

Tomcat服务器&servlet入门程序

第1章 tomcat服务器

1.1 软件的架构

1.1.1 网络应用程序（软件）的组成

网络中有很多的计算机，它们直接的信息交流，我们称之为：交互。在互联网交互的过程的有两个非常典型的交互方式——B/S 交互模型和C/S 交互模型。

什么是B/S 交互模型？

答：就是浏览器和服务器交互模型。

什么是C/S 交互模型？

答：就是客户端（例如：百度网盘）和服务器交互模型。

B/S 和C/S交互模型相同点和不同点：

相同点：

1. 都是基于请求-响应交互模型，即：
浏览器（客户端）向服务器发送一个请求。
服务器向浏览器（客户端）回送一个响应。
2. 必须先有请求再有响应
3. 请求和响应成对出现

不同点：

1. 实现C/S模型需要用户在自己的操作系统安装各种客户端软件（百度网盘、腾讯QQ等）；实现B/S模型，只需要用户在操作系统中安装浏览器即可。

注：B/S模型可以理解作为一种特殊C/S模型。

1.2 web资源的类别

1.2.1 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、音频、视频。

1.2.2 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：你在不同时间搜索微博的热门话题内容是不一样的，每天的天气情况也是变化的。这些数据由程序生成，JSP/Servlet、ASP、PHP等技术都可以完成。

总结：

静态资源的数据都是写死在页面上的固定不变。

动态资源浏览器访问的结果是变化的（动态web资源你的数据都是程序读取数据库、xml等文件生成的数据）。

接下来主要学习动态web资源。

1.3 服务器的概念

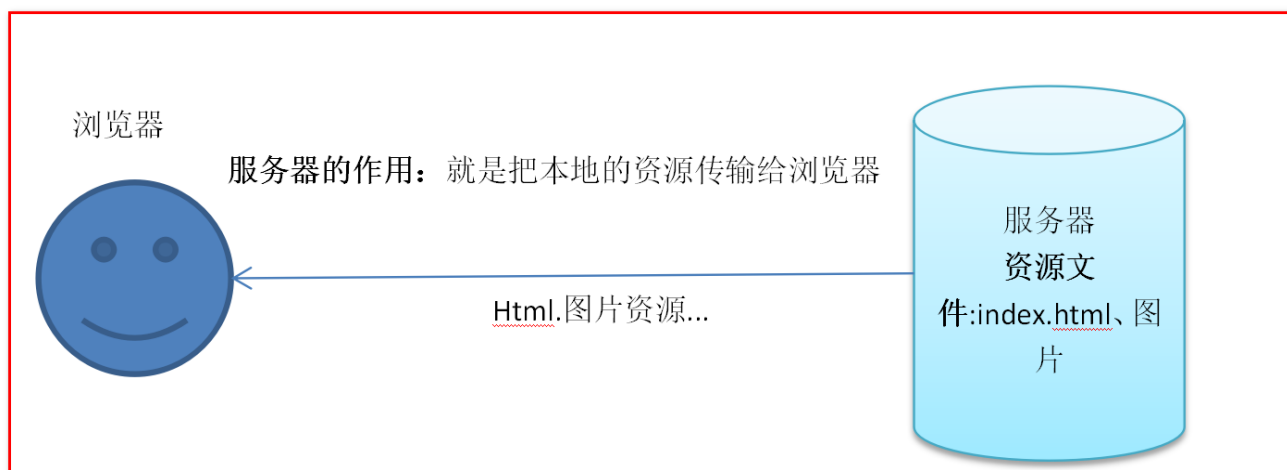
1.3.1 什么是服务器

服务器就是一个软件，任何电脑只需要安装上了服务器软件，然后该电脑的指定目录下的资源就能提供对外访问的资源。

1.4 服务器的作用

1.4.1 服务器的作用

提供计算服务的设备，服务类型有很多，常见的有：游戏服务，购物服务，新闻服务等。



1.5 常见的服务器软件

1.5.1 常见的服务器软件介绍

1. WebLogic

Oracle公司的产品，是目前应用最广泛的Web服务器，支持J2EE规范。WebLogic是用于开发、集成、部署和管理大型分布式Web应用、网络应用和数据库应用的Java应用服务器。

2. WebSphere

IBM公司的WebSphere，支持JavaEE规范。WebSphere 是按需应变的电子商务时代的最主要的软件平台，可用于企业开发、部署和整合新一代的电子商务应用。

3. Glass Fish

最早是Sun公司的产品，后来被Oracle收购，开源，中型服务器。

4. JBoss

JBoss公司产品，开源，支持JavaEE规范，占用内存、硬盘小，安全性和性能高。

5. Tomcat

中小型的应用系统，免费开源，支持JSP和Servlet。

注意：今天我们学习和使用的是tomcat服务器。

1.6 tomcat服务器软件安装和介绍

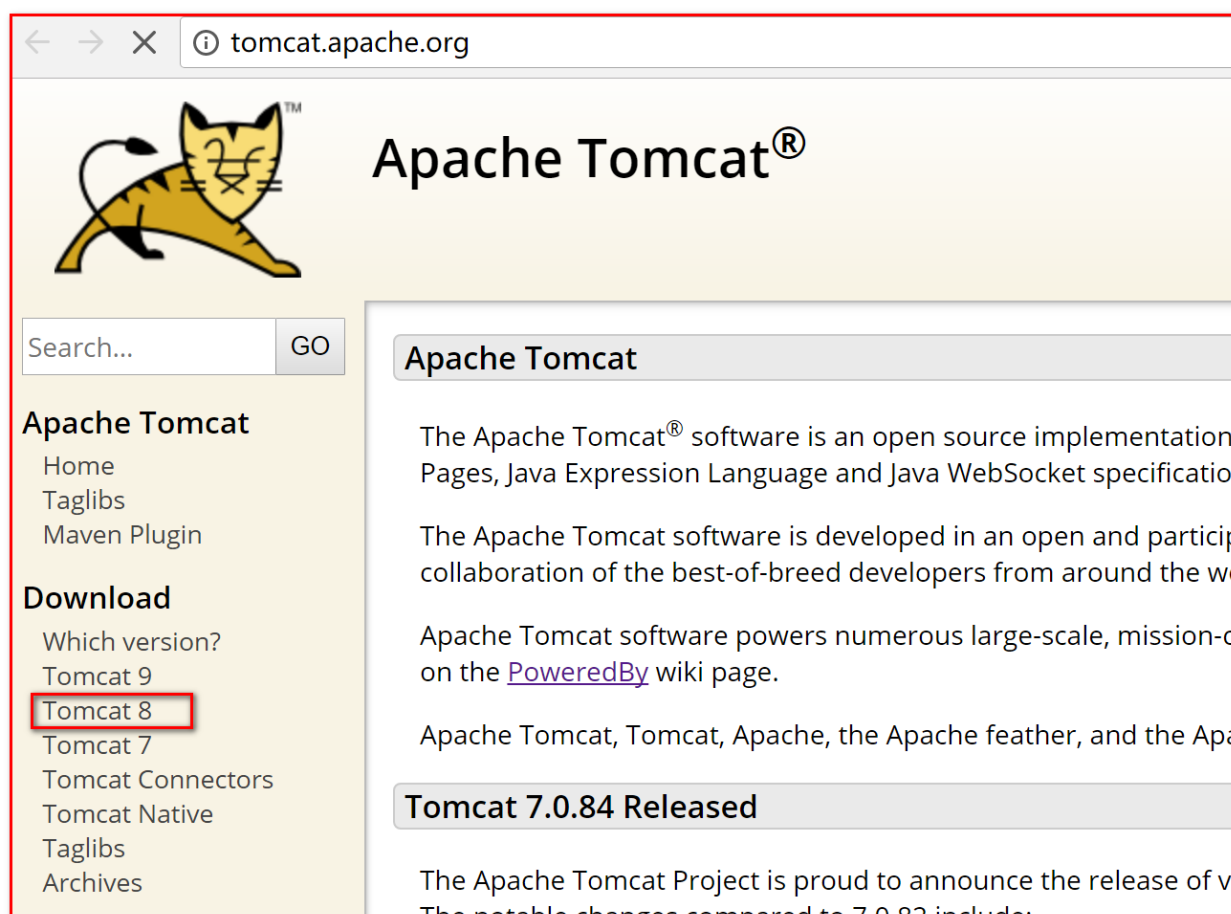
Tomcat基本概述：

Tomcat服务器是一个免费的开放源代码的Web应用服务器。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，是目前比较流行的Web应用服务器。

1.6.1 tomcat服务器软件下载

1. 先去官网下载：<http://tomcat.apache.org/>，选择tomcat8版本（红框所示）：



2. 选择要下载的文件（红框所示）：

8.5.27

Please see the [README](#) file for packaging information. It explains what every dist

