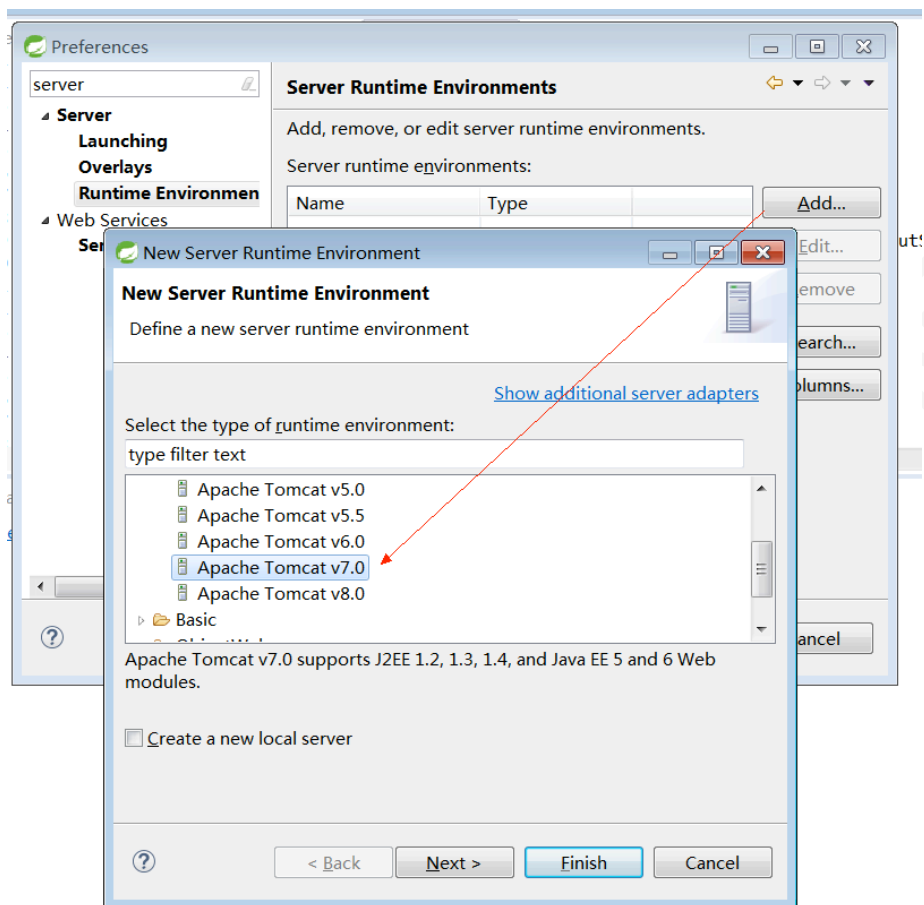
注意：WEB-INF 目录是受保护的，外界不能直接访问

7. 使用 Eclipse 绑定 Tomcat 并发布应用

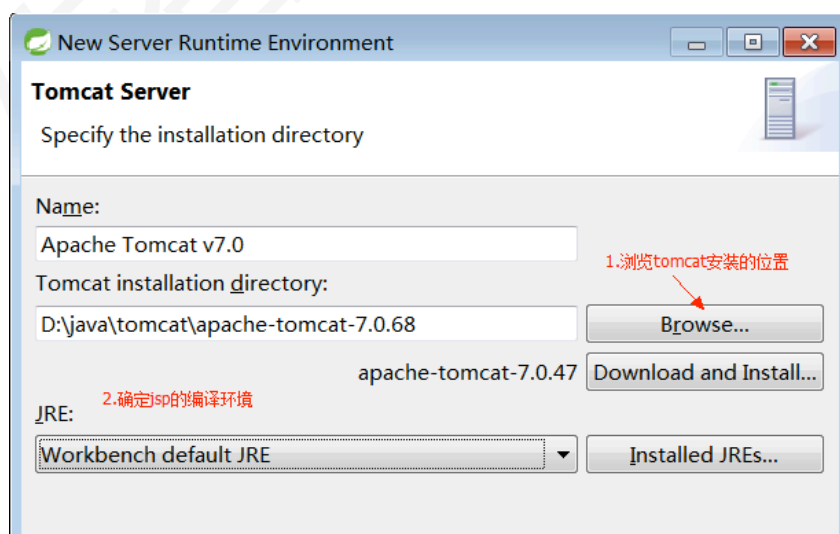
✦ 步骤 1 :获得服务器运行环境配置 ,Window/Preferences/Server/Runtime Environmen



