# Tomcat服务器&servlet入门程序

# 学习目标

1.能够理解软件的架构

2.能够理解WEB资源概念

3.能够理解WEB服务器

4.能够启动关闭Tomcat服务器

5.能够解决Tomcat服务器启动时遇到的问题

6.能够运用Tomcat服务器部署WEB项目

7.能够使用idea编写servlet

8.能够使用idea配置tomcat方式发布项目

9.能够使用注解开发servlet

10.理解servlet运行原理

# tomcat服务器

## 软件的架构

### 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(client)/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 。**

1. **必须先有请求 再有响应**
2. **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模型可以理解为一种特殊C/S模型。

## web资源的类别

web资源,只要是浏览器能看到的都是web资源

### 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。

使用浏览器访问的资源以资源文件形式存在的就是静态资源,处理的是一个物理文件.

### 动态资源

      指web页面中供人们浏览的数据是由程序(代码)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,浏览器访问的资源都是明确的资源文件。  
 动态资源,浏览器访问的资源是由程序运行产生的数据

​ **疑问:浏览器访问的web资源,web资源是由谁提供给浏览器的呢?**

## 服务器的概念

### 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。

## 服务器的作用

### 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。**

## 常见的web应用服务器软件

### 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JavaEE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。

优点:性能非常强大

缺点:商用非常贵,学习用免费

1. **WebSphere**

​ IBM公司的WebSphere，支持JavaEE规范。WebSphere 是随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。

1. **Glass Fish**

​ 最早是Sun公司的产品，后来被Oracle收购，开源，中型服务器。

1. **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。

1. **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。

## tomcat服务器软件安装和介绍

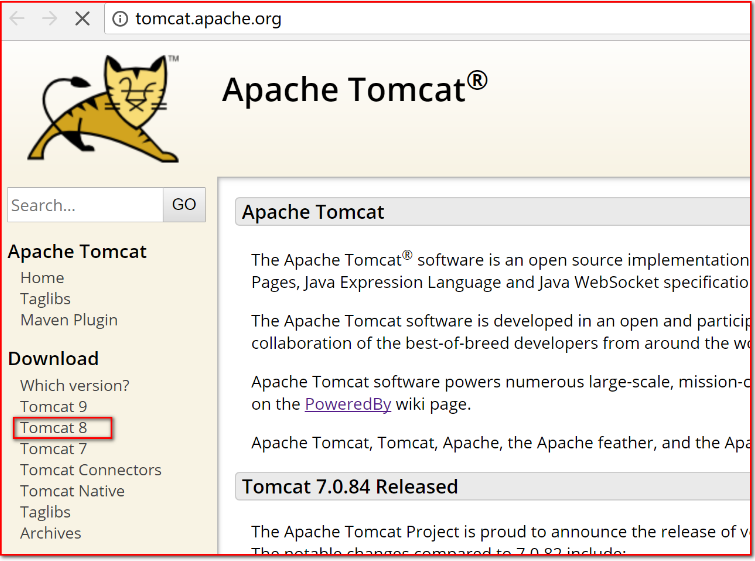
Tomcat基本概述：

​ 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。

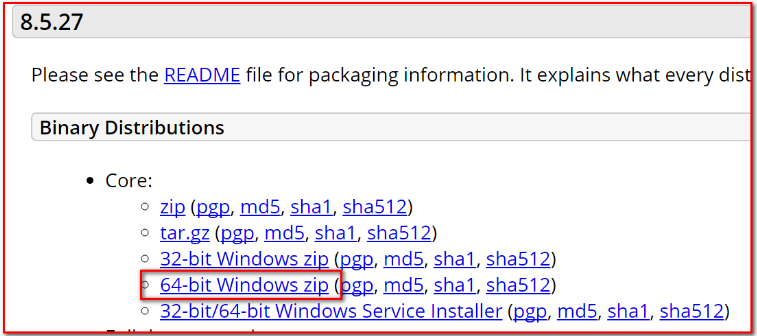
​ 因为Tomcat技术先进、性能稳定，而且免费，因而深受Java爱好者的喜爱并得到了部分软件开发商的认可，是目前比较流行的Web应用服务器。