Binary Distributions

- Core:
 - [zip](#) ([pgp](#), [md5](#), [sha1](#), [sha512](#))
 - [tar.gz](#) ([pgp](#), [md5](#), [sha1](#), [sha512](#))
 - [32-bit Windows zip](#) ([pgp](#), [md5](#), [sha1](#), [sha512](#))
 - [64-bit Windows zip](#) ([pgp](#), [md5](#), [sha1](#), [sha512](#))
 - [32-bit/64-bit Windows Service Installer](#) ([pgp](#), [md5](#), [sha1](#), [sha512](#))

tar.gz 文件 是linux操作系统下的安装版本

exe文件是window操作系统下的安装版本

zip文件是window操作系统下压缩版本（我们选择zip文件）

3. 下载完成：



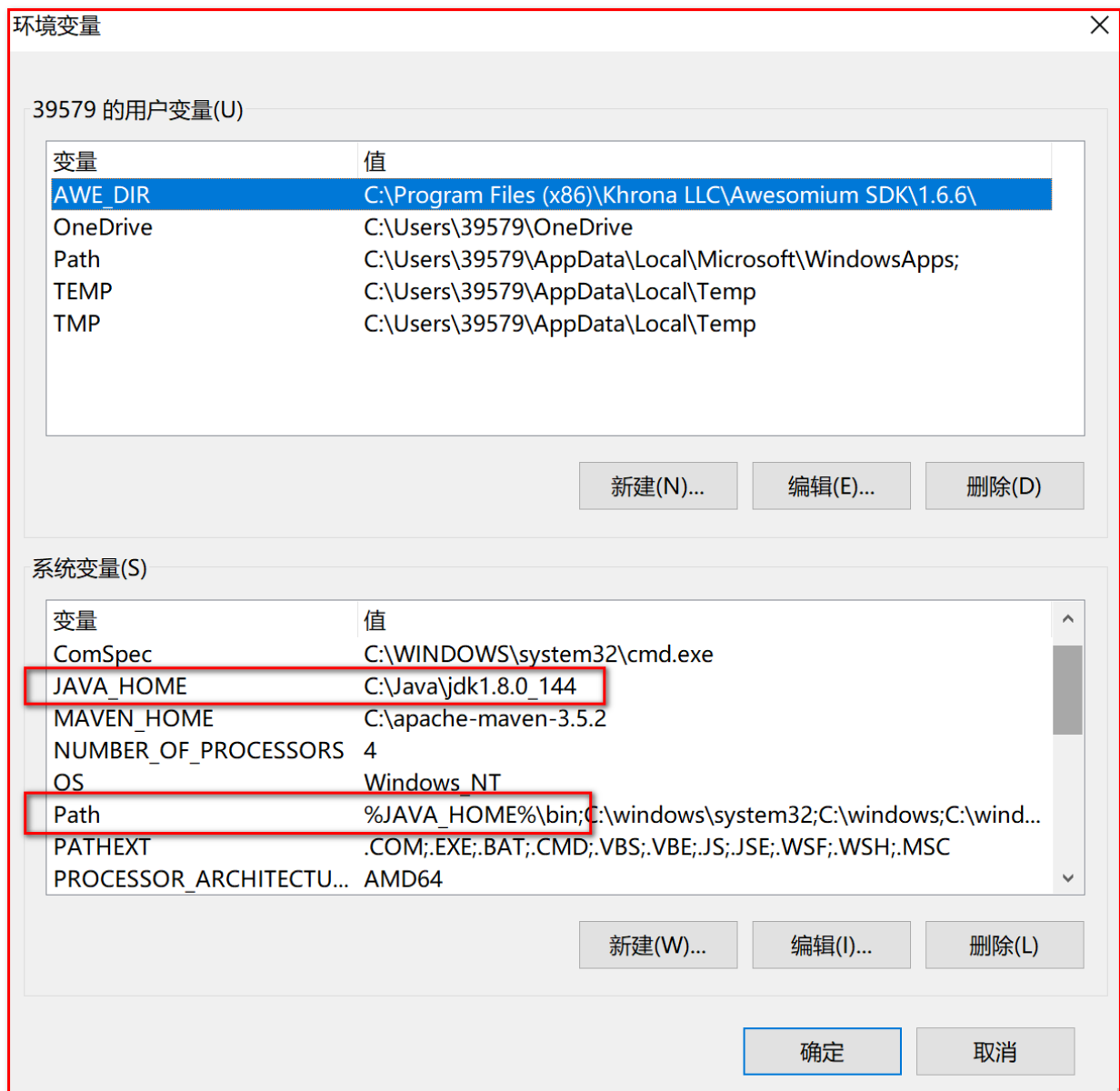
apache-tomcat-8.5.27-windows-x64.zip

1.6.2 tomcat服务器软件安装

1. 直接解压当前这个tomcat压缩包：

2. 配置环境变量：

tomcat运行依赖于java环境：






























1.6.3 启动与关闭tomcat服务器

1. 启动tomcat服务器

查找tomcat目录下bin目录，查找其中的startup.bat命令，双击启动服务器：

本地磁盘 (C:) > soft > tomcat > apache-tomcat-8.5.27 > bin

名称	修改日期
 bootstrap.jar	2018/1/18 20:13
 catalina.bat	2018/1/18 20:13
 catalina.sh	2018/1/18 20:13
 catalina-tasks.xml	2018/1/18 20:13
 commons-daemon.jar	2018/1/18 20:13
 commons-daemon-native.tar.gz	2018/1/18 20:13
 configtest.bat	2018/1/18 20:13
 configtest.sh	2018/1/18 20:13
 daemon.sh	2018/1/18 20:13
 digest.bat	2018/1/18 20:13
 digest.sh	2018/1/18 20:13
 service.bat	2018/1/18 20:13
 setclasspath.bat	2018/1/18 20:13
 setclasspath.sh	2018/1/18 20:13
 shutdown.bat	2018/1/18 20:13
 shutdown.sh	2018/1/18 20:13
 startup.bat	2018/1/18 20:13
 startup.sh	2018/1/18 20:13
 tcnative-1.dll	2018/1/18 20:13
 tomcat8.exe	2018/1/18 20:13
 tomcat8w.exe	2018/1/18 20:13
 tomcat-juli.jar	2018/1/18 20:13
 tomcat-native.tar.gz	2018/1/18 20:13
 tool-wrapper.bat	2018/1/18 20:13
 tool-wrapper.sh	2018/1/18 20:13
 version.bat	2018/1/18 20:13
 version.sh	2018/1/18 20:13

tomcat启动命令

启动效果：

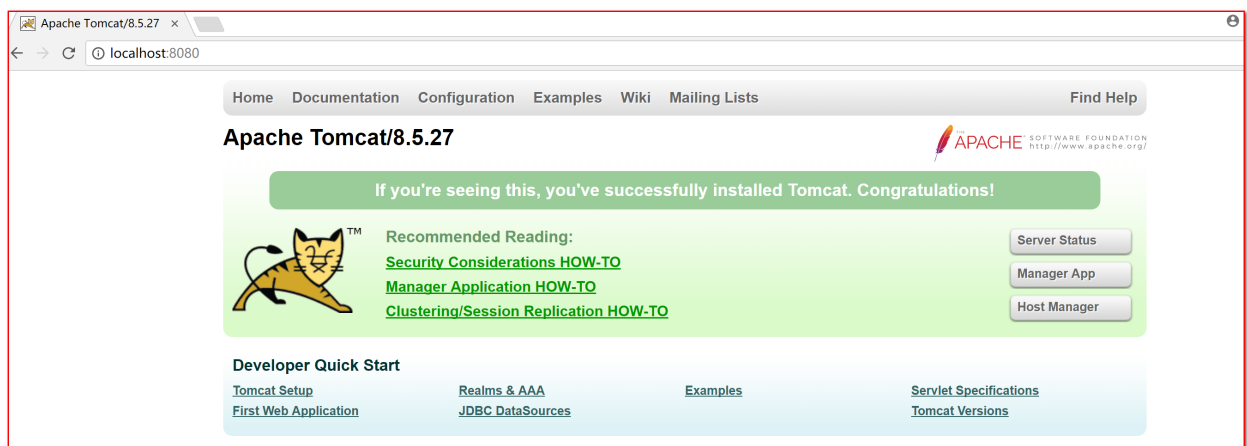
```
Tomcat
r for servlet write/read
26-Jan-2018 22:20:36.136 信息 [main] org.apache.catalina.startup.Catalina.load Initialization processed in 1263 ms
26-Jan-2018 22:20:36.163 信息 [main] org.apache.catalina.core.StandardService.startInternal Starting service [Catalina]
26-Jan-2018 22:20:36.164 信息 [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet Engine: Apache Tomcat/8.5.27
26-Jan-2018 22:20:36.187 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\docs]
26-Jan-2018 22:20:36.467 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\docs] has finished in [279] ms
26-Jan-2018 22:20:36.468 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\examples]
26-Jan-2018 22:20:36.793 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\examples] has finished in [325] ms
26-Jan-2018 22:20:36.793 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\host-manager]
26-Jan-2018 22:20:36.826 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\host-manager] has finished in [33] ms
26-Jan-2018 22:20:36.827 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\manager]
26-Jan-2018 22:20:36.858 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\manager] has finished in [31] ms
26-Jan-2018 22:20:36.858 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\ROOT]
26-Jan-2018 22:20:36.883 信息 [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application directory [C:\soft\tomcat\apache-tomcat-8.5.27\webapps\ROOT] has finished in [25] ms
26-Jan-2018 22:20:36.888 信息 [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
26-Jan-2018 22:20:36.900 信息 [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-nio-8009"]
26-Jan-2018 22:20:36.905 信息 [main] org.apache.catalina.startup.Catalina.start Server startup in 767 ms
```

2. 测试访问tomcat服务器

打开浏览器在，在浏览器的地址栏中输入：

<http://127.0.0.1:8080>

<http://localhost:8080>






























注：Localhost相当于127.0.0.1

3. 关闭tomcat服务器

查找tomcat目录下bin目录，查找其中的shutdown.bat命令，双击关闭服务器：

本地磁盘 (C:) > soft > tomcat > apache-tomcat-8.5.27 > bin

名称	修改日期
 bootstrap.jar	2018/1/18 20:13
 catalina.bat	2018/1/18 20:13
 catalina.sh	2018/1/18 20:13
 catalina-tasks.xml	2018/1/18 20:13
 commons-daemon.jar	2018/1/18 20:13
 commons-daemon-native.tar.gz	2018/1/18 20:13
 configtest.bat	2018/1/18 20:13
 configtest.sh	2018/1/18 20:13
 daemon.sh	2018/1/18 20:13
 digest.bat	2018/1/18 20:13
 digest.sh	2018/1/18 20:13
 service.bat	2018/1/18 20:13
 setclasspath.bat	2018/1/18 20:13
 setclasspath.sh	2018/1/18 20:13
 shutdown.bat	2018/1/18 20:13
 shutdown.sh	2018/1/18 20:13
 startup.bat	2018/1/18 20:13
 startup.sh	2018/1/18 20:13
 tcnative-1.dll	2018/1/18 20:13
 tomcat8.exe	2018/1/18 20:13
 tomcat8w.exe	2018/1/18 20:13
 tomcat-juli.jar	2018/1/18 20:13
 tomcat-native.tar.gz	2018/1/18 20:13
 tool-wrapper.bat	2018/1/18 20:13
 tool-wrapper.sh	2018/1/18 20:13
 version.bat	2018/1/18 20:13
 version.sh	2018/1/18 20:13

