# WEB环境搭建

## 1.Web服务器

* Web服务器主要用来接收客户端发送的请求和响应客户端请求。
* 常见的JavaWeb服务器：
* **Tomcat（Apache）**：当前应用最广的JavaWeb服务器；
* JBoss（Redhat红帽）：支持JavaEE，应用比较广EJB容器 –> SSH轻量级的框架代替
* GlassFish（Orcale）：Oracle开发JavaWeb服务器，应用不是很广；
* Resin（Caucho）：支持JavaEE，应用越来越广；
* Weblogic（Orcale）：要钱的！支持JavaEE，适合大型项目；
* Websphere（IBM）：要钱的！支持JavaEE，适合大型项目；

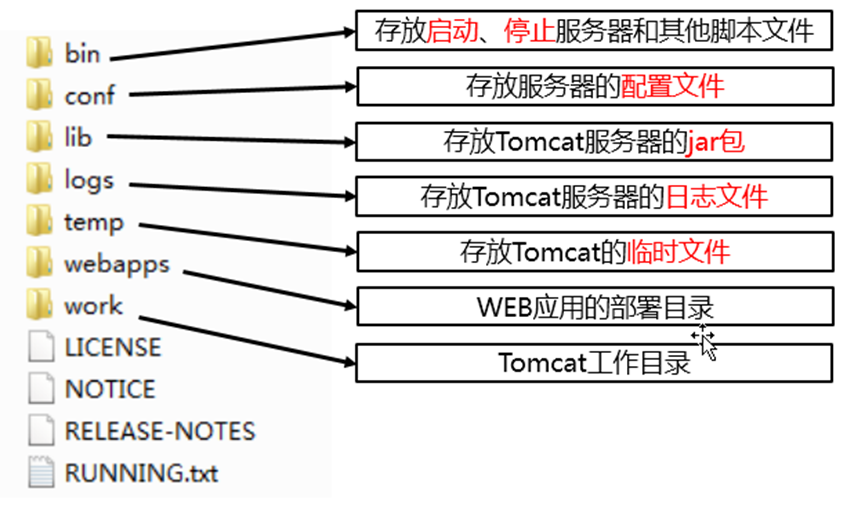
## 2.Tomcat服务器

### 2.1 Tomcat简介

* Tomcat是Apache 软件基金会（Apache Software Foundation）的Jakarta 项目中的一个核心项目，由Apache、Sun 和其他一些公司及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，因为Tomcat 技术先进、性能稳定，而且免费，因而深受Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的Web 应用服务器。目前最新版本是9.0。我们要用的是7.0。

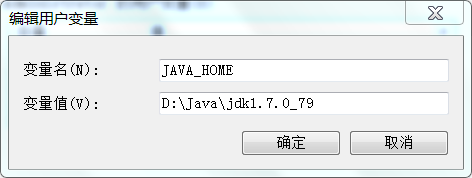
### 2.2 Tomcat安装、配置、启动

①解压apache-tomcat-xxx-windows-x64.zip到**非中文无空格**目录中



* bin：该目录下存放的是二进制可执行文件，如果是安装版，那么这个目录下会有两个exe文件：tomcat6.exe、tomcat6w.exe，前者是在控制台下启动Tomcat，后者是弹出GUI窗口启动Tomcat；如果是解压版，那么会有startup.bat和shutdown.bat文件，startup.bat用来启动Tomcat，但需要先配置JAVA\_HOME环境变量才能启动，shutdawn.bat用来停止Tomcat；
* conf：这是一个非常非常重要的目录，这个目录下有四个最为重要的文件：
* server.xml：配置整个服务器信息。例如修改端口号
* tomcat-users.xml：存储tomcat用户的文件，这里保存的是tomcat的用户名及密码，以及用户的角色信息。可以按着该文件中的注释信息添加tomcat用户，然后就可以在Tomcat主页中进入Tomcat Manager页面了；
* web.xml：部署描述符文件，这个文件中注册了很多MIME类型，即文档类型。这些MIME类型是客户端与服务器之间说明文档类型的，如用户请求一个html网页，那么服务器还会告诉客户端浏览器响应的文档是text/html类型的，这就是一个MIME类型。客户端浏览器通过这个MIME类型就知道如何处理它了。当然是在浏览器中显示这个html文件了。但如果服务器响应的是一个exe文件，那么浏览器就不可能显示它，而是应该弹出下载窗口才对。MIME就是用来说明文档的内容是什么类型的！
* context.xml：对所有应用的统一配置，通常我们不会去配置它。
* lib：Tomcat的类库，里面是一大堆jar文件。如果需要添加Tomcat依赖的jar文件，可以把它放到这个目录中，当然也可以把应用依赖的jar文件放到这个目录中，这个目录中的jar所有项目都可以共享之，但这样你的应用放到其他Tomcat下时就不能再共享这个目录下的jar包了，所以建议只把Tomcat需要的jar包放到这个目录下；
* logs：这个目录中都是日志文件，记录了Tomcat启动和关闭的信息，如果启动Tomcat时有错误，那么异常也会记录在日志文件中。
* temp：存放Tomcat的临时文件，这个目录下的东西可以在停止Tomcat后删除！
* webapps：存放web项目的目录，其中每个文件夹都是一个项目；如果这个目录下已经存在了目录，那么都是tomcat自带的项目。其中ROOT是一个特殊的项目，在地址栏中没有给出项目目录时，对应的就是ROOT项目。<http://localhost:8080/examples，进入示例项目。其中examples>就是项目名，即文件夹的名字。
* work：运行时生成的文件，最终运行的文件都在这里。通过webapps中的项目生成的！可以把这个目录下的内容删除，再次运行时会生再次生成work目录。当客户端用户访问一个JSP文件时，Tomcat会通过JSP生成Java文件，然后再编译Java文件生成class文件，生成的java和class文件都会存放到这个目录下。
* LICENSE：许可证。
* NOTICE：说明文件。