### tomcat服务器软件下载

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



1. **选择要下载的文件（红框所示）**：

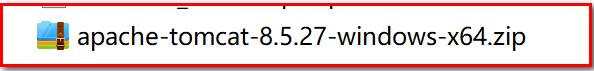


tar.gz 文件 是linux操作系统下的安装版本

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）

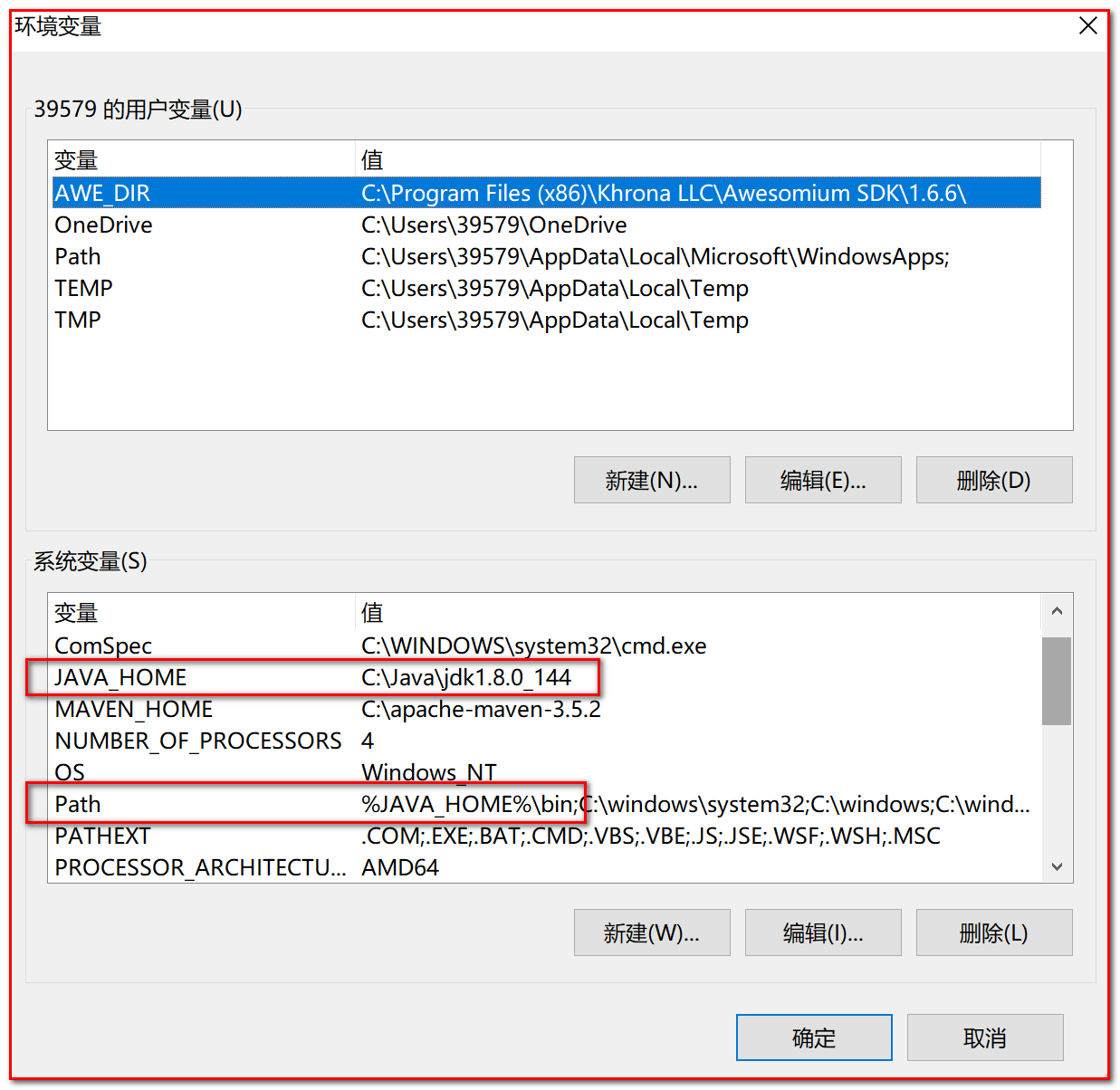
1. **下载完成**：



### 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**：

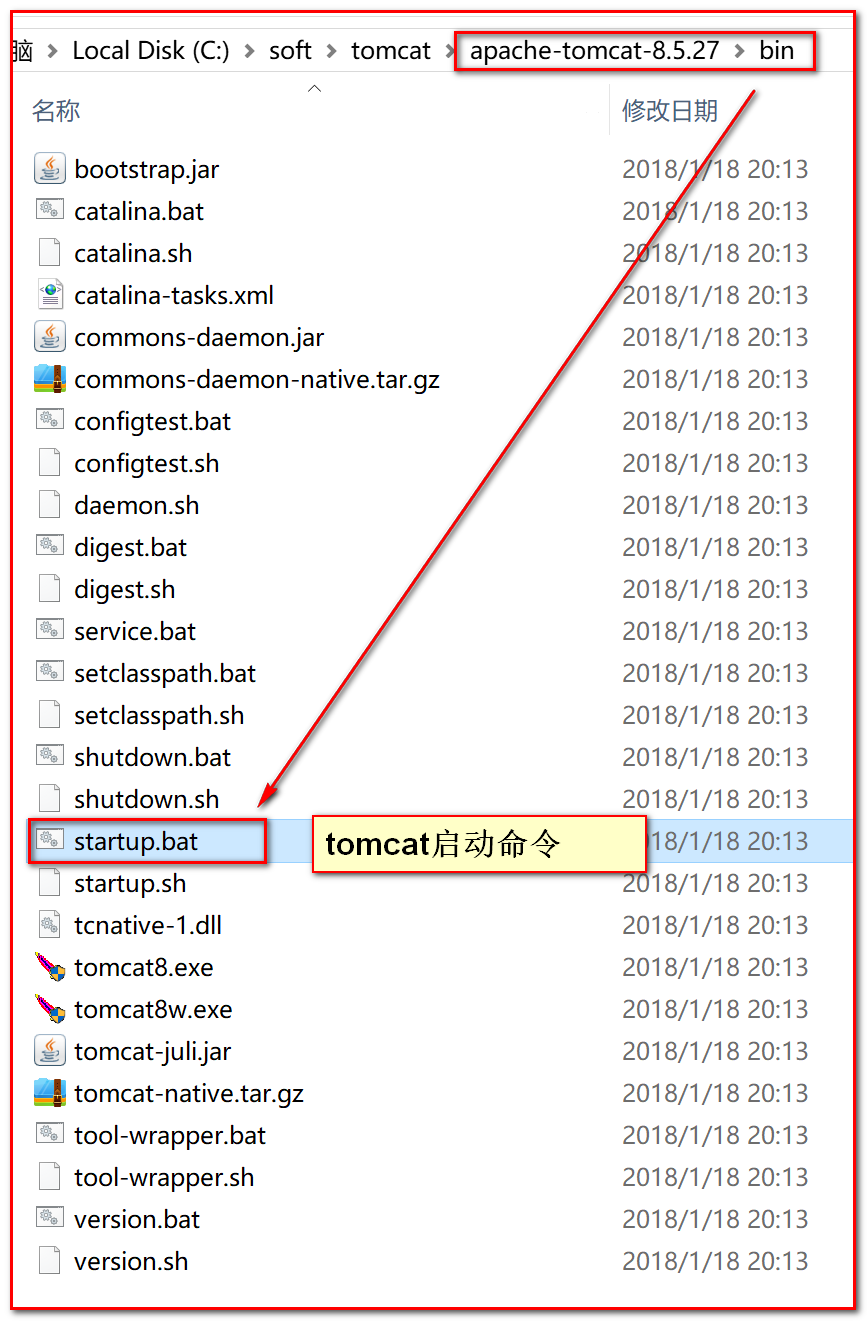
​tomcat运行依赖于java环境：



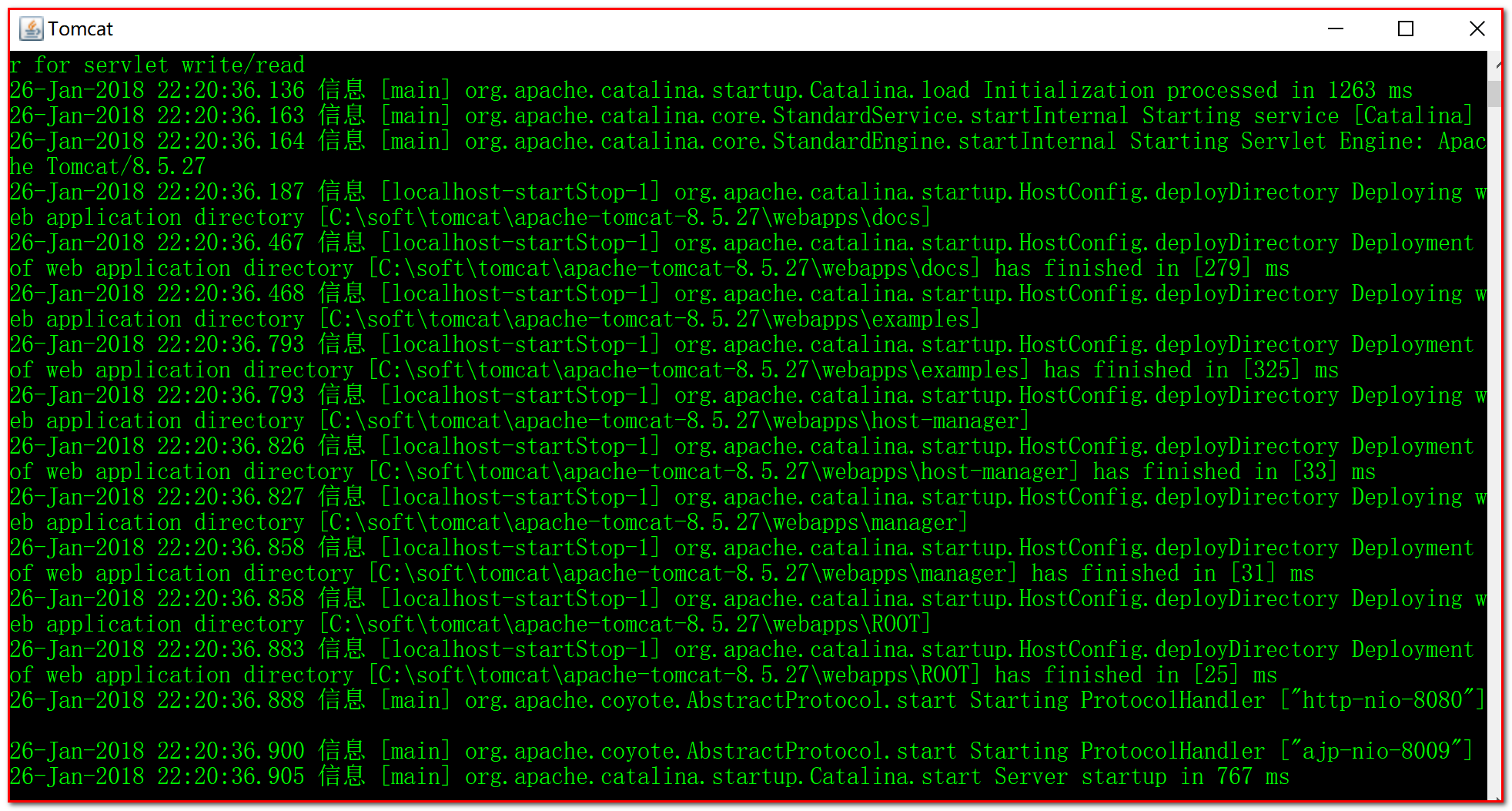
### 启动与关闭tomcat服务器

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：



启动效果：