tomcat服务器关闭命令

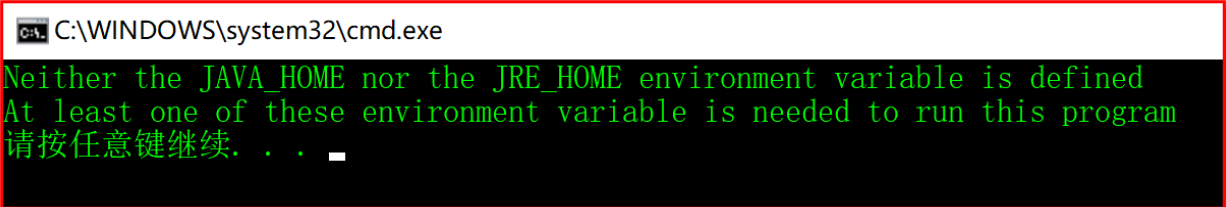
1.6.4 tomcat安装常见的问题

1. 无法启动（闪退：cmd命令窗口出现一下，就消失）：

主要原因：没有配置JAVA_HOME环境变量。JAVA_HOME 环境变量 中配置的是JDK的安装目录，不包含bin目录，不是tomcat的安装目录。

闪退的原因查看：可以在startup.bat文件末尾书写pause命令。让运行的窗口暂停。

效果：

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.exe'. The window has a black background with green text. The text reads: 'Neither the JAVA_HOME nor the JRE_HOME environment variable is defined. At least one of these environment variable is needed to run this program. 请按任意键继续. . .'.

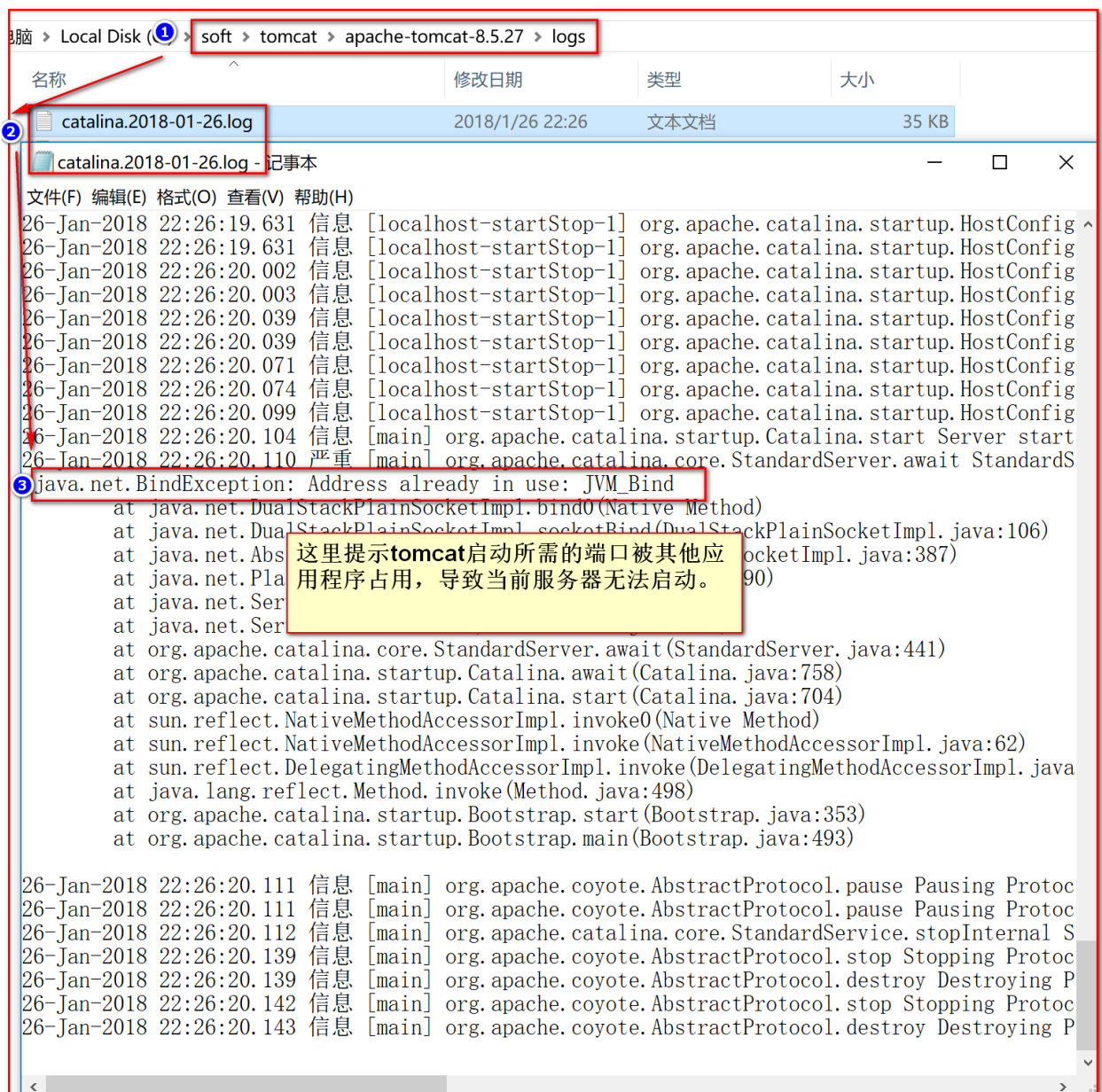
```
C:\WINDOWS\system32\cmd.exe
Neither the JAVA_HOME nor the JRE_HOME environment variable is defined
At least one of these environment variable is needed to run this program
请按任意键继续. . .
```

2. 端口被占用导致启动失败

如果启动的时候，发生异常问题，这有可能是端口被占用。

Tomcat服务器在启动的时候默认占用本地的8080端口，如果这个端口被占用，启动的时候就会报错。

报错内容可以通过查询tomcat目录下的logs目录中Catalina.当前系统年月日.log文件查看，如下图：



我们需要做的是查看本地端口使用情况，关闭占用端口的程序：在dos窗口中输入 netstat -nao 就可以查看当前端口的占用情况：

```

C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.16299.125]
(c) 2017 Microsoft Corporation。保留所有权利。

C:\Users\39579>netstat -nao

活动连接

协议 本地地址 外部地址 状态 PID
TCP 0.0.0.0:135 0.0.0.0:0 LISTENING 1040
TCP 0.0.0.0:445 0.0.0.0:0 LISTENING 4
TCP 0.0.0.0:902 0.0.0.0:0 LISTENING 4812
TCP 0.0.0.0:912 0.0.0.0:0 LISTENING 4812
TCP 0.0.0.0:3306 0.0.0.0:0 LISTENING 3380
TCP 0.0.0.0:5357 0.0.0.0:0 LISTENING 4
TCP 0.0.0.0:7680 0.0.0.0:0 LISTENING 13120
TCP 0.0.0.0:8009 0.0.0.0:0 LISTENING 11604
TCP 0.0.0.0:8080 0.0.0.0:0 LISTENING 11604
TCP 0.0.0.0:13231 0.0.0.0:0 LISTENING
TCP 0.0.0.0:49664 0.0.0.0:0 LISTENING
TCP 0.0.0.0:49665 0.0.0.0:0 LISTENING
TCP 0.0.0.0:49666 0.0.0.0:0 LISTENING
TCP 0.0.0.0:49670 0.0.0.0:0 LISTENING

```

当前应用程序的编号 (pid)

使用任务管理器---->选择详细信息---->查看pid为11604（查询出来的值），关闭结束进程即可：

任务管理器

文件(F) 选项(O) 查看(V)

进程

性能

应用历史记录

启动

用户

详细信息












名称	PID [^]	状态
WUDFHost.exe	11140	正在运行
NVIDIA Web Helper...	11168	正在运行
conhost.exe	11260	正在运行
dasHost.exe	11276	正在运行
java.exe	11604	正在运行
SearchUI.exe	11776	已暂停
svchost.exe	11816	正在运行

注意：如果这个进程是操作系统的任务进程，这时一般是不能停止这个进程。

如果是系统进程占用端口，那么我们只能换一个端口，下面看如何修改端口：

3. 修改tomcat启动端口

Tomcat服务器的配置文件，全部都在tomcat的安装目录下conf目录下：

脑 > Local Disk (C:) > soft > tomcat > apache-tomcat-8.5.27 > conf		
名称		修改日期
 Catalina		2018/1/26 22:20
 catalina.policy		2018/1/18 20:13
 catalina.properties		2018/1/18 20:13
 context.xml		2018/1/18 20:13
 jaspic-providers.xml		2018/1/18 20:13
 jaspic-providers.xsd		2018/1/18 20:13
 logging.properties		2018/1/18 20:13
 server.xml		2018/1/18 20:13
 tomcat-users.xml		2018/1/18 20:13
 tomcat-users.xsd		2018/1/18 20:13
 web.xml		2018/1/18 20:13