②如果双击startup.bat后窗口一闪而过，请查看JAVA\_HOME是否配置正确



③新建环境变量CATALINA\_HOME=解压目录（选配）

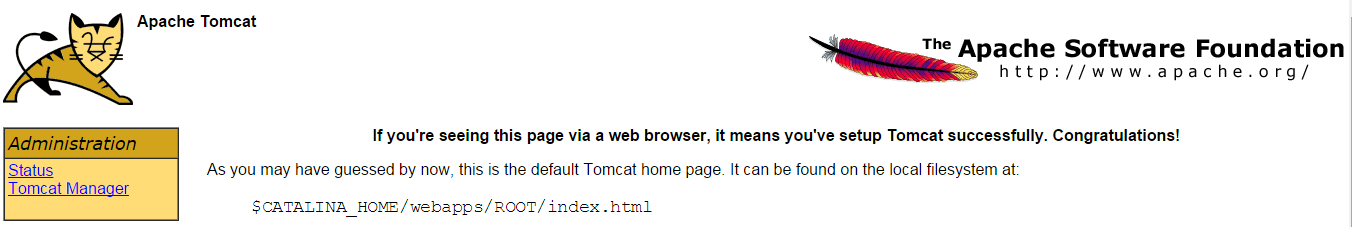


④在Path环境变量中加入Tomcat解压目录\bin目录



⑤在命令行中运行catalina run或者 startup启动Tomcat服务器，在浏览器地址栏访问如下地址进行测试

**http://localhost:8080**



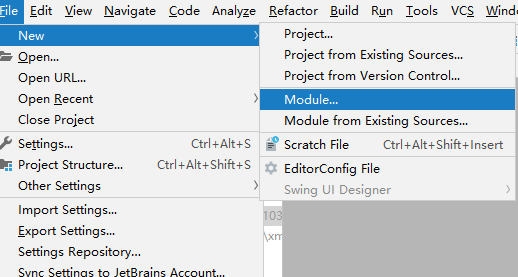
⑥如果启动失败，提示端口号被占用，则将默认的8080端口修改为其他未使用的值，例如10086等。