​

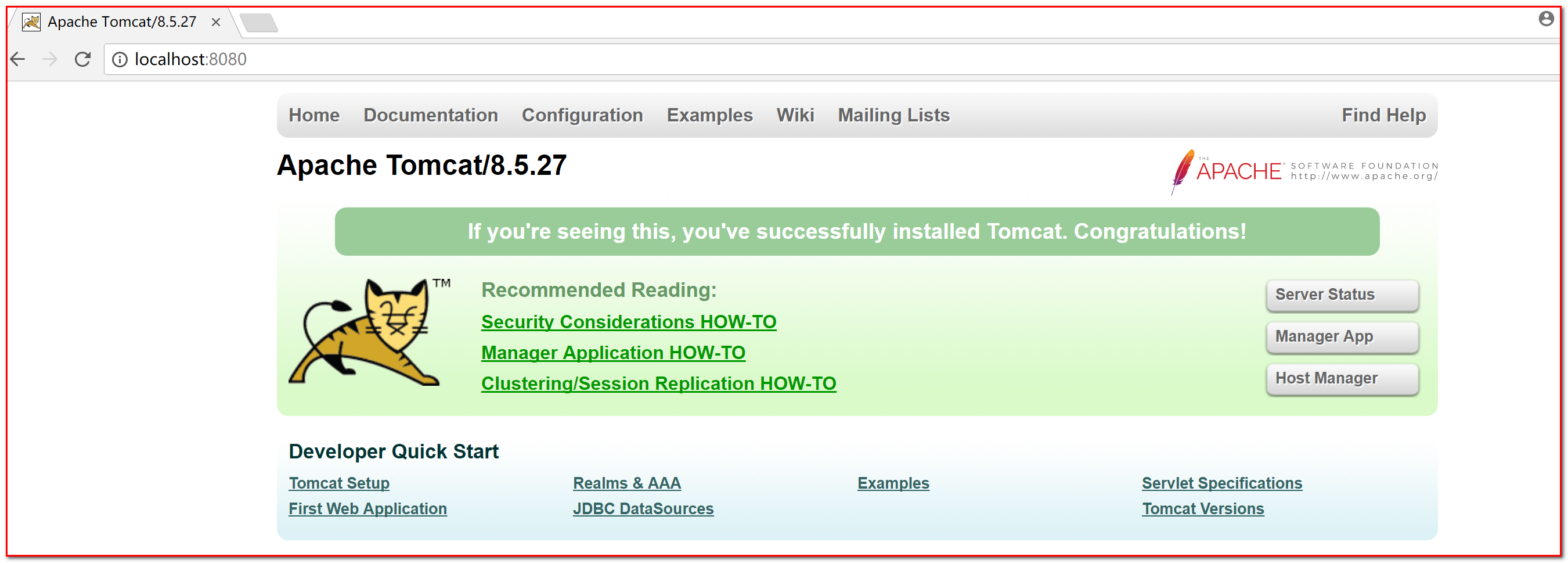
1. **测试访问tomcat服务器**

打开浏览器在，在浏览器的地址栏中输入：

​<http://127.0.0.1:8080>

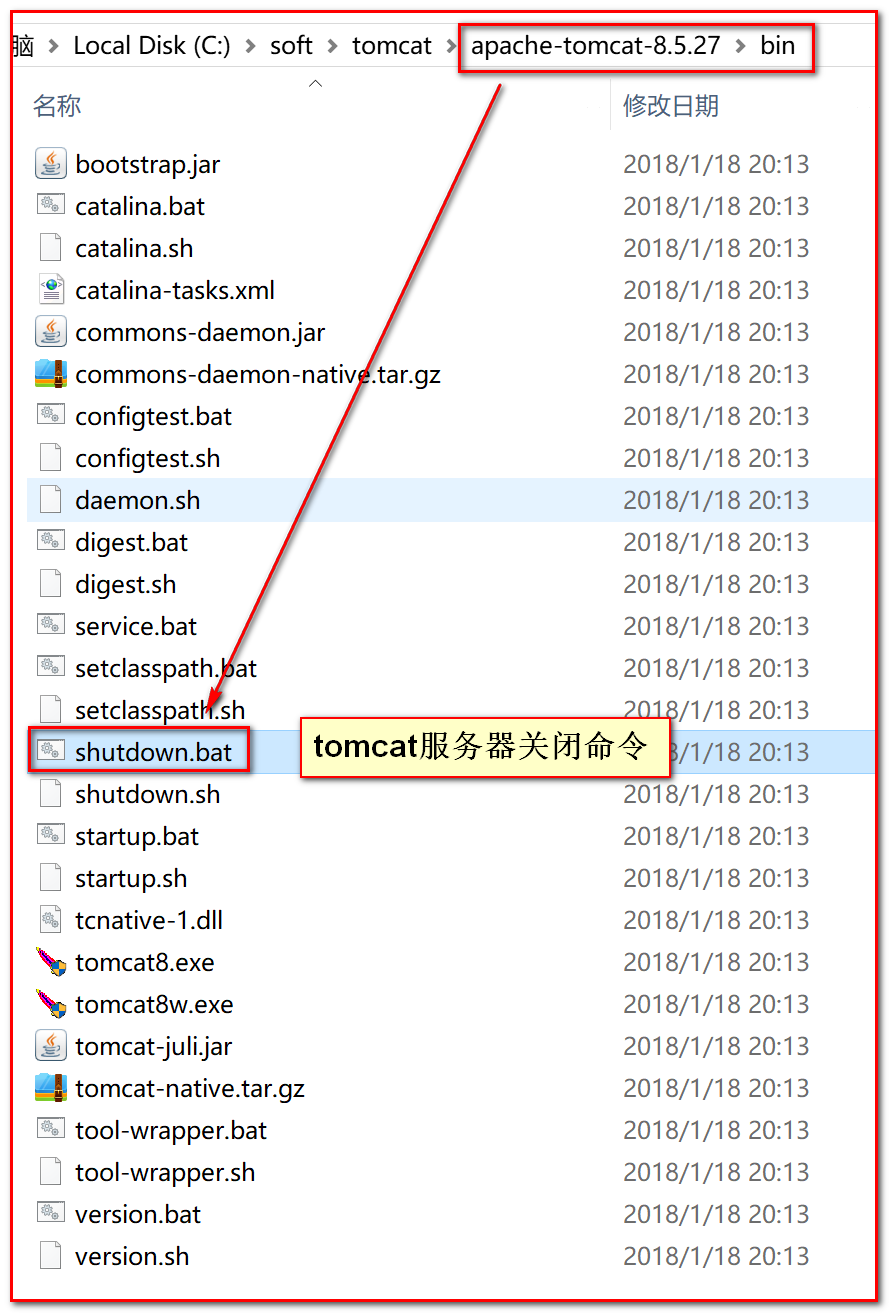
​<http://localhost:8080>

注： Localhost相当于127.0.0.1

​

1. **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：



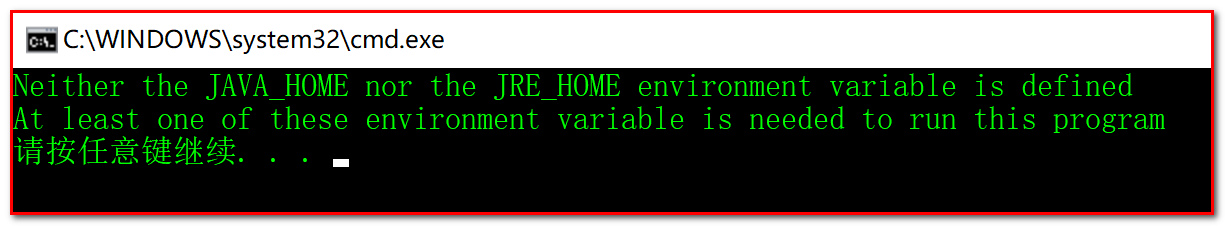
### tomcat安装常见的问题

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：

​

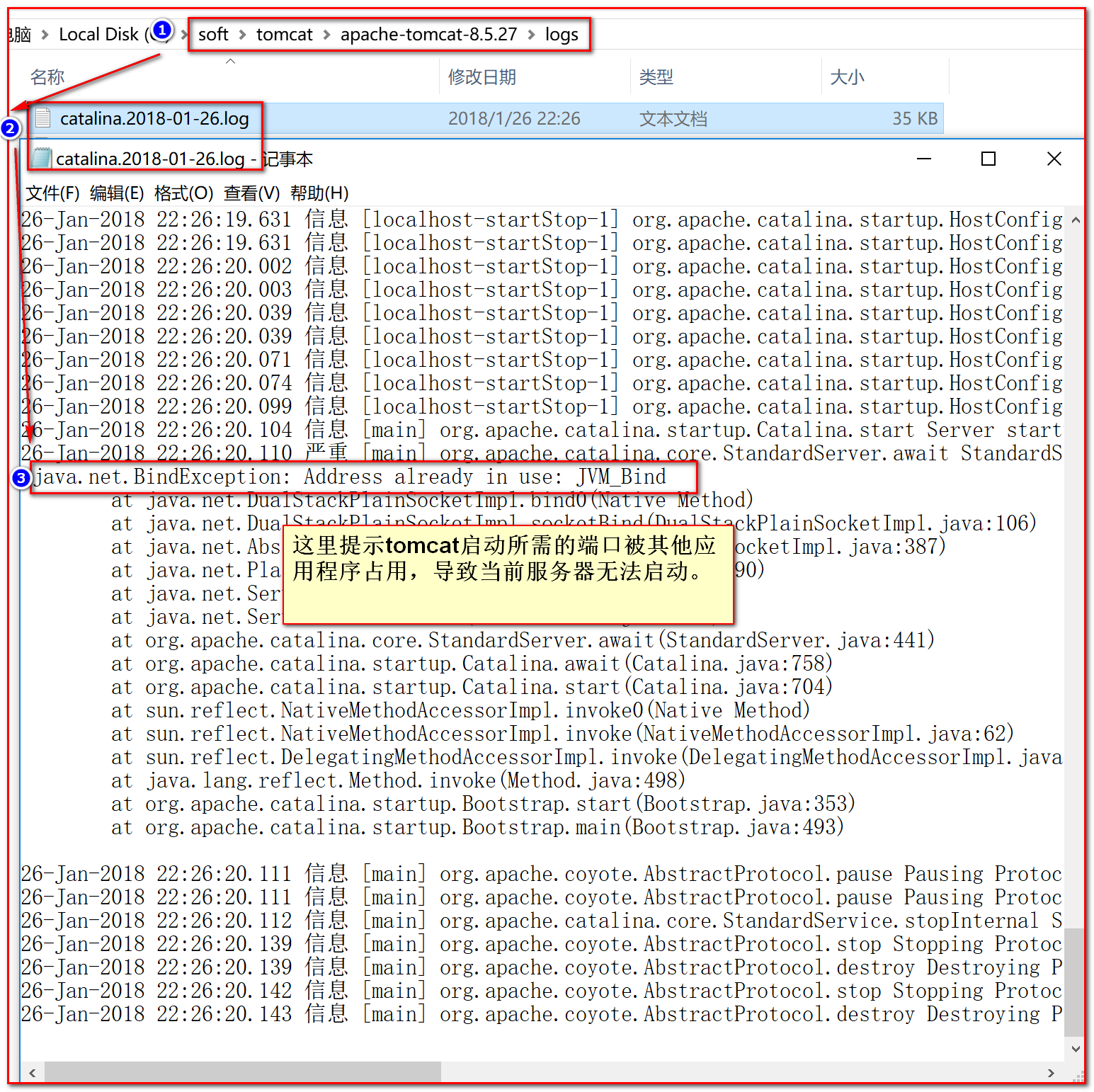
有的同学配置上面的JAVA\_HOME后闪退依然存在,这样的学员电脑需要再配置CATALINA\_HOME环境变量, CATALINA\_HOME: H:\server\apache-tomcat-8.5.27