server.xml:服务器端口配置、服务器自身配置文件

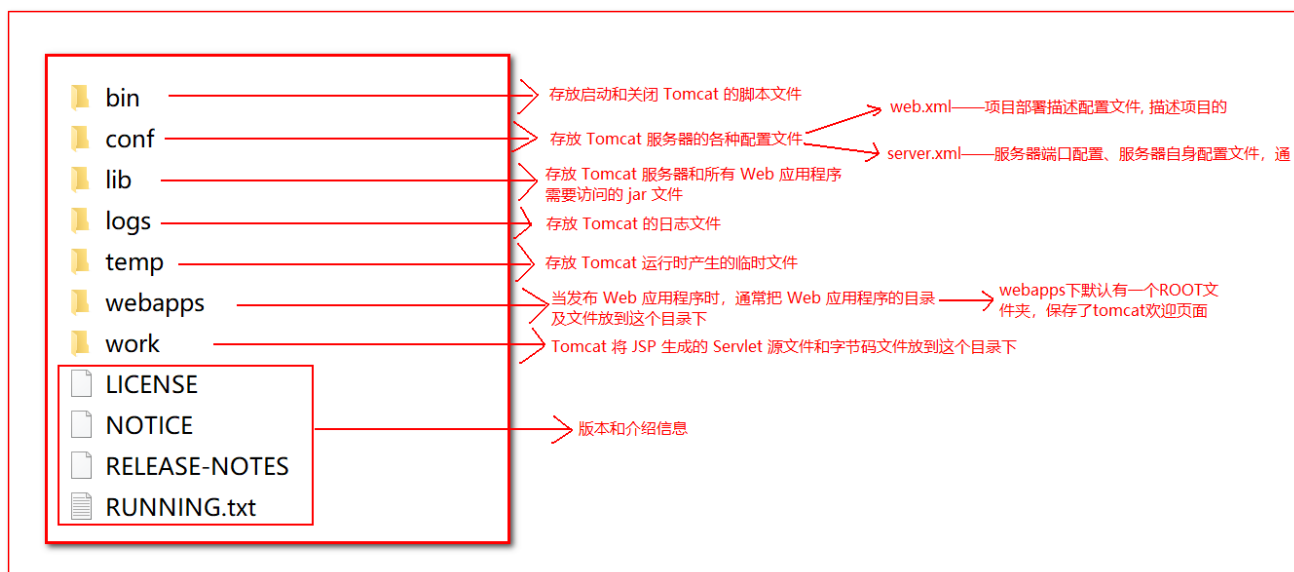
打开server.xml，修改服务器端口：

69	<Connector	port="8080"	protocol="HTTP/1.1"
70		connectionTimeou	默认端口
71		redirectPort="8443"	

修改tomcat的端口为9090：

修改完server.xml文件必须重启服务器才能有效。 通过浏览器的地址栏访问测试：<http://localhost:9090>

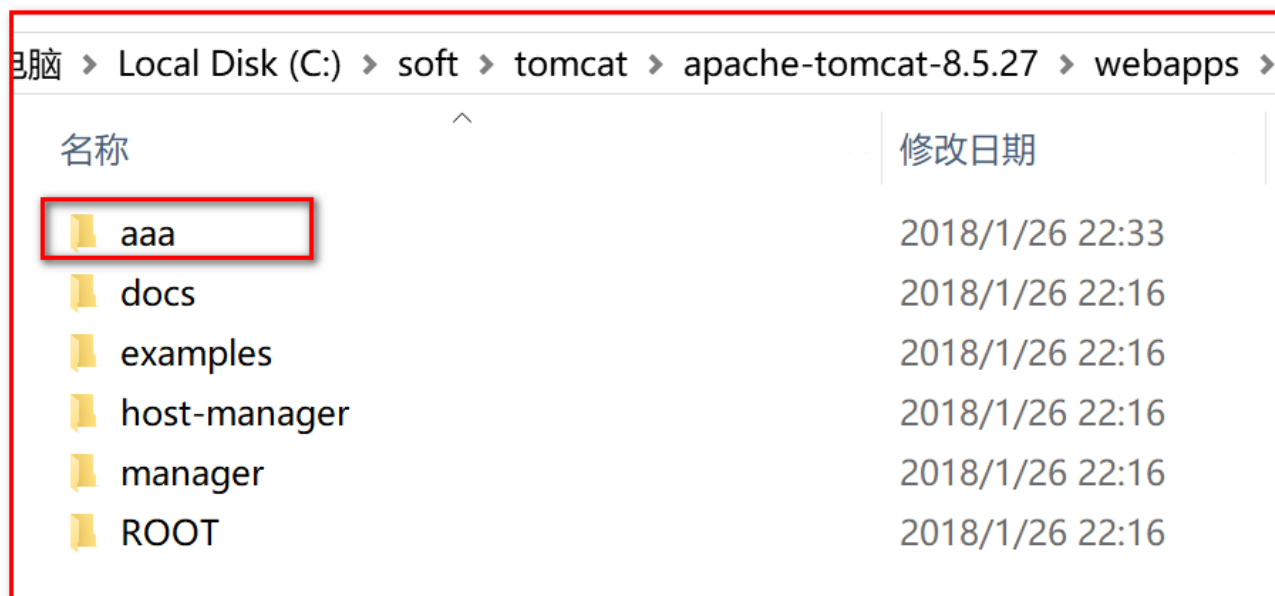
1.6.5 tomcat目录介绍



1.7 tomcat的发布方式

1.7.1 在webapps文件夹下面直接发布

只要将准备好的web资源直接复制到tomcat/webapps文件夹下, 就可以通过浏览器使用http协议访问获取
创建aaa文件夹:



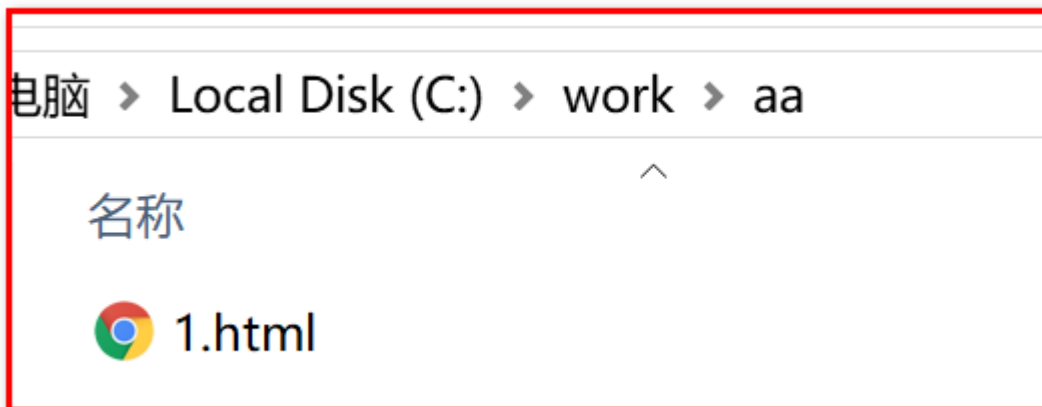
定义一个html文件, 内容为:

```
<h1>Hello Tomcat</h1>
```

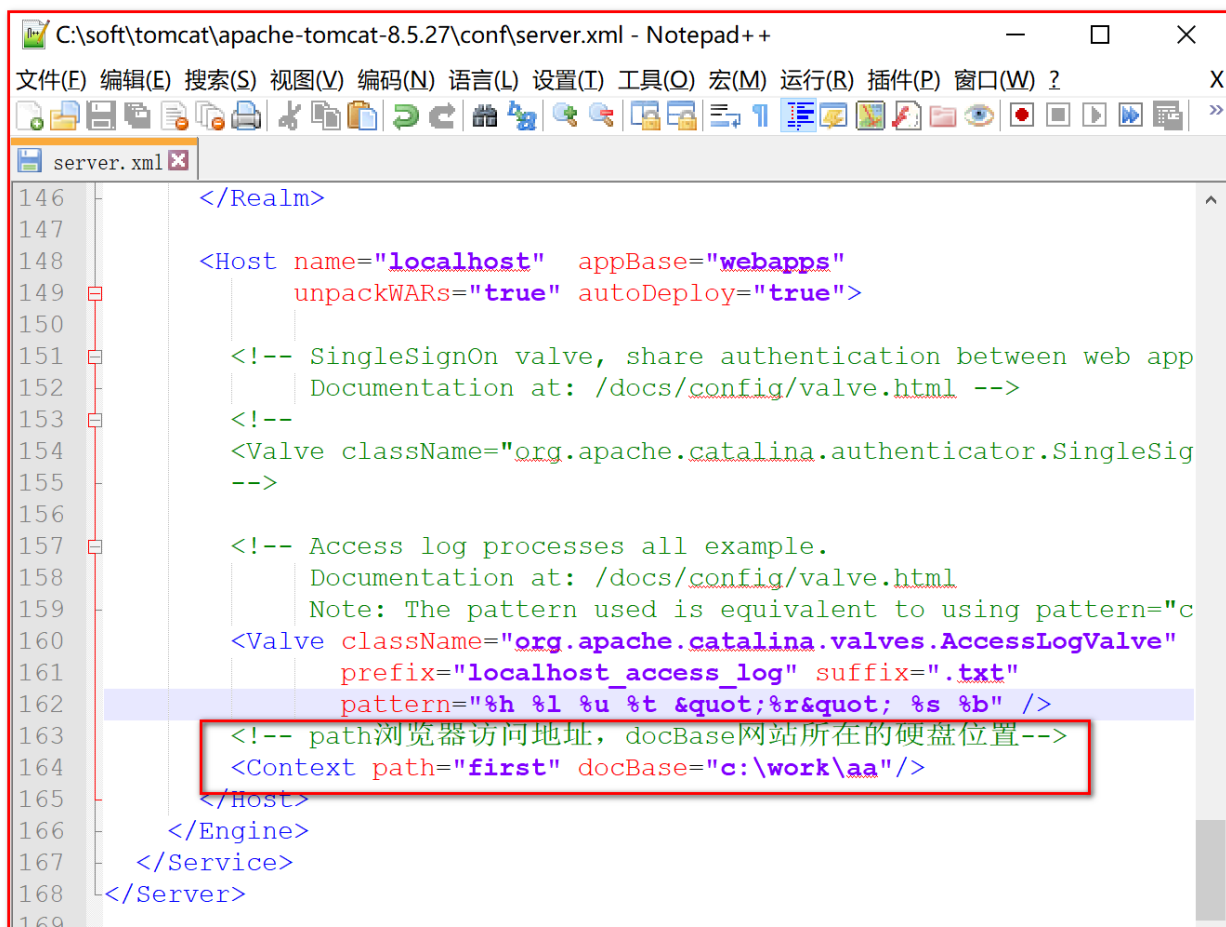
1.7.2 使用虚拟路径的方式发布项目 (两种方式)

1. 第一种: 配置server.xml, 添加context标签

第一步：在c盘work目录下创建一个文件夹为aa，在aa文件夹中添加一个1.html文件内容为：hello world！！




第二步：在tomcat/conf/server.xml中找到标签，添加标签，如图所示：




注：这种方式有一个缺点，就是server.xml是tomcat核心文件一旦出错，导致整个tomcat无法启动。

2. 第二种：配置独立xml文件

第一步：在tomcat/conf目录下新建一个Catalina目录（如果已经存在无需创建）

电脑 > Local Disk (C:) > soft > tomcat > apache-tomcat-8.5.27 > conf	
名称	修改日期
 Catalina	2018/1/26 22:20

第二步：在Catalina目录下创建localhost目录

电脑 > Local Disk (C:) > soft > tomcat > apache-tomcat-8.5.27 > conf > Catalina >		
名称	修改日期	类型
 localhost	2018/1/26 22:20	文件夹

第三步：在localhost中创建xml配置文件，名称为：second（注：这个名称是浏览器访问路径）

电脑 > Local Disk (C:) > soft > tomcat > apache-tomcat-8.5.27 > conf > Catalina > localhost			
名称	修改日期	类型	大小
 second.xml	2018/1/24 11:15	XML 文档	

第四步：添加xml文件的内容为：

第五步：在C:\work\bb下创建1.html，内容为“hello tomcat！！！！”，访问测试

← → ↻ ⓘ localhost:9090/second/1.html	
hello tomcat!!!	