打开：解压目录\conf\server.xml，找到第一个Connector标签，修改port属性

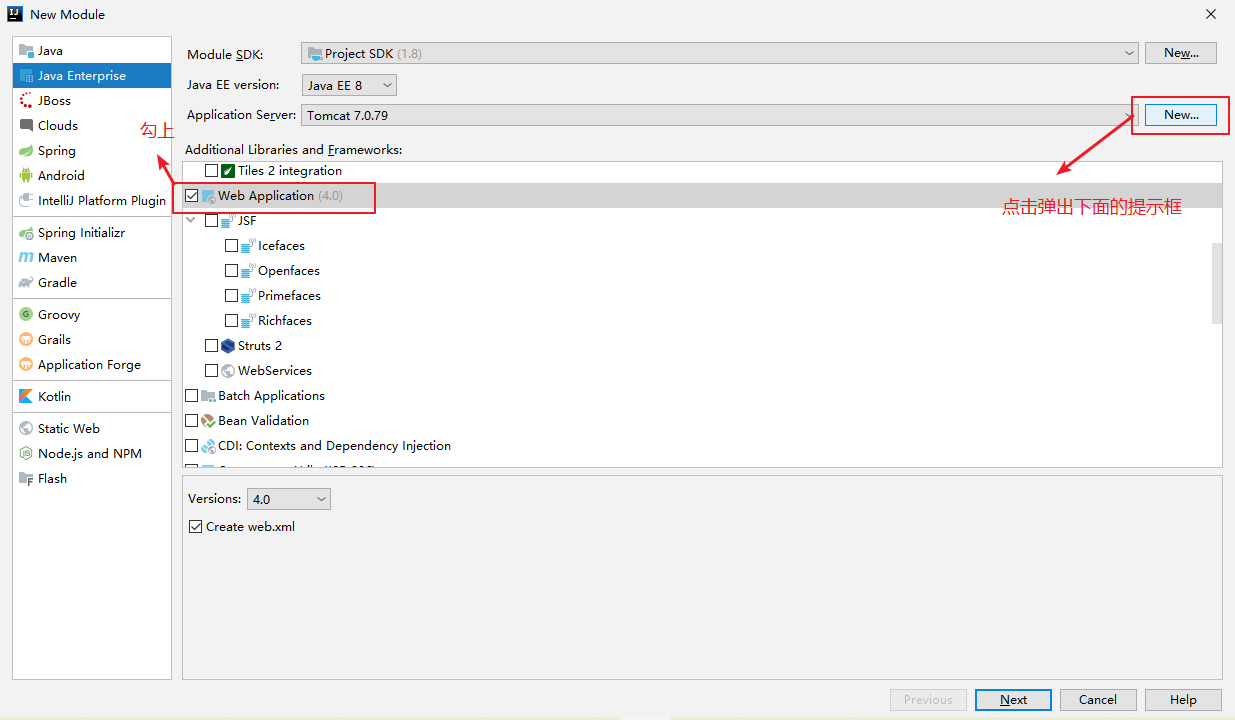
## 3.在Idea中配置Tomcat服务器

### 3.1 2019版Idea

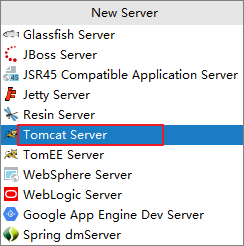
1. 点击File→New→Module



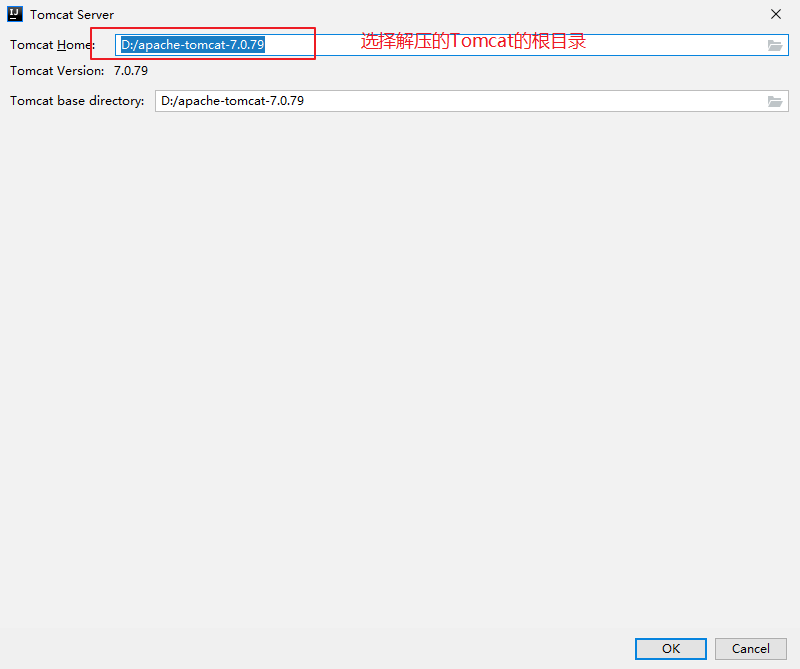
1. 选择Java EnterPrise，选中Web Application，点击New配置Tomcat