步骤 2：添加服务器



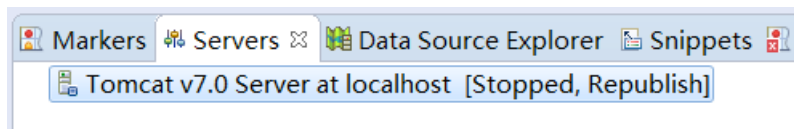
步骤 3：选择服务器在硬盘的地址，然后所有的都是确定/Next/Finish



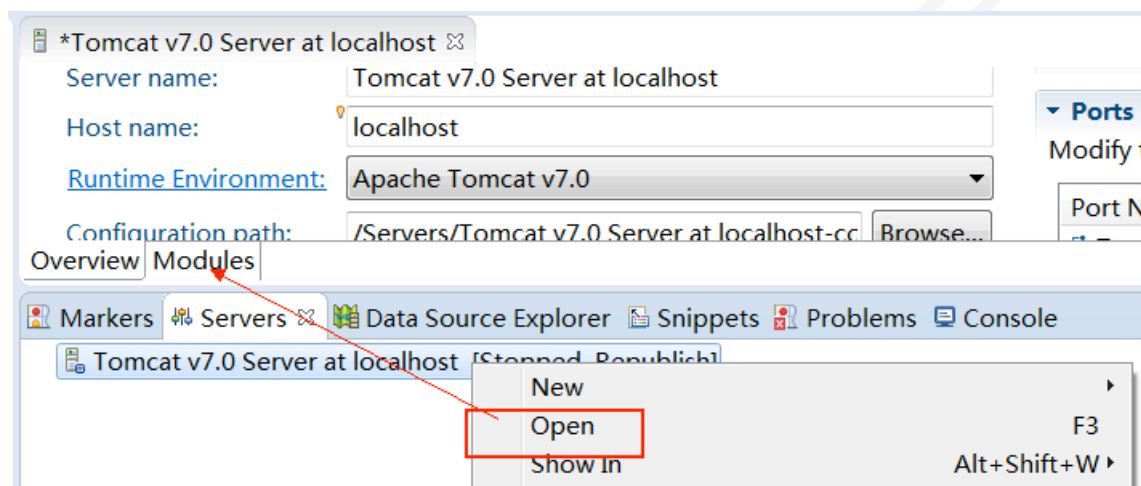


+

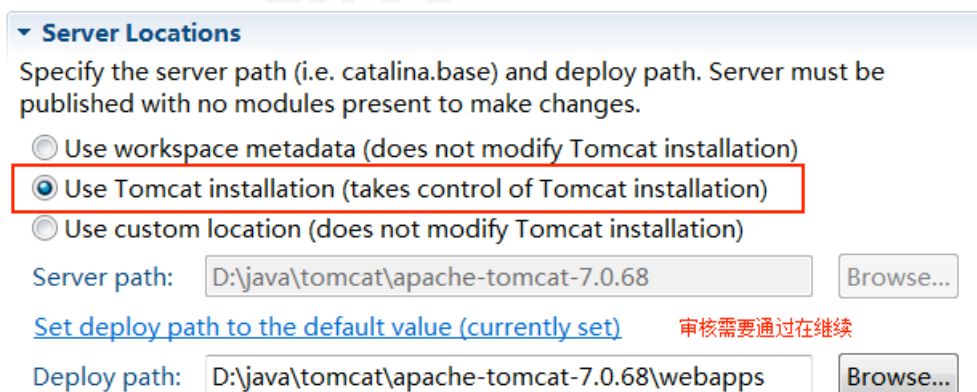
步骤 4：完成成功



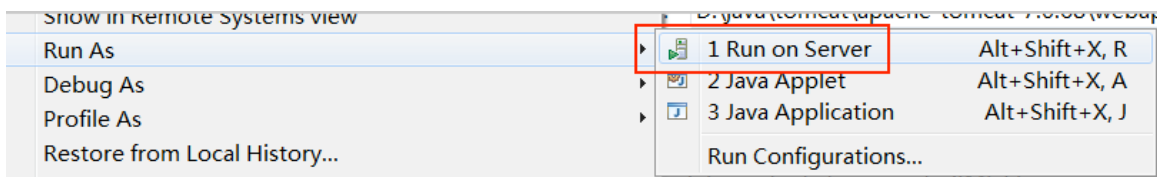
步骤 5：设置发布位置



修改 tomcat 发布的位置



步骤 6：项目右键/Run As/Run on Server





黑马程序员
www.itheima.com

传智播客旗下
高端IT教育品牌

改变中国IT教育，我们正在行动

传智播客Java学院