第二种发布方式的优点：无需重启服务器自动加载和卸载项目

演示：在second.xml创建一个bak文件夹，将second.xml移动到bak文件夹之后：

电脑 > Local Disk (C:) > soft > tomcat > apache-tomcat-8.5.27 > conf > Catalina > localhost		
名称	修改日期	类型
 bak	2018/1/26 22:54	文件夹

服务器自动卸载项目：

```
2] ms
26-Jan-2018 23:01:19.254 信息 [ContainerBackgroundProcessor[StandardEngine[Catalina]]] org.apache.catalina.startup.HostC
onfig.undeploy Undeploying context [/second]
```

将second.xml移动回到localhost目录下：

电脑 > Local Disk (C:) > soft > tomcat > apache-tomcat-8.5.27 > conf > Catalina > localhost >

名称	修改日期	类型	大小
 bak	2018/1/26 23:02	文件夹	
 second.xml	2018/1/24 11:15	XML 文档	

服务器自动加载项目（需要等待一会儿时间）：

```
26-Jan-2018 23:02:49.312 信息 [localhost-startStop-3] org.apache.catalina.startup.HostConfig.deployDescriptor Deploying configuration descriptor [C:\soft\tomcat\apache-tomcat-8.5.27\conf\Catalina\localhost\second.xml]
26-Jan-2018 23:02:49.332 信息 [localhost-startStop-3] org.apache.catalina.startup.HostConfig.deployDescriptor Deployment of configuration descriptor [C:\soft\tomcat\apache-tomcat-8.5.27\conf\Catalina\localhost\second.xml] has finished in [19] ms
```

第2章servlet入门

2.1 Servlet2.5实现Hello world例子

2.1.1 servlet的基本概述

Servlet 运行在服务端的Java小程序，是sun公司提供一套规范，用来处理客户端请求、响应给浏览器的动态资源。但servlet的实质就是java代码，通过java的API动态的向客户端输出内容

1. 查阅JavaEE手册（帮助文档）阅读Servlet规范：

javax.servlet

Interface Servlet

All Known Subinterfaces:
[HttpJspPage](#), [JspPage](#)

All Known Implementing Classes:
[FacesServlet](#), [GenericServlet](#), [HttpServlet](#)

public interface Servlet

Implemented by: [FacesServlet](#), [GenericServlet](#), [JspPage](#)

定义所有 servlet 都必须实现的方法。

servlet 是运行在 Web 服务器中的小型 Java 程序。servlet 通常通过 HTTP（超文本传输协议）接收和响应来自 Web 客户端的请求。

要实现此接口，可以编写一个扩展 javax.servlet.GenericServlet 的一般 servlet，或者编写一个扩展 javax.servlet.http.HttpServlet 的 HTTP servlet。

此接口定义了初始化 servlet 的方法、为请求提供服务的方法和从服务器移除 servlet 的方法。这些方法称为生命周期方法，它们是按以下顺序调用的：

1. 构造 servlet，然后使用 init 方法将其初始化。

2. 处理来自客户端的对 service 方法的所有调用。

3. 从服务中取出 servlet，然后使用 destroy 方法销毁它，最后进行垃圾回收并终止它。

根据文档总结，书写servlet一个三个步骤：

- 1) 创建一个class实现servlet接口
- 2) 重写service方法
- 3) 创建的类必须在web.xml文件中做配置

2. 为什么要做配置？

答：必须将请求路径和java程序的对应关系建立起来。

2.1.2 servlet与普通的java程序的区别

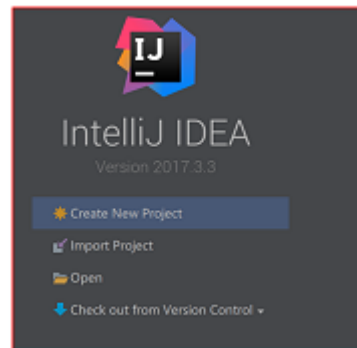
1. 必须实现servlet接口
2. 必须在servlet容器（服务器）中运行
3. servlet程序可以接收用户请求参数以及向浏览器输出数据

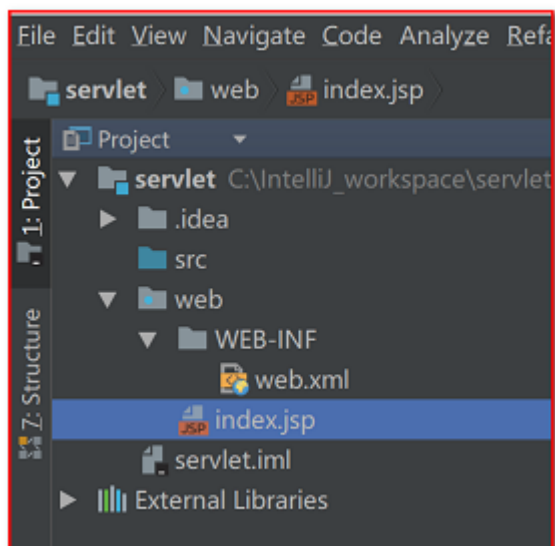
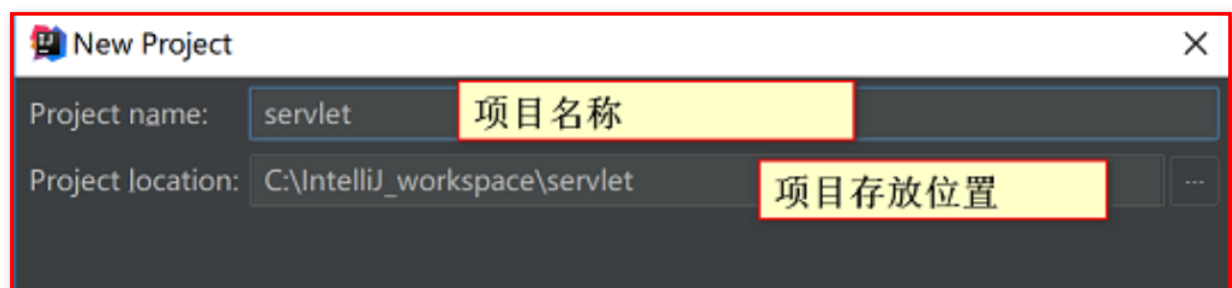
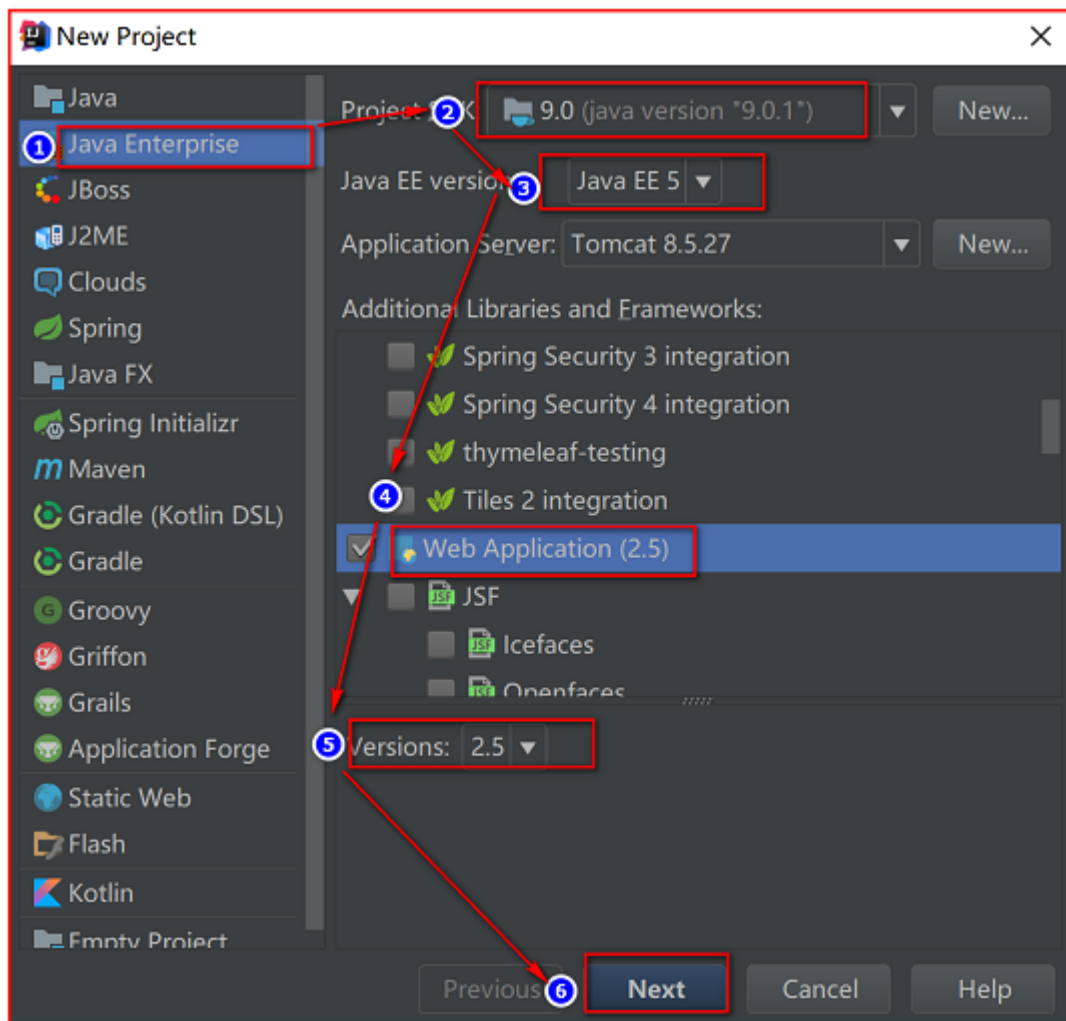
2.1.3 代码实现servlet的步骤