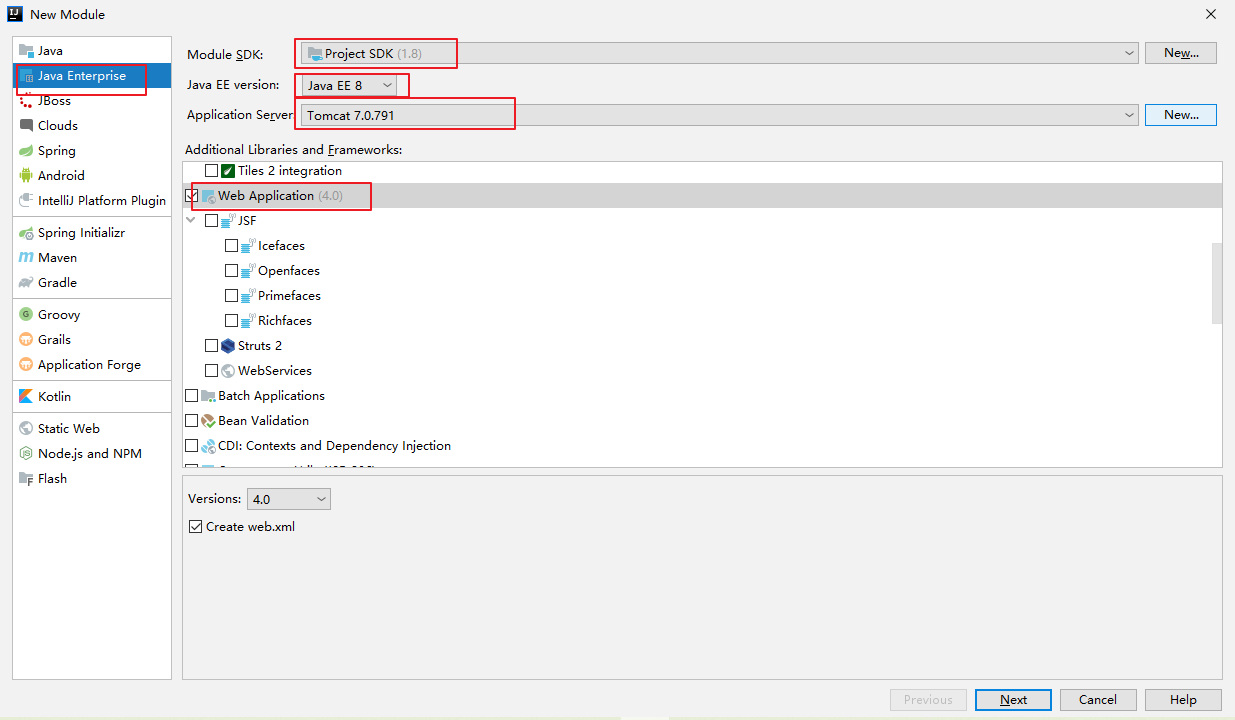
1. 点击New之后弹出的提示框中选择Tomcat Server



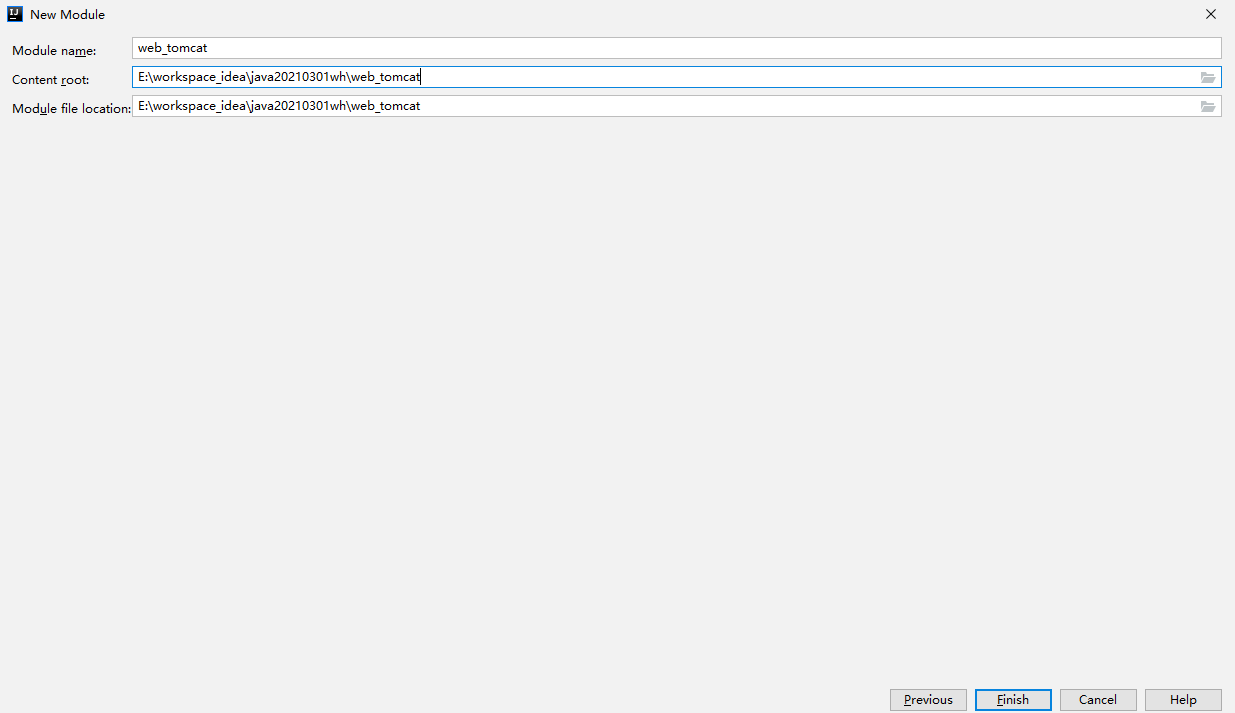
1. 选择解压的Tomcat的根目录



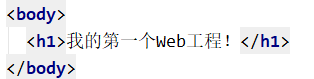
1. 点击Next



1. 设置动态Web工程的工程名后点击Finish



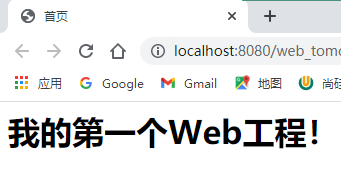
1. 打开web目录下的index.jsp页面在doby标签中添加标题



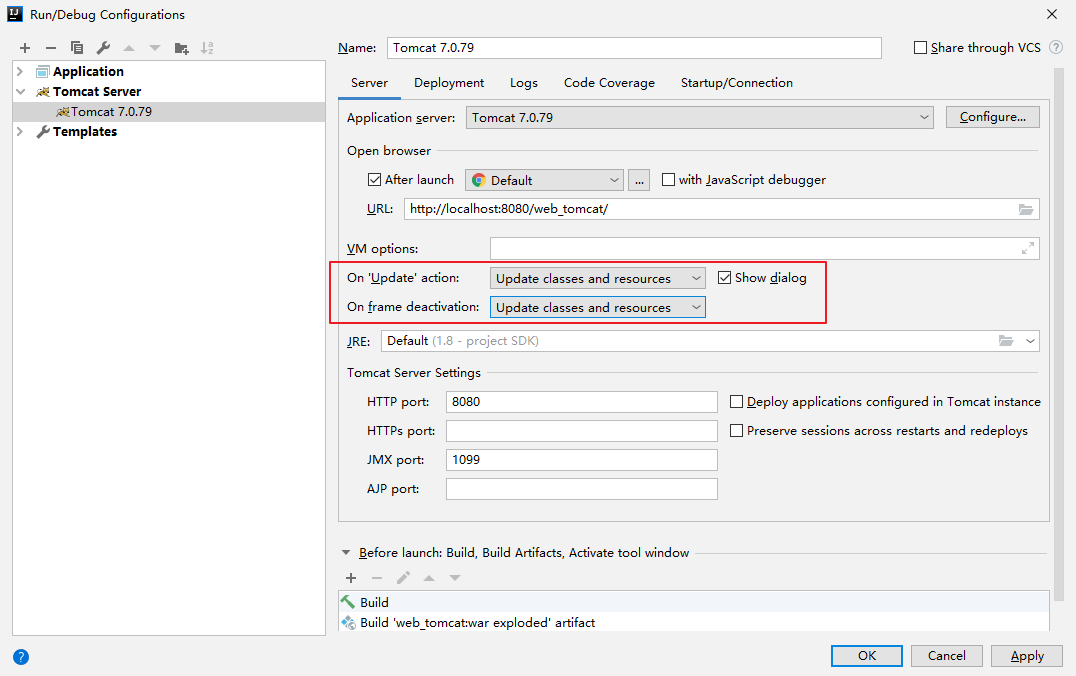
1. 点击按钮启动Tomcat