1. **端口被占用导致启动失败**

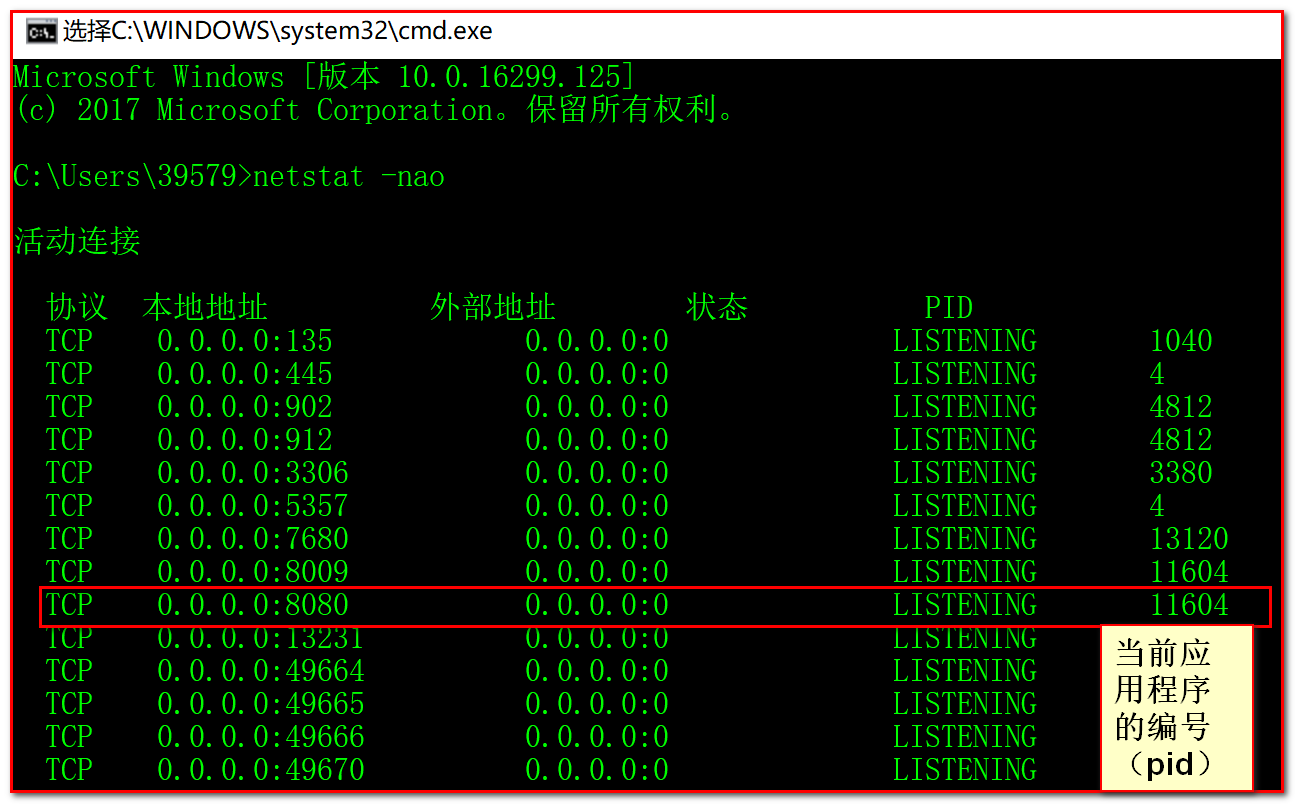
如果启动的时候，发生异常问题，这时有可能是端口被占用。

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。

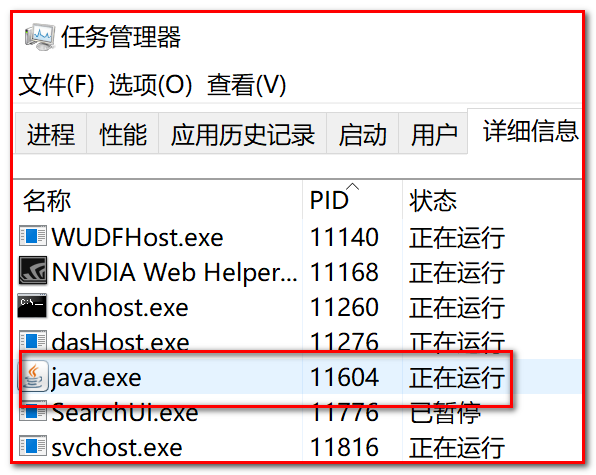
报错内容可以通过查询tomcat目录下的logs目录中Catalina.当前系统年月日.log文件查看，如下图：



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：



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：

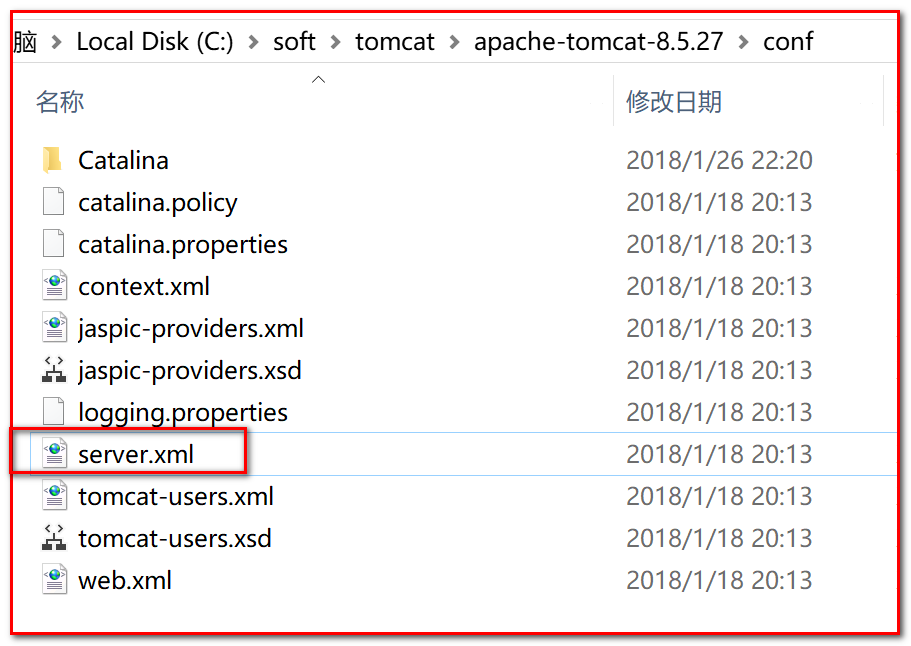


注意：如果这个进程是操作系统的任务进程，这时一般是不能停止这个进程。

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：

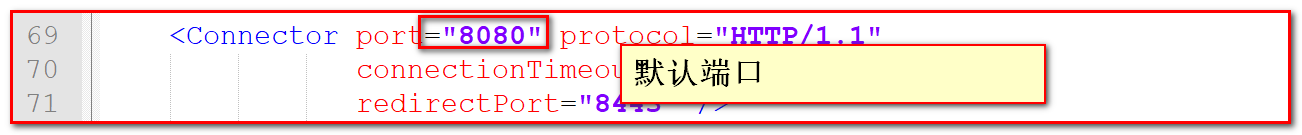
1. **修改tomcat启动端口**

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



​ 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：



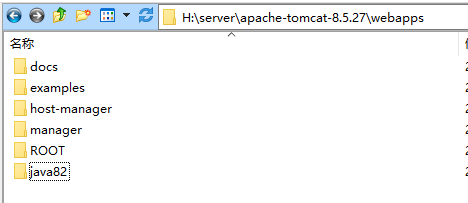
​ 修改tomcat的端口为9090：

​ 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>

服务器作用:提供服务,服务器有能力将本地资源(web资源)发布出去给外界用户使用浏览器访问.