1. 创建web工程
2. 在cn.itcast.web包下创建一个类实现 Servlet接口
3. 实现service方法
4. 在web.xml中配置书写好的servlet

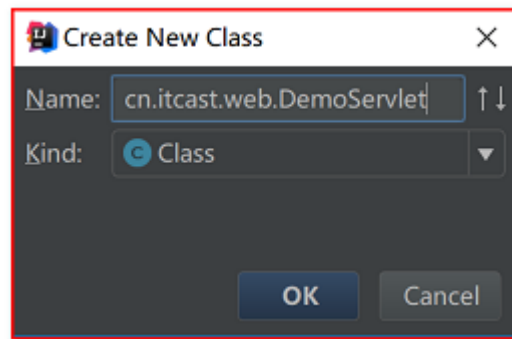
2.1.4 servlet代码实现

1. 创建web工程





2. 在cn.itcast.web包下创建一个类实现Servlet接口



servlet代码：

```
package cn.itcast.web;

import javax.servlet.*;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class DemoServlet implements Servlet{

    @Override
    public void init(ServletConfig servletConfig) throws ServletException {
    }

    @Override
    public ServletConfig getServletConfig() {
        return null;
    }

    @Override
    public void service(ServletRequest servletRequest, ServletResponse servletResponse) throws
    ServletException, IOException {
        System.out.println("第一个servlet程序");
    }

    @Override
    public String getServletInfo() {
        return null;
    }

    @Override
    public void destroy() {
    }
}
```

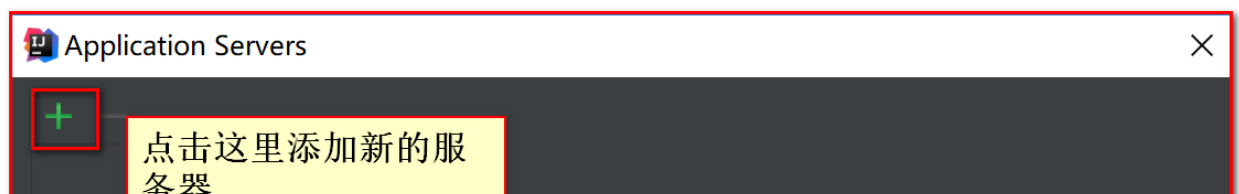
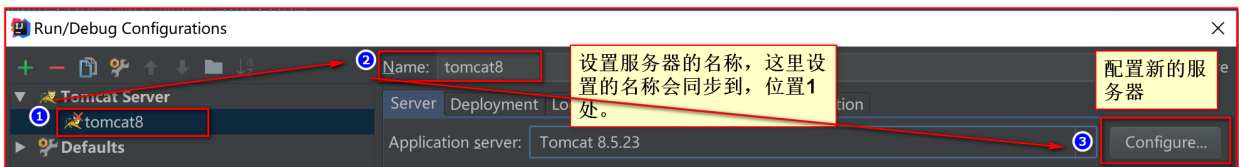
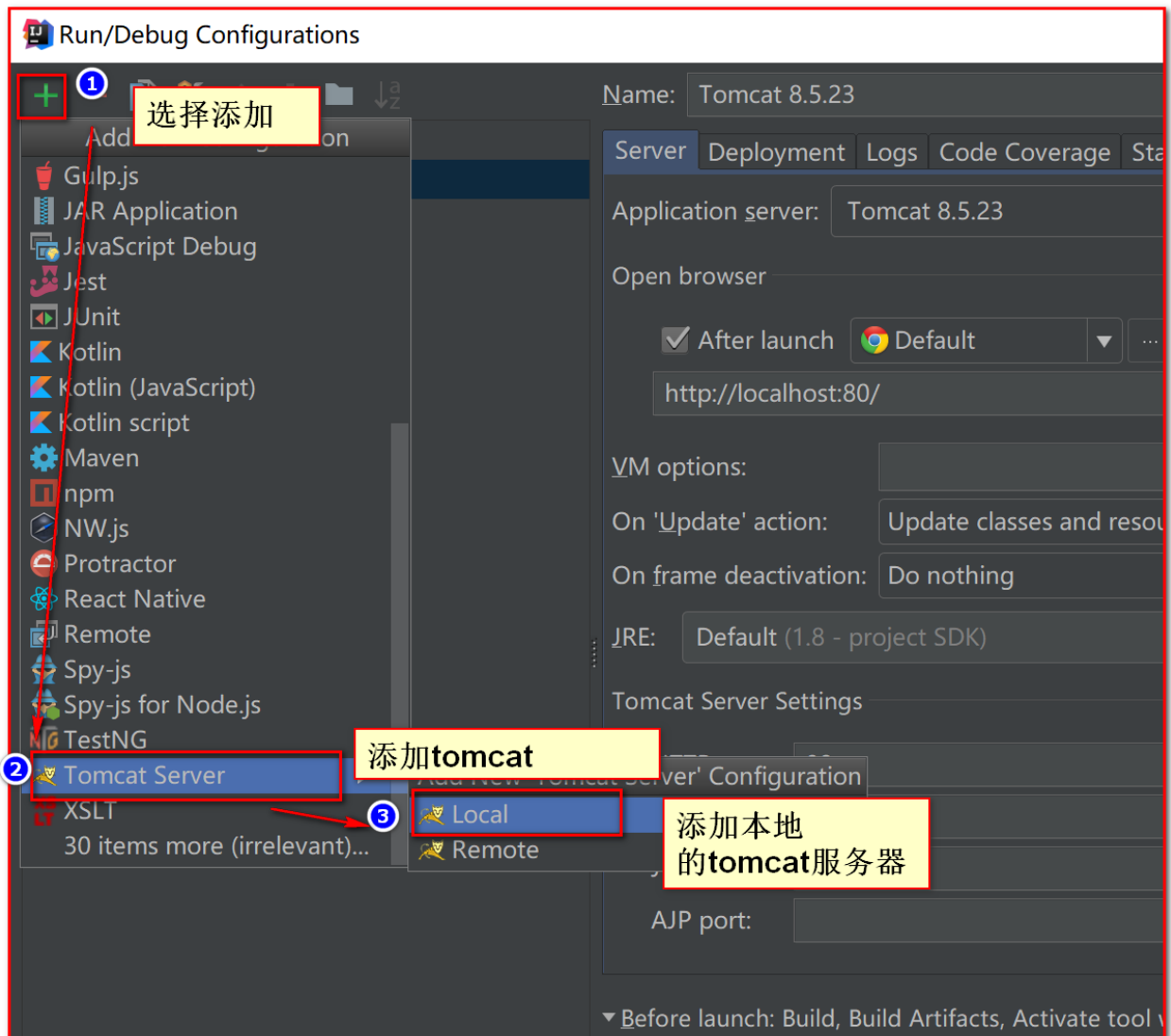
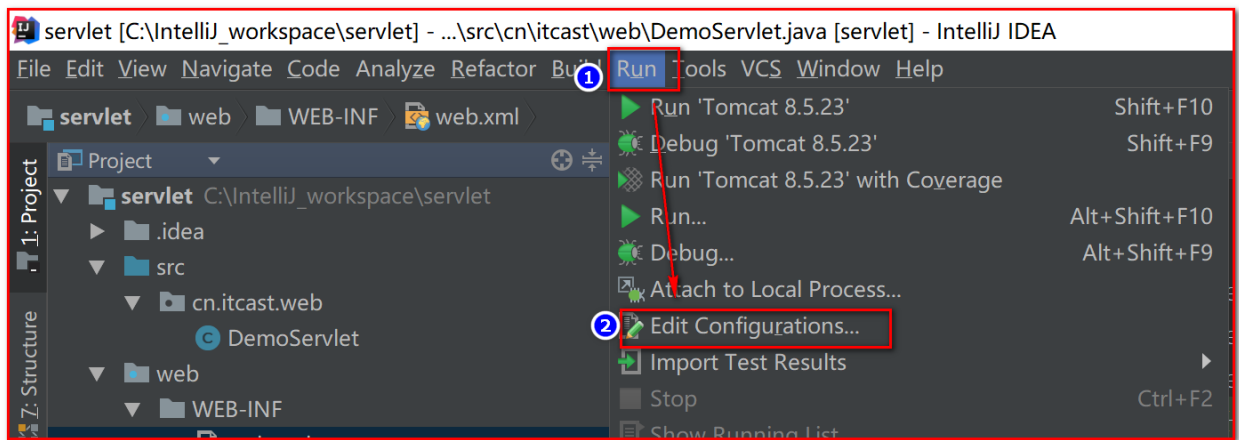
Web.xml配置（该文件在web/WEB-INF 文件夹下）：