1. 自动打开浏览器，显示index.jsp页面中的内容

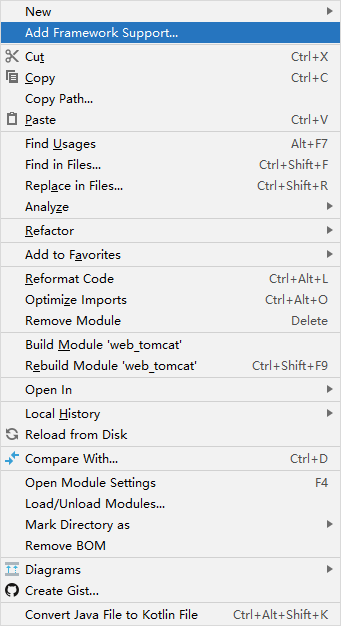


1. 如果每次修改了页面中的内容想让Tomcat自动更新，需要在Tomcat中做如下配置

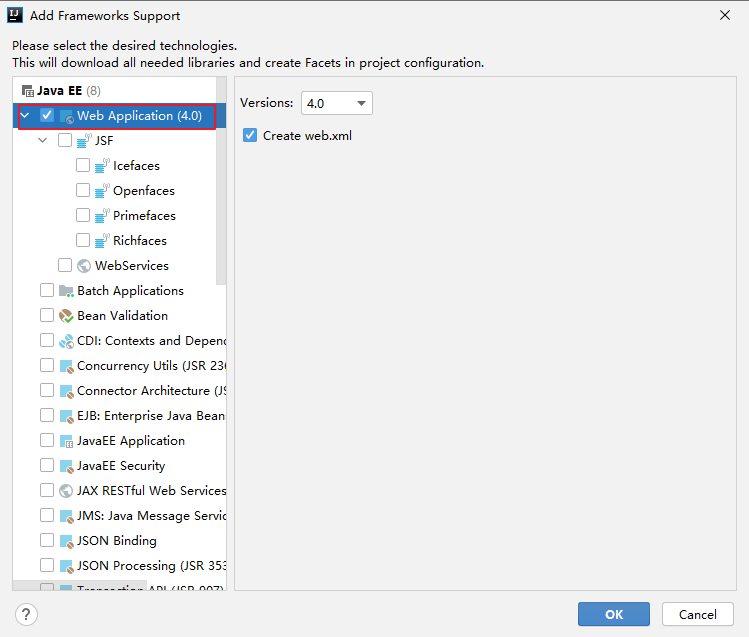


### 3.2 2020版Idea及以后的版本

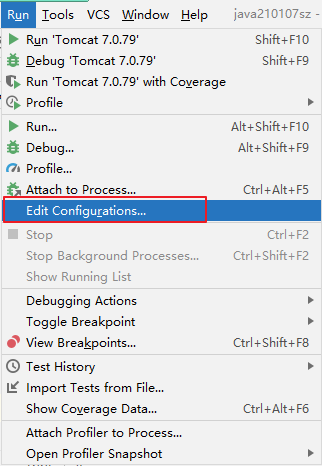
1. 先创建一个Java工程
2. 然后在工程名上右键→Add FrameWork Support…



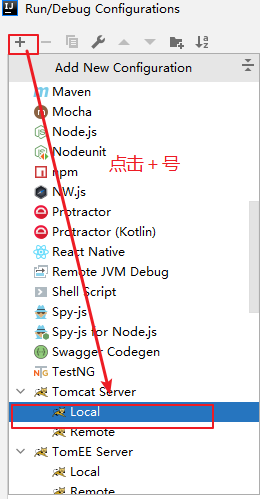
1. 选中Web Application(4.0)