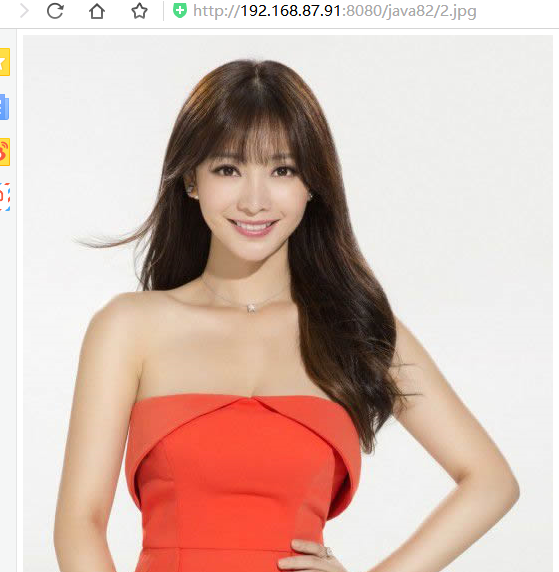
步骤1:进入webapps目录

步骤2:创建一个目录名为java82

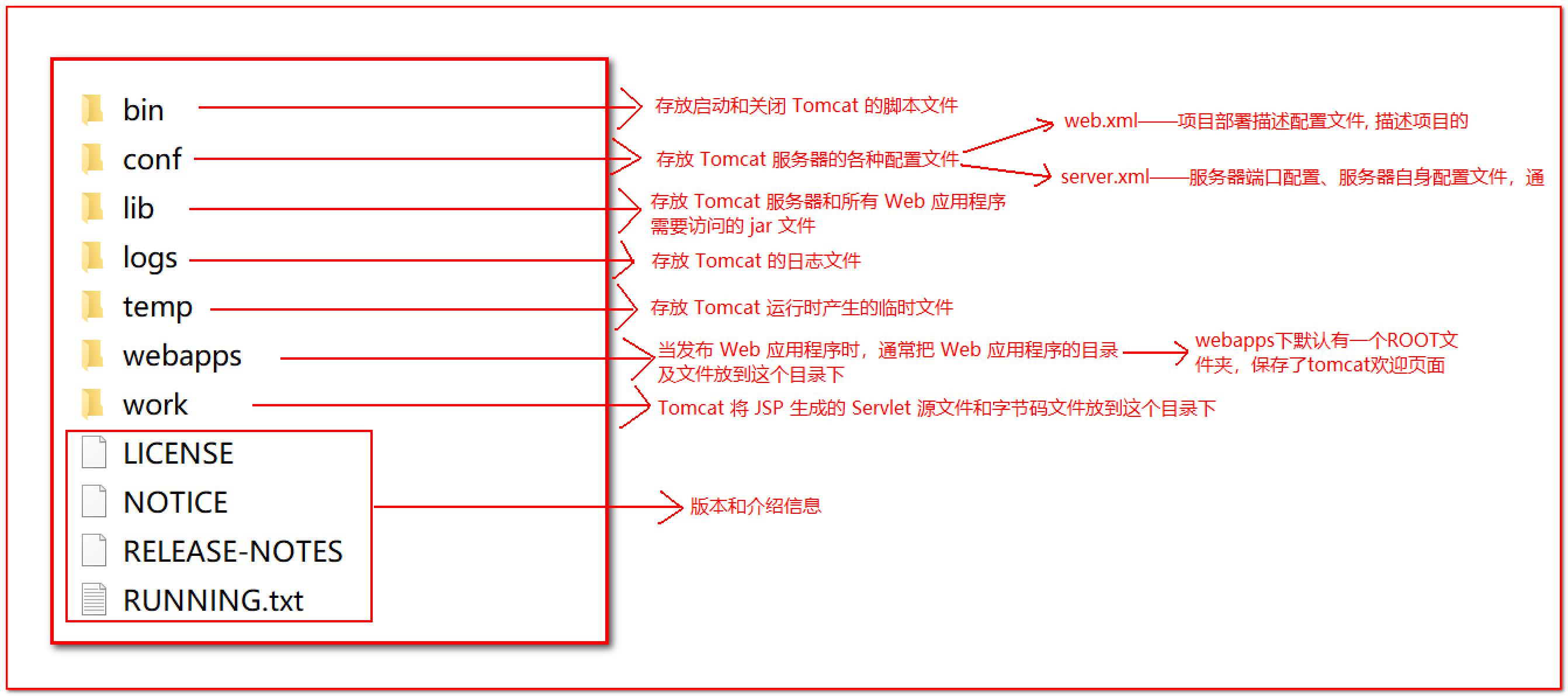


步骤3:在java82目录里面发布一张图片

步骤4:打开浏览器访问<http://localhost:8080/java82/2.jpg>



### tomcat目录介绍



webapps目录下发布资源,资源文件必须存放在资源目录里面,不能单独存在.如果希望直接访问资源文件,可以将资源文件放到ROOT目录下

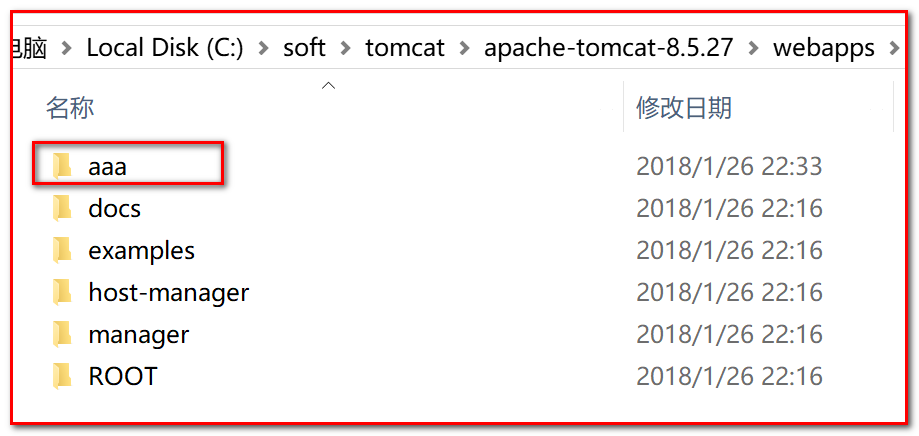
重点掌握的目录:bin,conf,webapps,logs

## tomcat的发布方式

### 在webapps文件夹下面直接发布(重要)

​ **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：



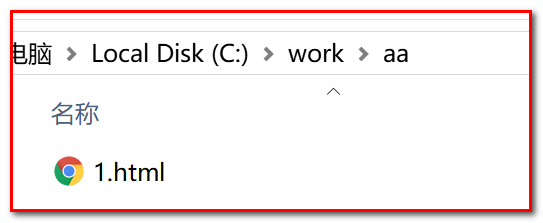
定义一个html文件，内容为：

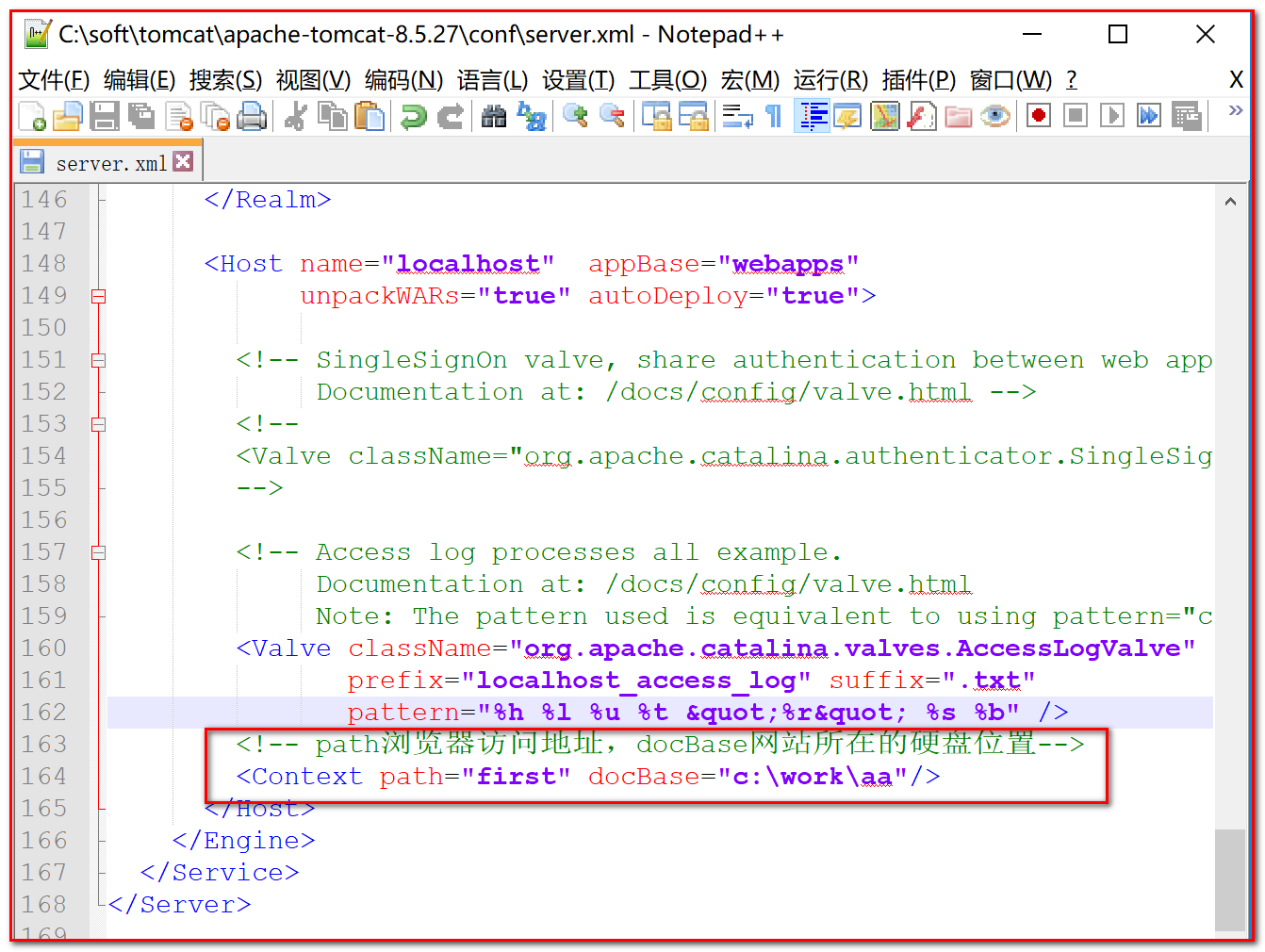
<h1>Hello Tomcat</h1>

### 使用虚拟路径的方式发布项目（两种方式,了解）

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！！！

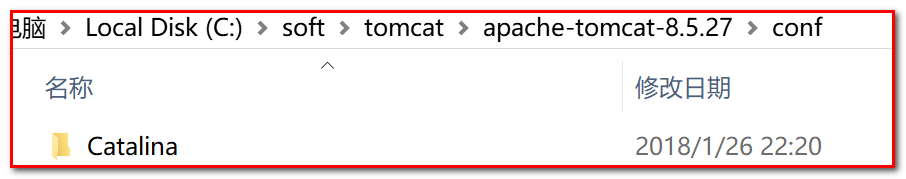
​ 

​ 第二步：在tomcat/conf/server.xml中找到<Host>标签，添加<Context>标签，如图所示：

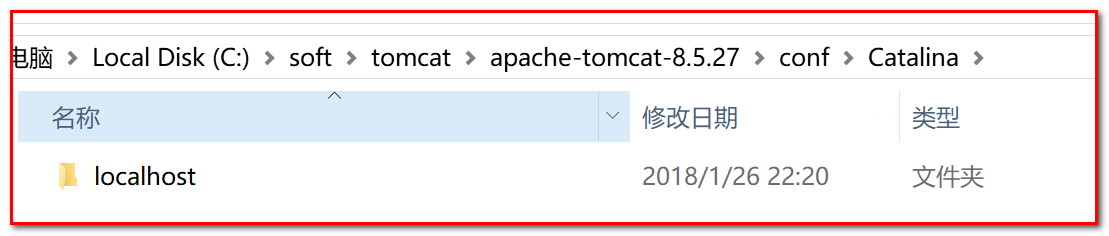
**注：这一种方式有一个缺点，就是server.xml是tomcat核心文件一旦出错，导致整个tomcat无法启动。**