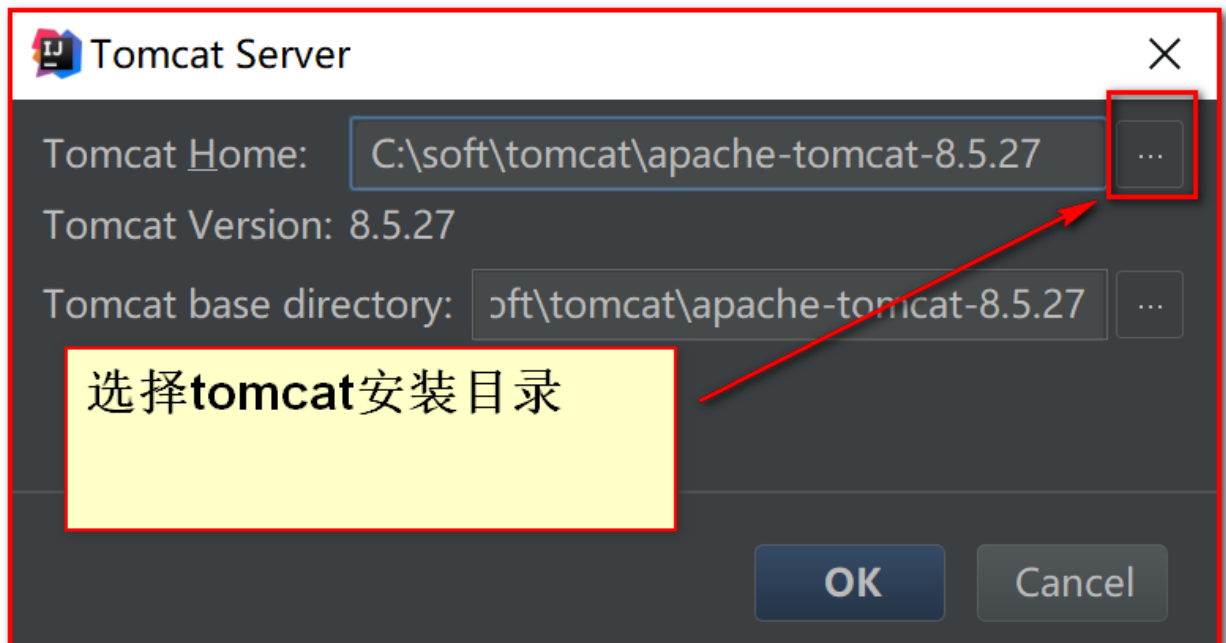
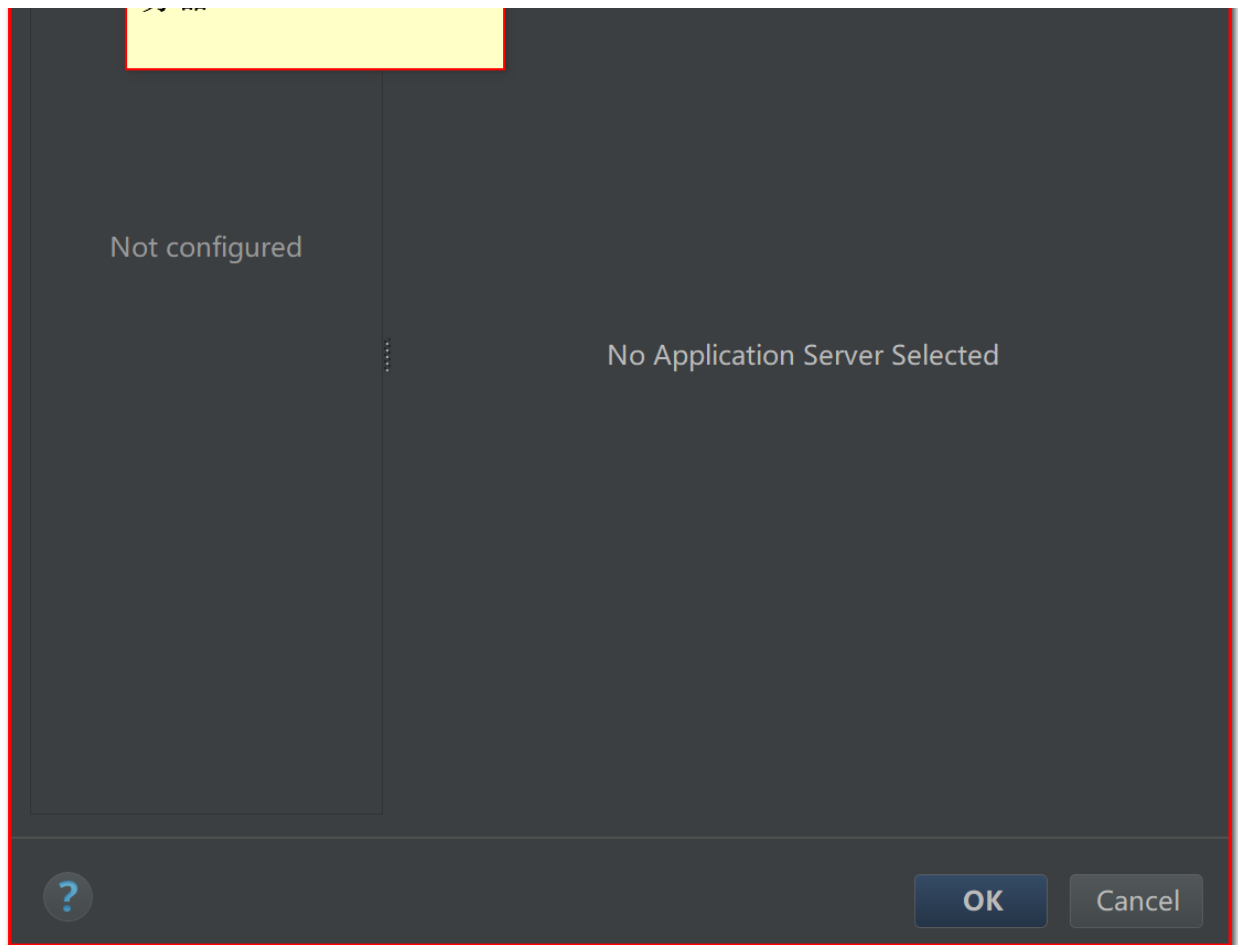
```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
version="2.5">
  <servlet>
    <servlet-name>DemoServlet</servlet-name>
    <servlet-class>cn.itcast.web.DemoServlet</servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>DemoServlet</servlet-name>
    <url-pattern>/demo</url-pattern>
  </servlet-mapping>
</web-app>
```