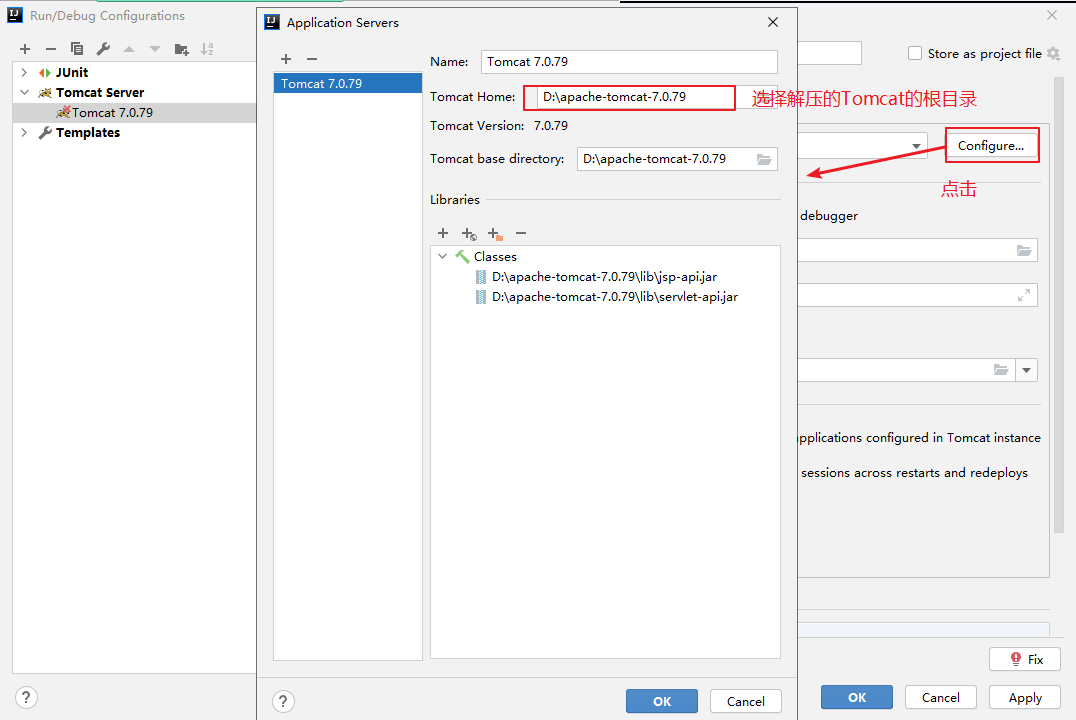
1. 选择Run→Edit Configurations…

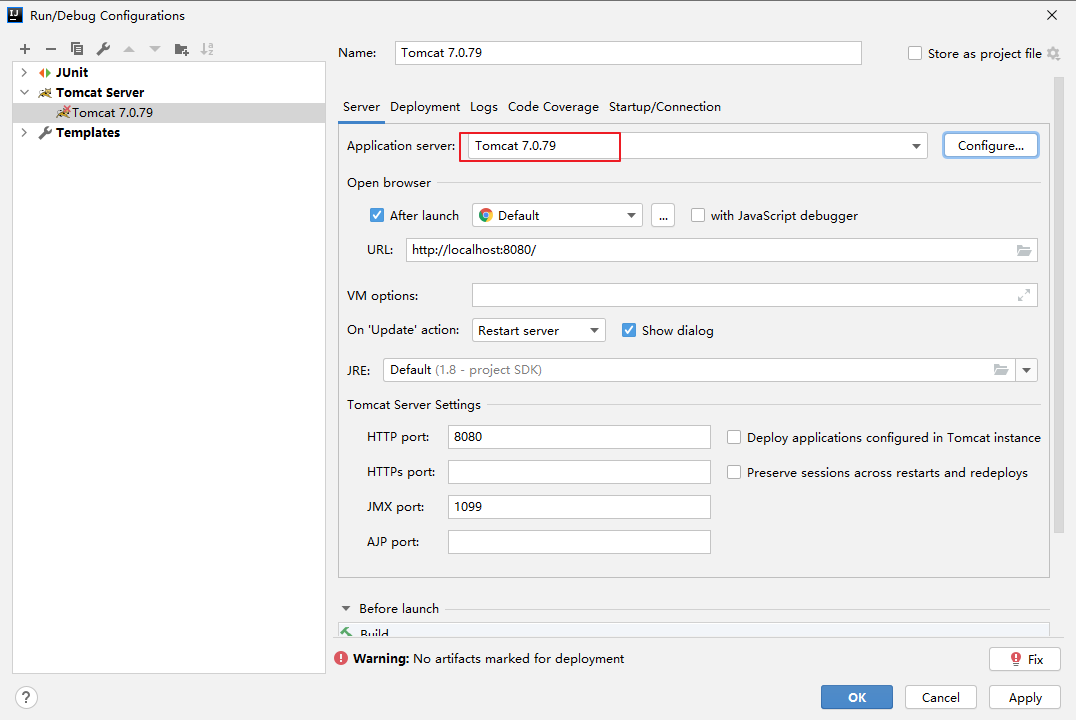


1. 点击＋号选择Tomcat Server→Local

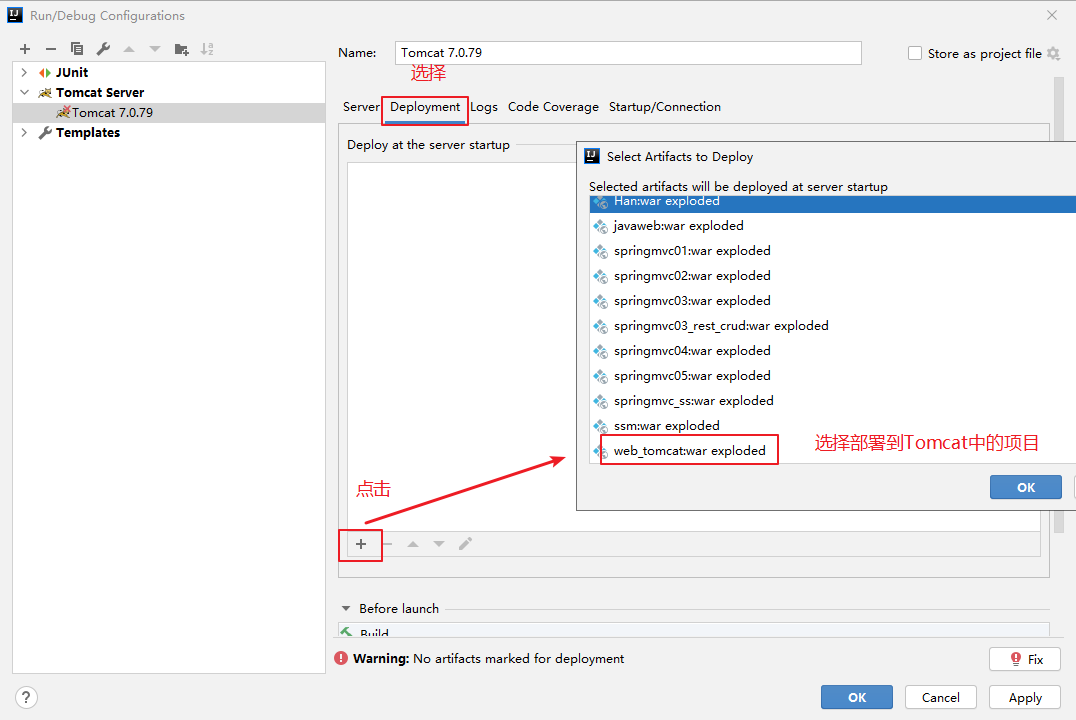