1. **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）



​第二步：在Catalina目录下创建localhost目录



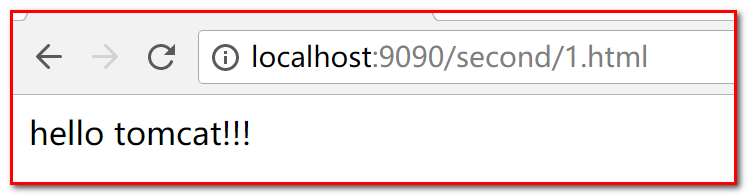
​第三步：在localhost中创建xml配置文件，名称为：second（注：这个名称是浏览器访问路径）



​第四步：添加xml文件的内容为：

<Context docBase="C:\work\bb" />

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



**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**

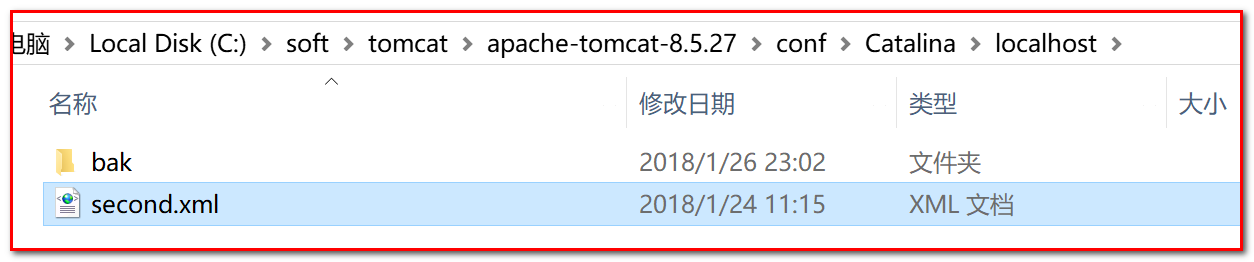
演示：在second.xml创建一个bak文件夹，将second.xml移动到bak文件夹之后：



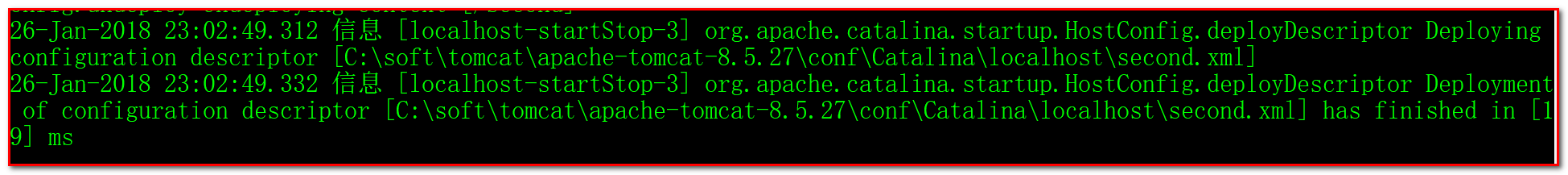
服务器自动卸载项目：



将second.xml移动回到localhost目录下：



服务器自动加载项目（需要等待一会儿时间）：



### tomcat与nginx的区别

1. **设计目的：**

Tomcat是一个免费的开源的Servlet容器，实现了JAVAEE规范，遵循http协议的的服务器Nginx是一款轻量级的电子邮件（电子邮件遵循IMAP/POP3协议）代理服务器，后来又发展成可以部署web应用程序和进行反向代理的服务器

1. **存放内容：**

tomcat可以存放静态和动态资源

nginx可以存放静态资源

1. **应用场景：**

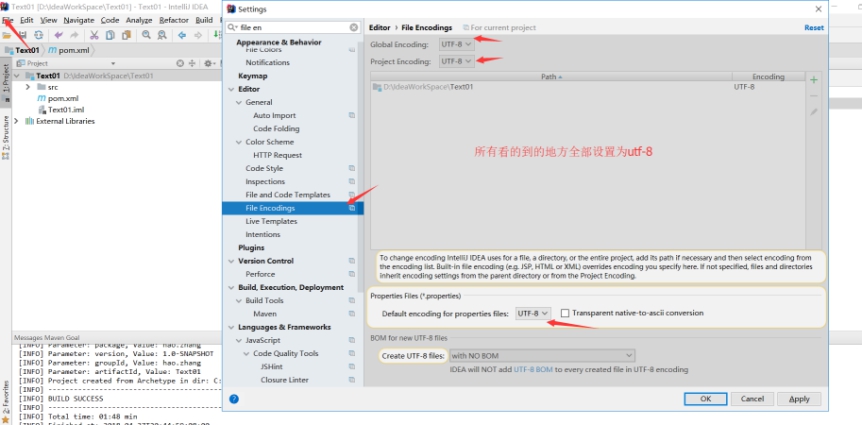
tomcat用来开发和测试javaweb应用程序

nginx用来做负载均衡服务器

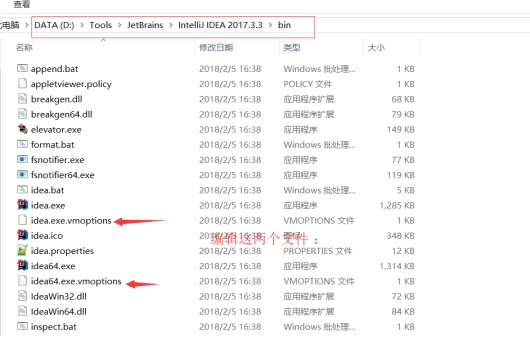
## 使用idea开发web应用程序统一idea码表

**web应用:tomcat8,jdk8或9**

第一步

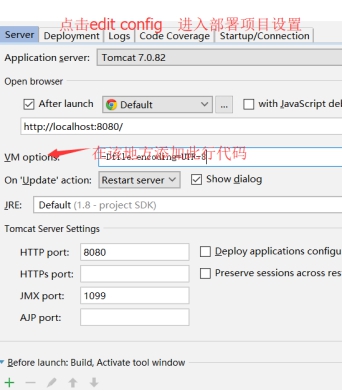


第二步 ： 找到idea的安装目录 ： -Dfile.encoding=UTF-8





第三步 ：：：



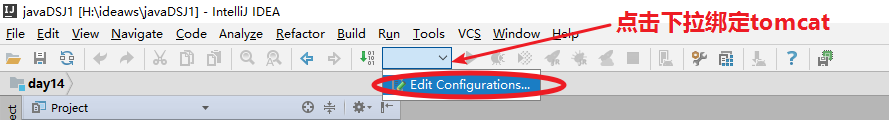
## idea绑定tomcat

### 作用

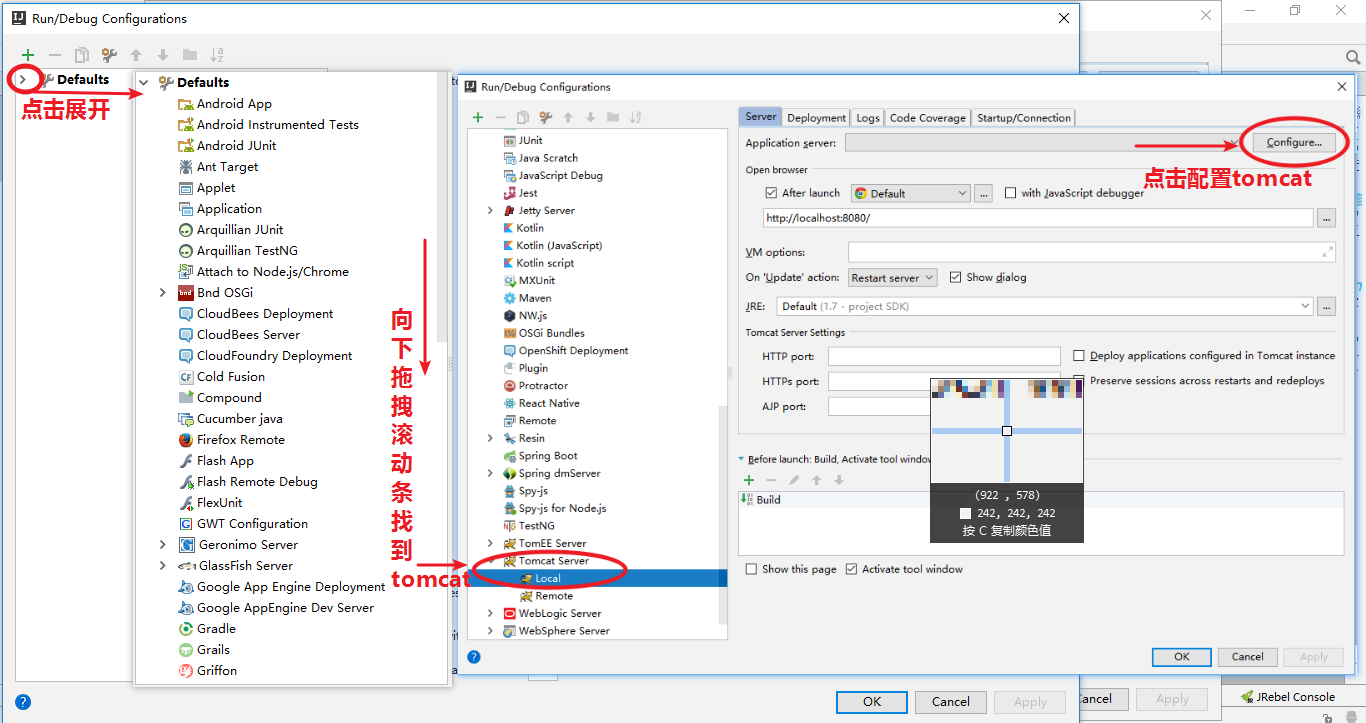
使用idea进行将web项目部署到tomcat上,并以开发方式运行或调试tomcat上的部署项目。