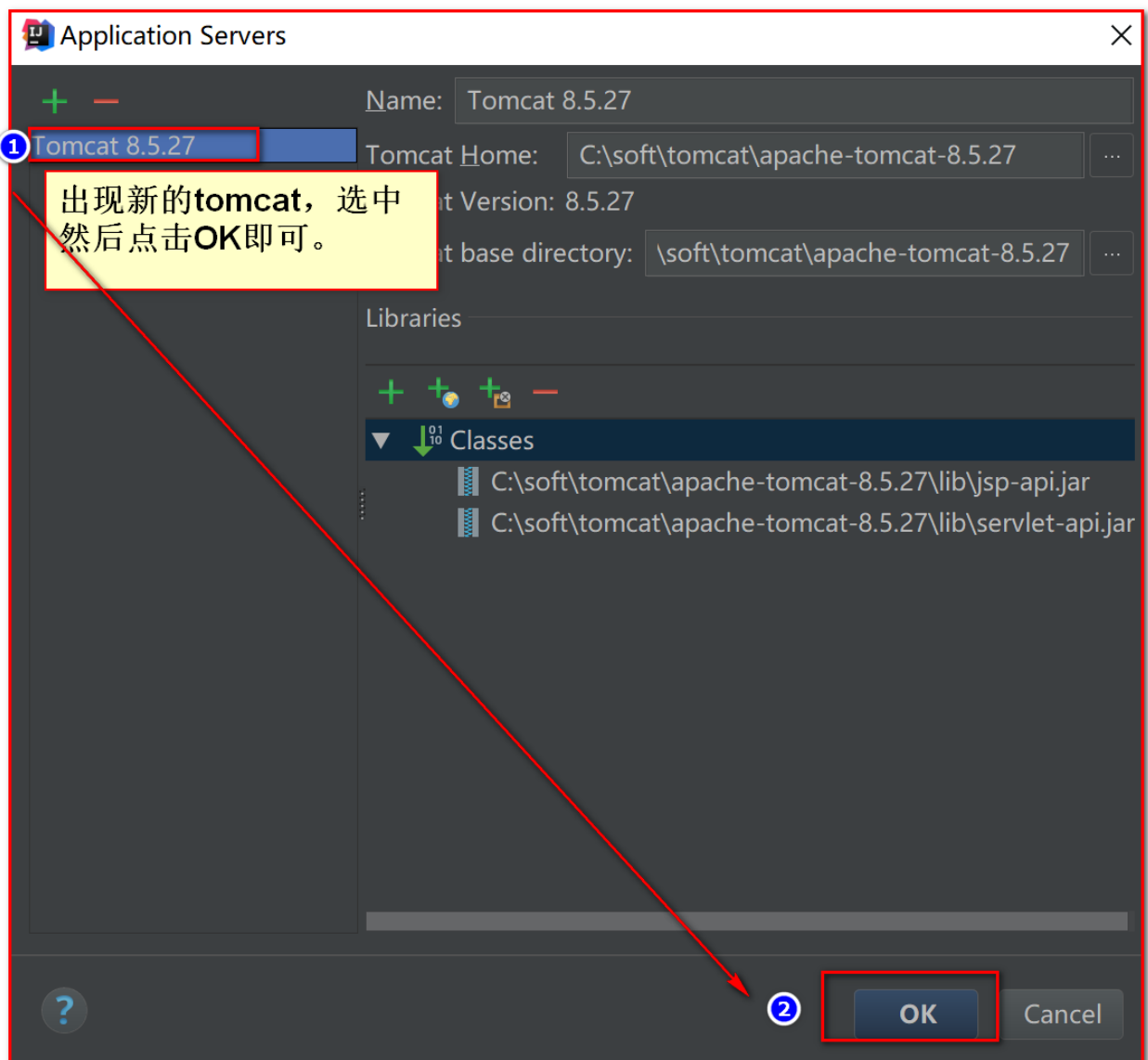
2.1.4 idea配置tomcat方式发布

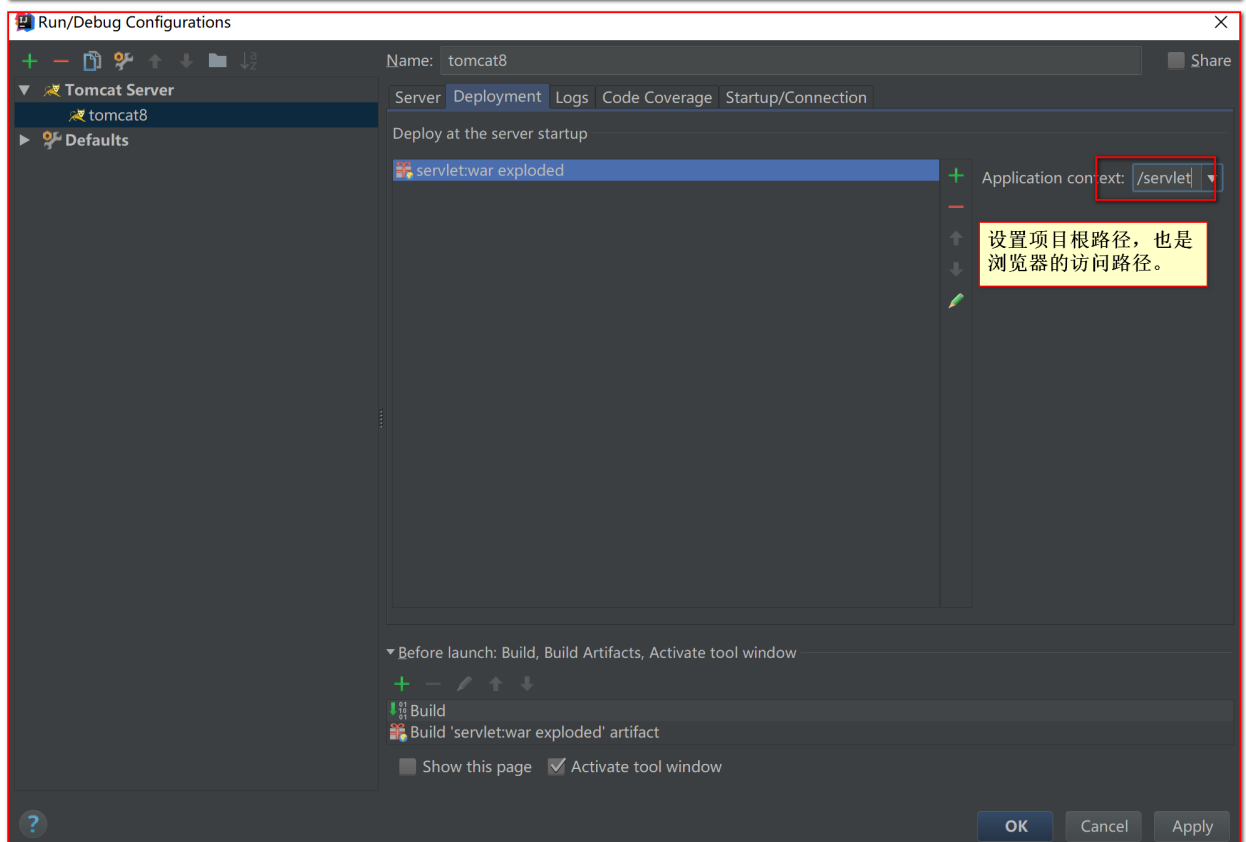
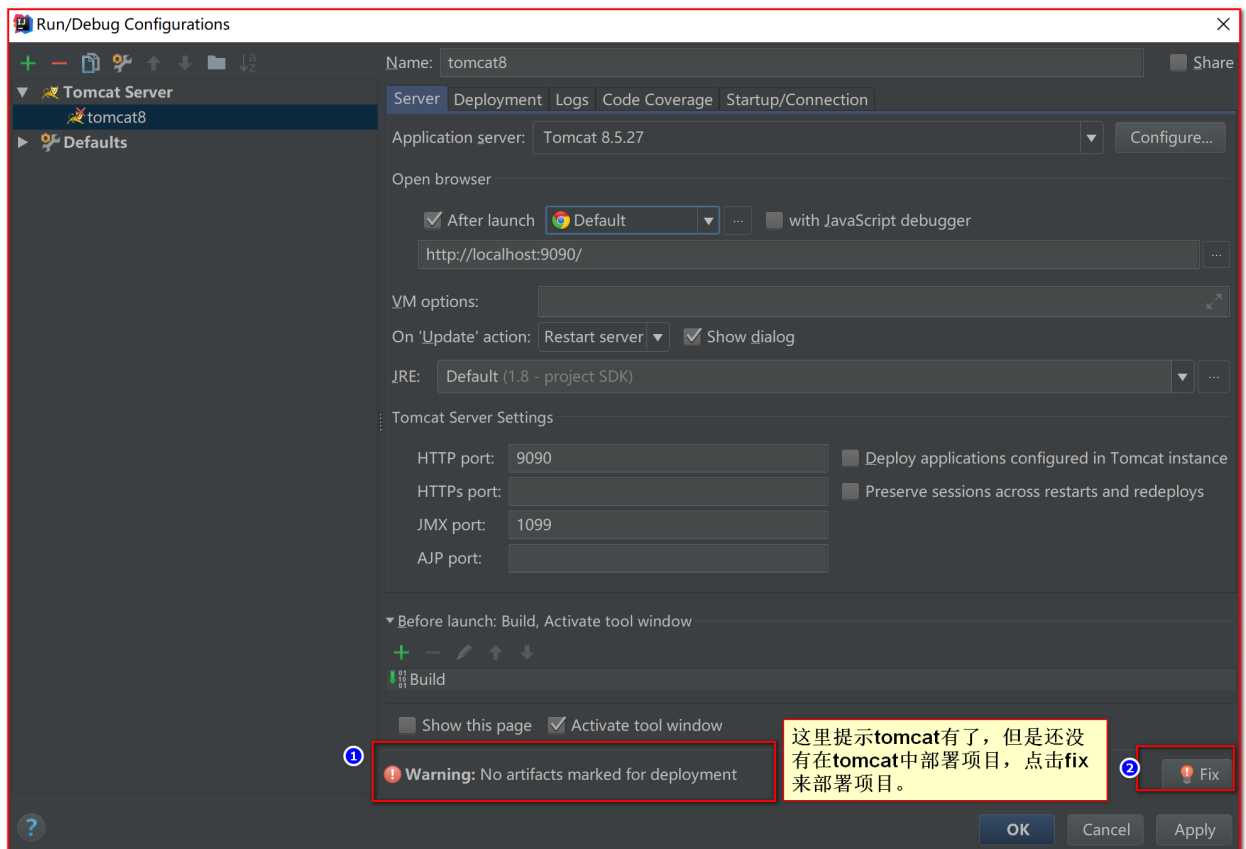
Servlet已经书写完成，接下来，我们要将书写好的servlet发布到tomcat上去。 接下里我们要将idea和tomcat集成到一起，可以通过idea就控制tomcat的启动和关闭：

1. 添加tomcat服务器部署项目



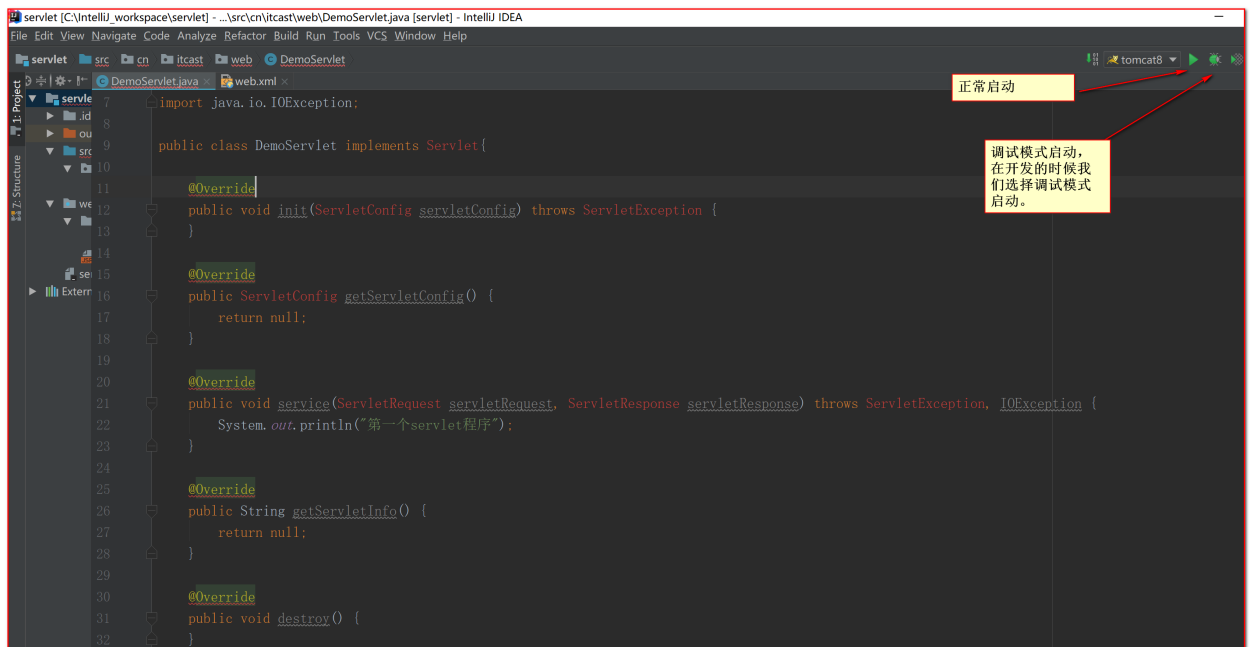






点击OK设置完成！

2. 启动服务器：



3. 浏览器测试访问：

<http://localhost:9090/servlet/demo>

4. 控制台打印效果：