1. 点击Configure…在弹出的提示框中选择解压的Tomcat的根目录

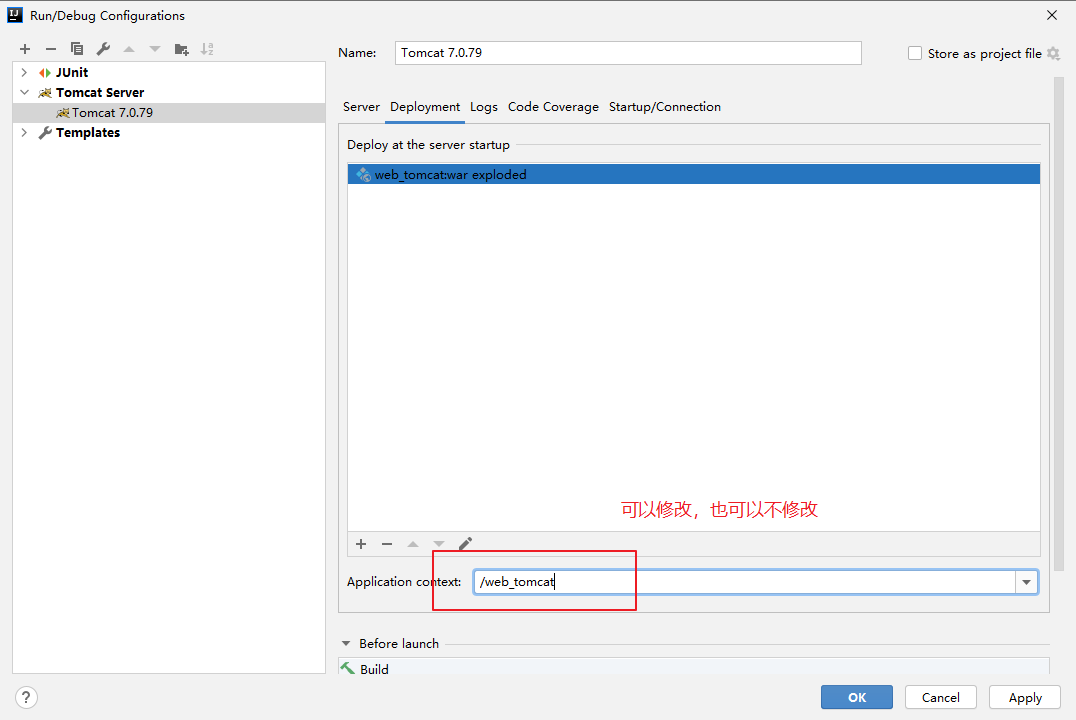




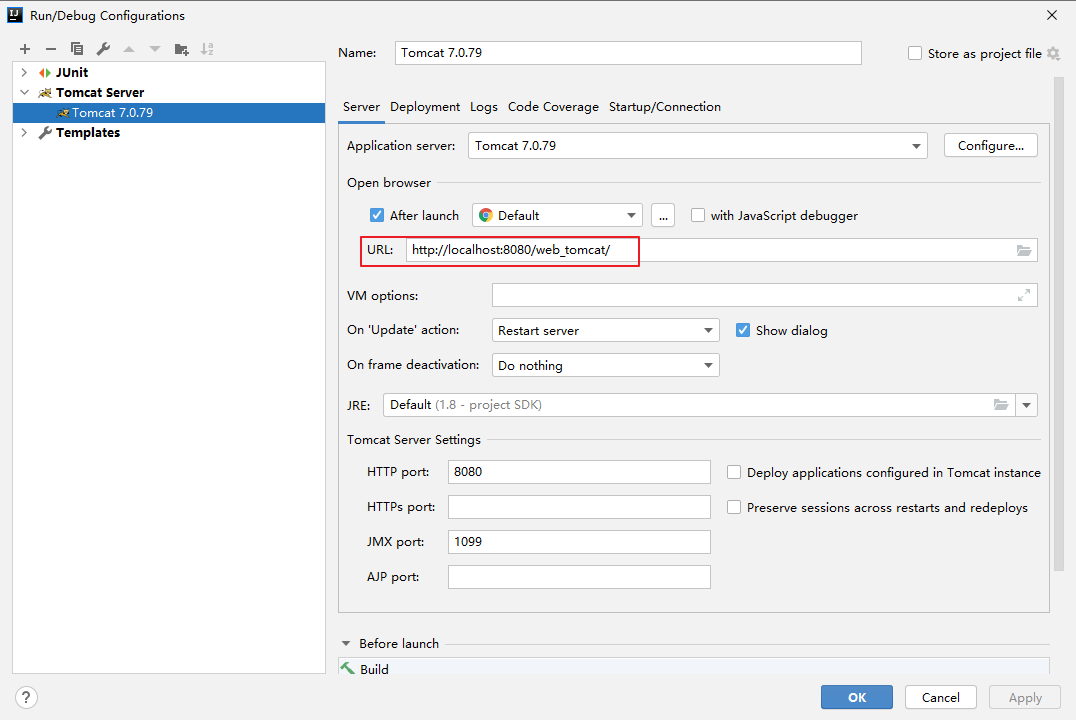
1. 点击Deployment，点击＋，选择部署到Tomcat中的动态Web工程



1. 可以通过修改Application Context修改在Tomcat服务器中部署的项目名



1. 修改了Application Context之后Server中的URL也会自动变化，点击OK即可



1. 编辑web目录下的index.jsp页面，点击启动Tomcat的按钮即可启动Tomcat