### idea绑定tomcat操作步骤

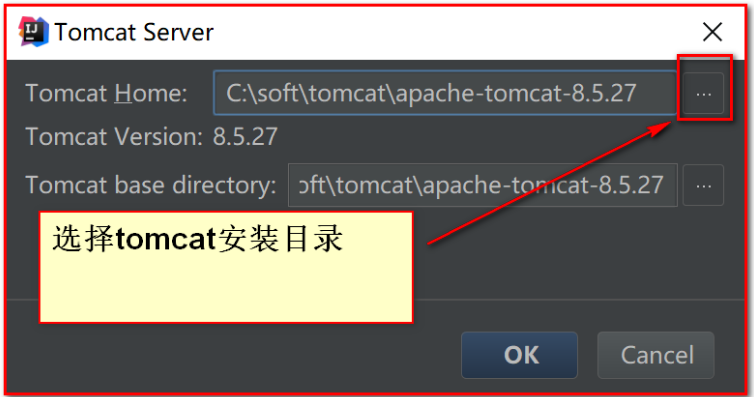
1.点击如下信息



在弹出框,进行如下操作



点击configure…



点击ok,完成配置

# servlet入门

## 2.1 Servlet2.5实现Hello world例子

### 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**：



**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**

1. **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。

### 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

java类需要有main方法才可以运行

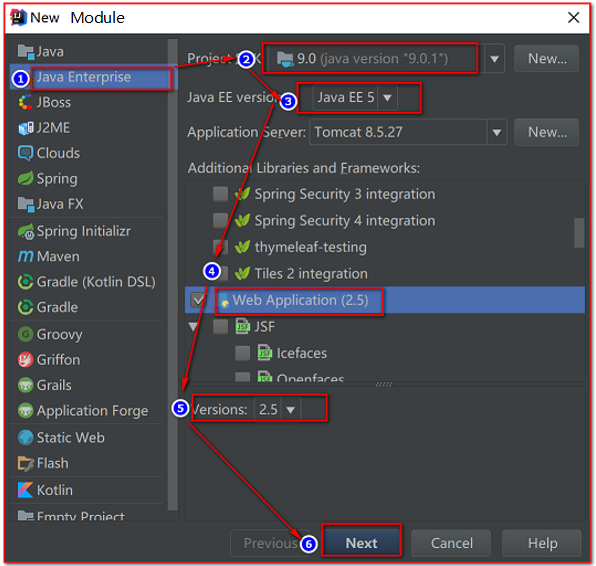
### 代码实现servlet的步骤

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

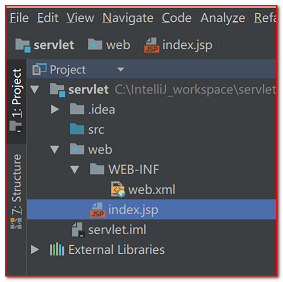
### servlet代码实现

1. **创建web工程**

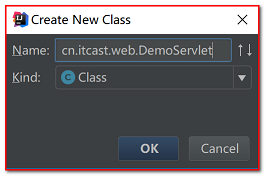
点击导航”file”🡪”new Module”



