

# Tomcat

## 一 Tomcat 概念

### 1 Servlet 容器

可以运行 Servlet 的环境称为 Servlet 容器

### 2 Web 容器

可以运行 web 应用的环境称为 web 容器，Web 应用可能是某个语言开发的，如 Servlet、php  
Servlet 容器也是 web 容器，只不过是只能运行 Servlet 的 web 容器

### 3 服务器(这里指软件)

能够被外部访问，对外提供服务的就是服务器

3.A WebLogic: *BEA 大型，收费*

3.B WebSphere: *IBM 大型 收费*

3.C Tomcat: *Apache 小型 免费*

## 二 Tomcat 安装

安装路径不能中文

# 1 目录介绍

1.A bin:可执行文件

1.B conf: 配置文件目录

1.C lib: 存放 jar 包目录

1.D logs: 日志目录

1.E temp: 产生临时存放的目录

1.F webapps : 存放 web 应用的目录

1.G work: 运行中产生的工作文件存放的目录

# 2 启动

bin/startup.bat

# 3 关闭

bin/shutdown.bat 或者直接关闭 cmd 窗口

# 4 报错

4.A 闪退

4.A.1 JAVA\_HOME 配置

4.A.2 端口号是否占用，cmd 用 netstat -ano 命令查看哪个 pid 占用

# 5 配置

5.A 配置端口：

目录：conf/server.xml 默认 8080

```

69  -->
70  <Connector port="8080" protocol="HTTP/1.1"
71  connectionTimeout="20000"
72  redirectPort="8443" />
73  <!-- A "Connector" using the shared thread pool--

```

## 三 Web 应用

### 1 部署 web 应用

一个 web 项目称为一个 web 应用，web 应用中可以包含动态资源和静态资源

一个只包含静态资源的 web 应用

创建一个目录，目录名称为 web 应用名称

在此目录中可以添加一些静态资源

#### 1.A 方式 1

1.A.1 \conf\Catalina\localhost 中新建 news.xml 文件，xml 文件名称就是对外访问的路径。(TOMCAT 写自闭标签居然报错)

1.A.2 填入以下内容：html 文件不能在盘符根目录，即不能 D:\1.html

```

ver.xml x web.xml x server.xml x news.xml x
<Context docBase="D:\news"/>

```

1.A.3 当浏览器中输入

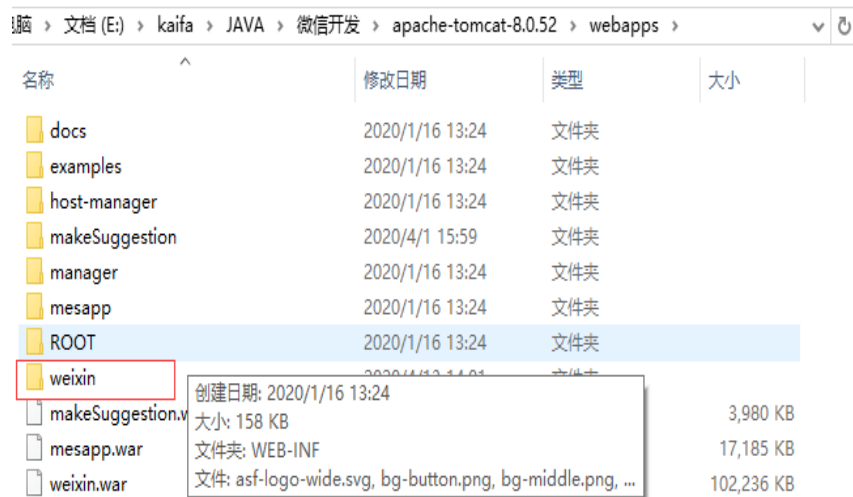
Localhost/news/1.html

原理：localhost/news 就会访问\conf\Catalina\localhost 目录中的 news 文件，找到该文件后读取 docBase 字段值为 D:\news，然后就回去查找 D:\news 下的 1.html

## 1.B 方式 2

1.B.1 将应用拷到 **webapps** 目录下，**web** 目录的名称就是访问的路径

1.B.2 例：本机 [ip/weixin](#) -> [localhost/weixin](#)



## 2 缺省的 Web 应用

### 2.A 方式一

将部署方式一中的 **xml** 文件名称改为 **ROOT**，访问时可以省略该路径

### 2.B 方式二

将部署方式二中的目录名称改为 **ROOT**，访问时可以省略该路径

## 2.C 注意:

2.C.1 当 xml 名称和 webapps 中的目录一样时, xml 名称优先级高

2.C.2 ROOT 应为大写

## 3 Web 应用默认主页

3.A /conf/web.xml 中 依次从上往下查找

```
4688 <!-- deployment descriptor, that also replaces the old  
4689 <!-- here, so be sure to include any of the default values  
4690 <!-- to use within your application.  
4691 <welcome-file-list>  
4692     <welcome-file>index.html</welcome-file>  
4693     <welcome-file>index.htm</welcome-file>  
4694     <welcome-file>index.jsp</welcome-file>  
4695 </welcome-file-list>  
4696  
4697 </web-app>
```

## 4 打包 War 包

4.A 为什么打包

Web 应用直接传输体积大, 打成 war 包体积小, 方便传输  
War 包直接放到 webapps 中会自动解压发布

4.B 怎么打包

先将文件压缩为 zip, 然后改 war 后缀即可

4.C 怎么使用

放到 webapps 中即可

# 四 http 协议

## 1 什么是 http 协议

超文本传输协议,用来规定浏览器(客户端)和服务端之间通信的方式

## 2 http 三个基本原则

2.A.1 基于请求响应模型

2.A.2 一次请求对应一次响应

如果用了输出流，然后刷新缓冲区 `flush` 输出了内容，就认为已经响应了，如果再输出(调用了 `flush`)则不会有效果，比如输出内容再重定向，这是一次请求二次响应了。

2.A.3 请求只能由客户端发出，服务器只能被动等待请求然后做出响应

## 3 http 请求

### 3.A 请求行

3.A.1 请求方式(HTTP 规定了 7 种)

GET/POST/HEAD 等等

3.A.2 请求的资源

也就是 url

3.A.3 http 协议版本

### 3.B 请求头

请求头有多个

3.B.1 `host:`

当前访问的域名

### 3.B.2 *Accept-Language:*

指出请求的语言，如果没有，服务器会返回默认语言

### 3.B.3 *cookie:*

和会话相关的头

### 3.B.4 *referer:*

指从哪个页面跳转过来

## 3.C 实体内容

请求的参数，当请求方式为 **post** 时，请求参数就在请求体中

# 4 http 响应

## 4.A 状态行

### 4.A.1 状态码

一般三位数值，表示服务器对请求的处理结果

常见状态码：

200:成功

404:请求资源不存在

500:代码错误

304、307：使用缓存

302：重定向

### 4.A.2 状态短语

状态码的描述

### 4.A.3 协议版本

## 4.B 响应头

响应头有多个

### 4.B.1 *Content-Type:*

服务器发送数据格式，包括可以指定数据的编码

#### 4.B.2 *Location:*

和请求头重定向相关

#### 4.B.3 *Refresh:*

定时刷新

#### 4.B.4 *Set-cookie*

和 cookie 相关的头

#### 4.B.5 *Expires、cache-control、pragma:*

都是和缓存相关的头

### 4.C 响应实体

就是服务器响应的所有内容