设置模块名字,之后点击Finish,如下图



1. **在cn.itcast.web包下创建一个类实现Servlet接口**



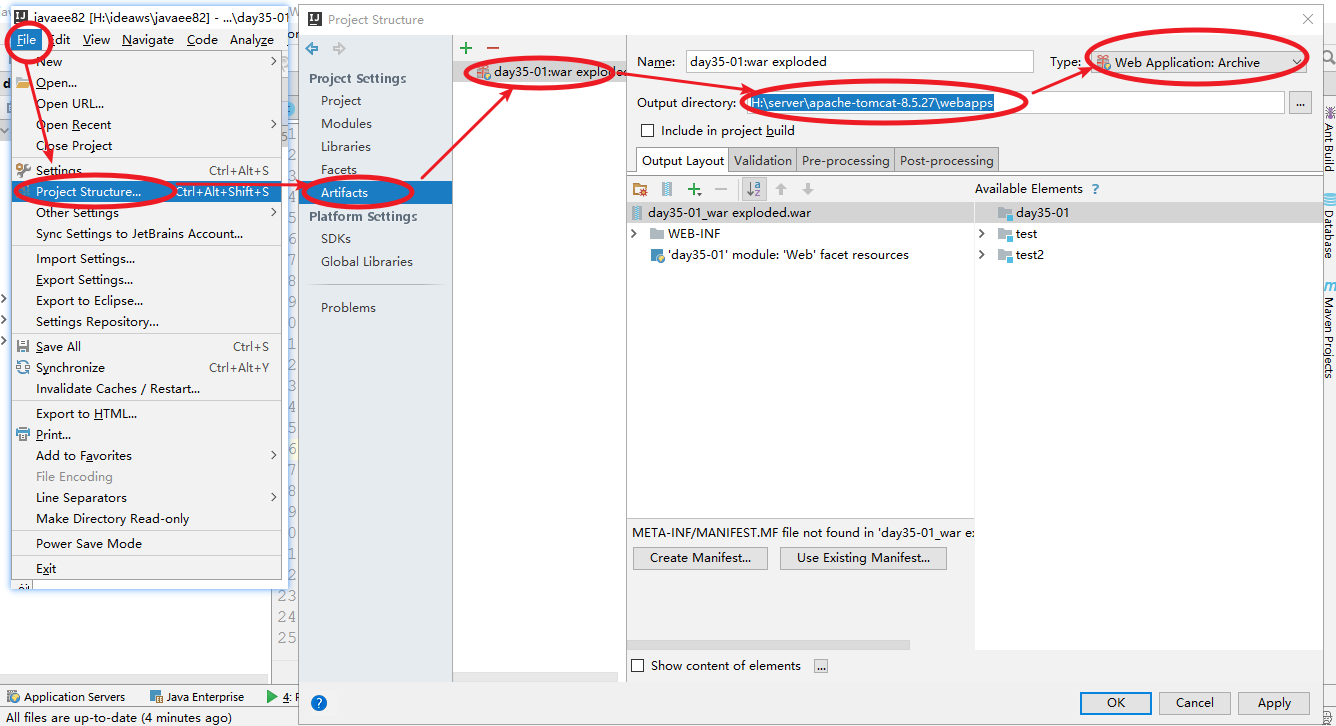
servlet代码：

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() {  
 }  
}

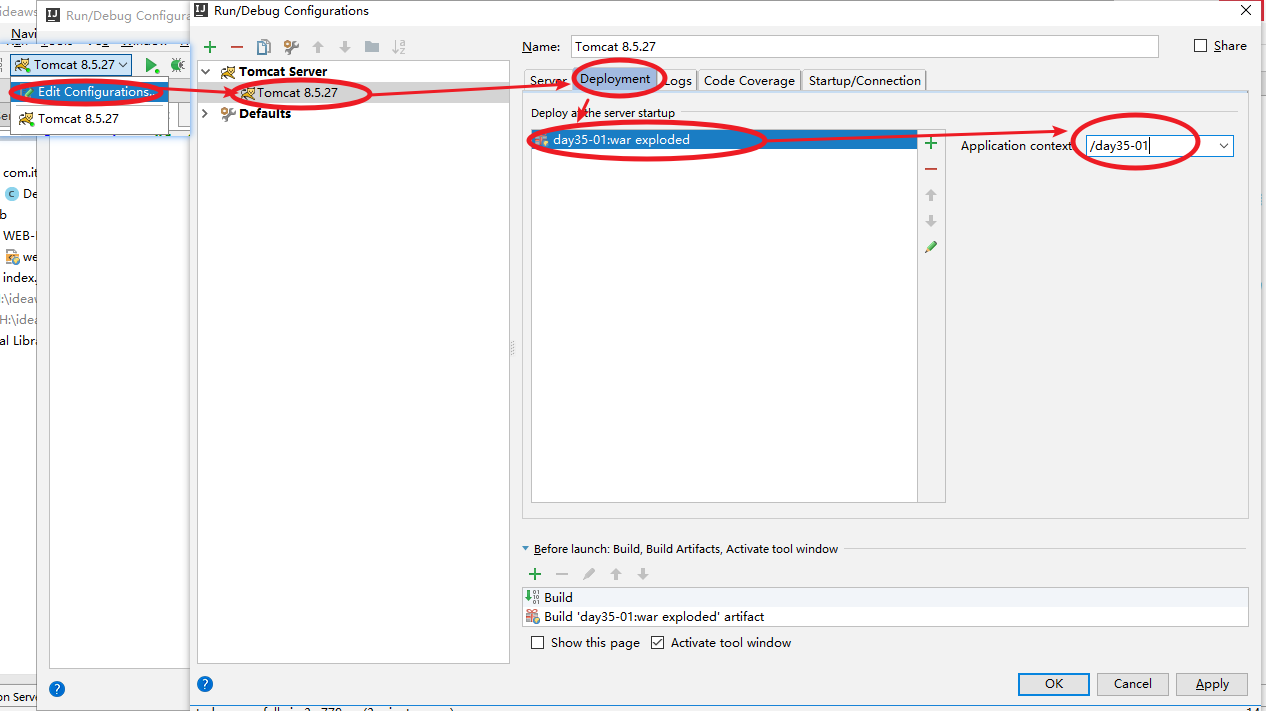
​ Web.xml配置（该文件在web/WEB-INF 文件夹下）：

<?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>

项目部署设置,将当前项目部署到tomcat/webapps目录下设置



设置部署到webapps目录下以资源目录方式存在



## Servlet3.0实现Hello world例子

### Servlet2.5与Servlet3.0的区别

**Servlet3.0相较于Servlet2.5：**

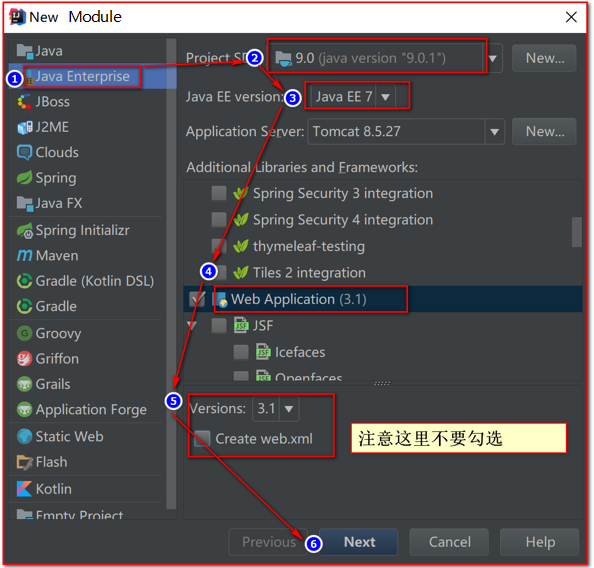
​ **新增了一些注解，简化的javaweb代码开发，可以省略web.xml配置文件** **支持异步处理（多线程技术）**​ **支持可插性特性（书写的代码编译后生成的class文件可以直接部署到其他项目的，自动加载执行）**

### 代码实现Servlet3.0步骤（注解配置servlet演示）

1. **创建JavaEE7工程**
2. 创建servlet，在@WebServlet注解中添加urlPatterns = "/hello"，作为请求路径

### 注解开发servlet代码演示

1. **创建JavaEE7的模块：**



2.注解开发servlet代码演示：

package cn.itcast.web;  
​  
import javax.servlet.\*;  
import javax.servlet.annotation.WebServlet;  
import java.io.IOException;  
​  
//name = "HelloServlet"：servlet名称，相当于web.xml中的<servlet-name>  
//urlPatterns = "/hello"：servlet的访问路径，相当于<url-pattern>  
@WebServlet(name = "HelloServlet",urlPatterns = "/hello")  
public class HelloServlet 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("HelloServlet执行.......");  
   }  
​  
    @Override  
    public String getServletInfo() {  
        return null;  
   }  
​  
    @Override  
    public void destroy() {  
​  
   }  
}  
​

配置tomcat服务器启动测试（配置过程之前已经展示，这里不再重复）

浏览器地址栏输入：<http://localhost:8080/servlet2/hello>

测试成功：



## 2.3 Servlet的体系结构概述

### 什么是Servlet的体系结构

目前我们已经学会创建一个类实现sevlet接口的方式开发Servlet程序，实现Servlet接口的时候，我们必须实现接口的所有方法。但是，在servlet中，真正执行程序逻辑的是service，对于servlet的初始化和销毁，由服务器调用执行，开发者本身不需要关心。因此，有没有一种更加简洁的方式来开发servlet程序呢？

我们先来查阅API回顾Servlet接口：



由上图可知在servlet接口规范下，官方推荐使用继承的方式，继承GenericServlet 或者HttpServlet来实现接口，那么我们接下来再去查看一下这两个类的API：

GenericServlet ：



阅读上图API可知，GenericServlet 是一个类，它简化了servlet的开发，已经提供好了一些servlet接口所需的方法，我们开发者只需要重写service方法即可

我们来使用GenericServlet 创建servlet：

1. 创建一个类
2. 继承GenericServlet
3. 重写service方法

package cn.itcast.web;  
​  
import javax.servlet.GenericServlet;  
import javax.servlet.ServletException;  
import javax.servlet.ServletRequest;  
import javax.servlet.ServletResponse;  
import javax.servlet.annotation.WebServlet;  
import java.io.IOException;  
​  
@WebServlet(name = "GenericDemoServlet",urlPatterns = "/generic")  
public class GenericDemoServlet extends GenericServlet {  
    @Override  
    public void service(ServletRequest servletRequest, ServletResponse servletResponse) throws ServletException, IOException {  
        System.out.println("GenericDemoServlet执行.......");  
   }  
}  
​

虽然，GenericServlet已经简化了servlet开发，但是我们平时开发程序需要按照一种互联网传输数据的协议来开发程序——http协议，因此，sun公司又专门提供了HttpServlet，来适配这种协议下的开发。

HttpServlet：



阅读上图的API可知，继承HttpServlet，我们需要重写doGet、doPost等方法中一个即可，根据Http不同的请求，我们需要实现相应的方法。

我们来使用HttpServlet创建servlet：

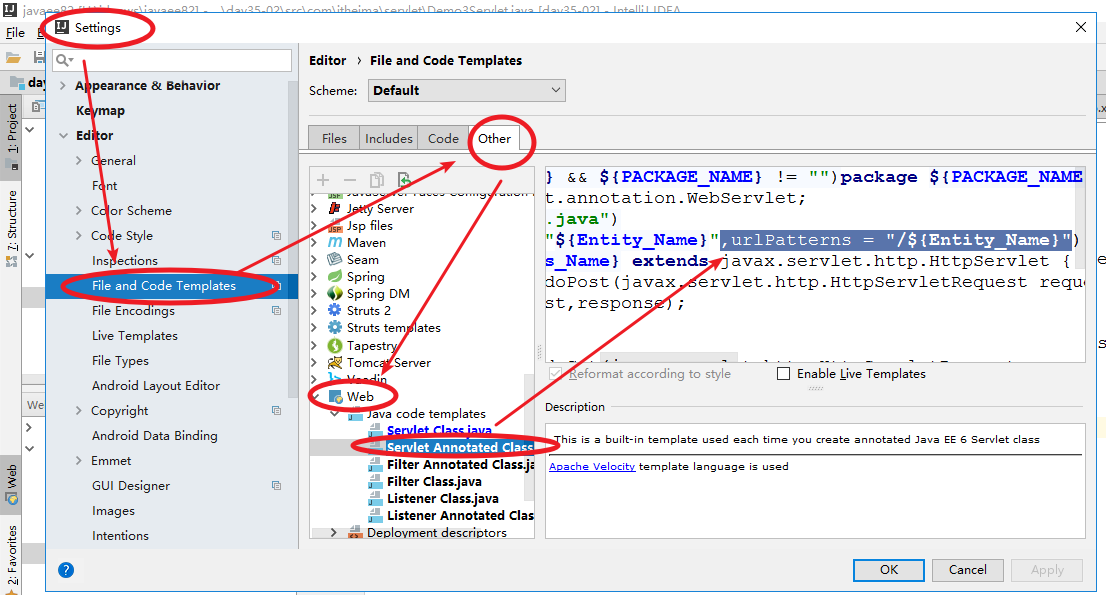
1. 创建一个类
2. 继承HttpServlet
3. 重写doGet方法

package cn.itcast.web;  
​  
import javax.servlet.ServletException;  
import javax.servlet.annotation.WebServlet;  
import javax.servlet.http.HttpServlet;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
import java.io.IOException;  
​  
@WebServlet(name = "HttpDemoServlet",urlPatterns = "/http")  
public class HttpDemoServlet extends HttpServlet {  
    @Override  
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {  
        System.out.println("HttpDemoServlet执行.......");  
   }  
}

通过以上两个API阅读，同学们注意一个细节HttpServlet是GenericServlet的子类，它增强了GenericServlet一些功能，因此，在后期使用的时候，我们都是选择继承HttpServlet来开发servlet程序。

### 使用servlet模板创建HttpServlet

修改模板



代码

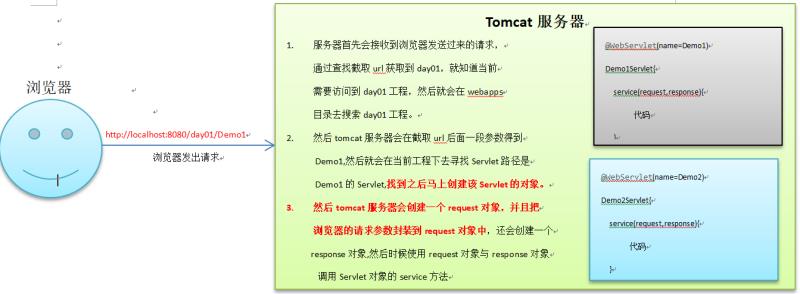
@WebServlet(name = "**${Entity\_Name}**",urlPatterns = "/**${Entity\_Name}**")

模板修改好,再去创建servlet

虽然目前开发servlet的选择继承类，已经确定，但是另一个问题一直还在我们头脑中，那就是我们学的这些浏览器、服务器、servlet等，到底是如何运行的？

那么接下来，我们来看看servlet的运行原理。

## 2.4 servlet运行原理



**总结**

**通过上述流程图我们重点需要掌握如下几个点：**

**1. Servlet对象是由服务器创建**

**2. request与response对象也是由tomcat服务器创建**

**3. request对象封装了浏览器过来的所有请求信息，response对象代表了服务器的响应信息。**