

1 Tomcat 服务器（很熟悉）

1.1 Web 开发概述

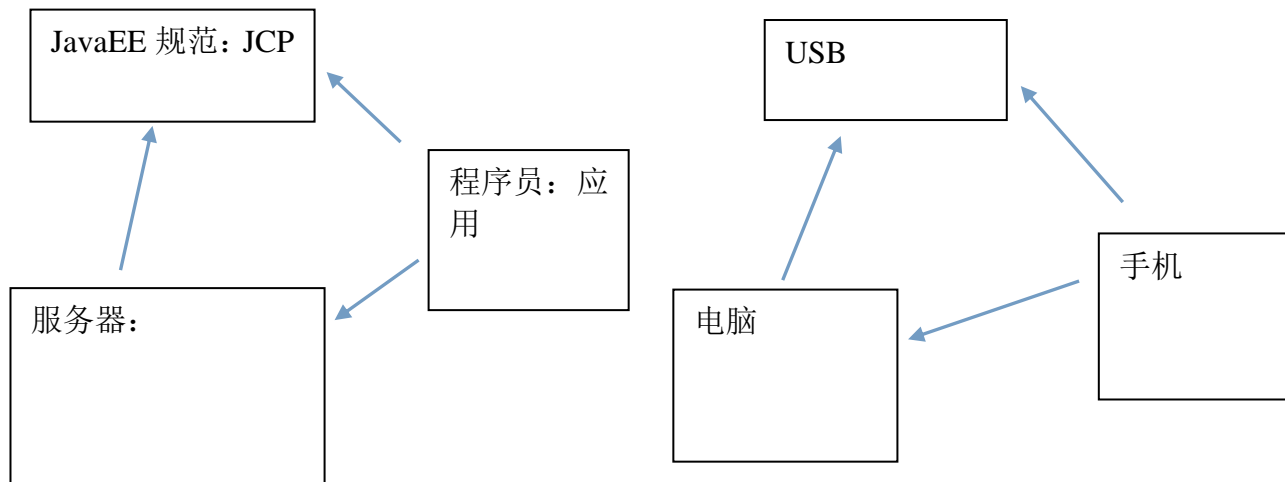
javaSE:

javaEE:13 种

javaME:

JavaEE 规范: 13 种技术的总称。Servlet/Jsp JDBC JNDI JTA...

Tomcat:Servlet/Jsp 容器,轻量级服务器。



1.2 Tomcat 下载

Tomcat 官方站点: <http://tomcat.apache.org>

1.3 Tomcat 的安装与配置

1.3.1 安装:

直接解压到指定目录即可。(注: 目录不要太深; 目录不要有中文或空格)

1.3.2 启动服务器:

F:\apache-tomcat-7.0.52\bin\startup.bat

启动服务器的前题:

配置 JAVA_HOME:

进入到 F:\apache-tomcat-7.0.52\bin>startup.bat 才能执行命令

配置 CATALINA_HOME:

F:\apache-tomcat-7.0.52

startup.bat 启动命令

shutdown.bat 停止命令

如果 tomcat 端口被占用, 解决办法

> 修改 F:\apache-tomcat-7.0.52\conf\server.xml

第 70 行: <Connector port="8081"/>

> 关闭端口号对应的进程:

cmd-->netstat -ano -->找到 pid-->任务管理器-->显示 pid-->关闭进

程

1.3.3 测试服务器是否启动成功

http://localhost:8080

1.3.4 Tomcat 的主要目录：（重要）

bin	可执行文件
conf	配置文件
lib	tomcat运行时需要用的jar文件
logs	tomcat运行产生的日志文件
temp	运行时产生临时文件
webapps	*** 存放我们编写的應用
work	tomcat的工作目录
LICENSE	
NOTICE	
RELEASE-NOTES	
RUNNING.txt	

1.4 标准的 JavaWeb 应用的目录结构（很重要：记住）

应用：

MyApp

1.html

Css

myStyle.css

Js

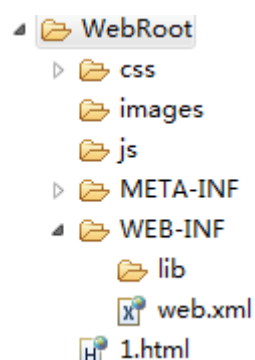
My.js

WEB-INF：注意：固定写法。此目录下的文件不能被外部直接访问。

classes：我们编写的程序代码。.class 文件(class 中的类层次与包的层次相同)

lib：应用需要用的 jar 文件

web.xml：应用的配置信息



1.5 部署应用到 Tomcat 服务器

> 开放目录部署方式

把应用直接复制到 F:\apache-tomcat-7.0.52\webapps 下。

> 把应用打成 war 包。

打 war 包命令：jar -cvf MyApp.war .

把 war 包直接复制到 F:\apache-tomcat-7.0.52\webapps 下，应用自动解压

注：webapps 目录下有几个目录就代表有几个应用。

Tomcat 服务器与 MyEclipse 集成(一定要掌握)

- 1、Tomcat 集成
- 2、创建 web 应用
- 3、部署

URL: 统一资源定位符 (网址)

URI: 统一资源标识符

http://localhost:8080/day08_02/1.html

- [http](#)→协议
- [localhost:8080](#)→主机 IP (端口号)
- [day08_02/1.html](#)→URI (当前应用的资源路径)

1.6 深入熟悉 Tomcat 服务器 (了解)

1.6.1 虚拟目录:

方式一: (不建议使用, 因为需要重启服务器)

在 tomcat 配置文件 server.xml 中, 设置虚拟目录:

虚拟目录映射:

```
<Host name="localhost" appBase="webapps"
      unpackWARs="true" autoDeploy="true">
  <Context path="/jspstudy" docBase="E:\Users\JspStudy\JspWeb"
    reloadable="true"></Context>
```

- docBase:代表的是应用的真实路径。
- path: 网络访问的虚拟目录名
- unpackWARS: 自动解压
- autoDeploy: 自动部署
- reloadable:如果这个属性设为 true, tomcat 服务器在运行状态下会监视在 WEB-INF/classes 和 WEB-INF/lib 目录下 class 文件的改动, 如果监测到有 class 文件被更新的, 服务器会自动重新加载 Web 应用。在开发阶段将 reloadable 属性设为 true, 有助于调试 servlet 和别的 class 文件, 但这样用加重服务器运行负荷, 建议在 Web 应用的发存阶段将 reloadable 设为 false。

注: 配置完成后需要重启服务器。

访问时访问的是虚拟目录, 访问以上网址的 url 为:

<http://localhost:8080/jspstudy>

若 path 改为 path="/aaa", 访问时 url 为:

<http://localhost:8080/aaa>

方式二:

新建 myAPP.xml 文件, 文件名就代表应用的虚拟目录名。

将文件放在 F:\apache-tomcat-7.0.52\conf\Catalina\localhost 目录下

```
<?xml version="1.0" encoding="utf-8"?>
```

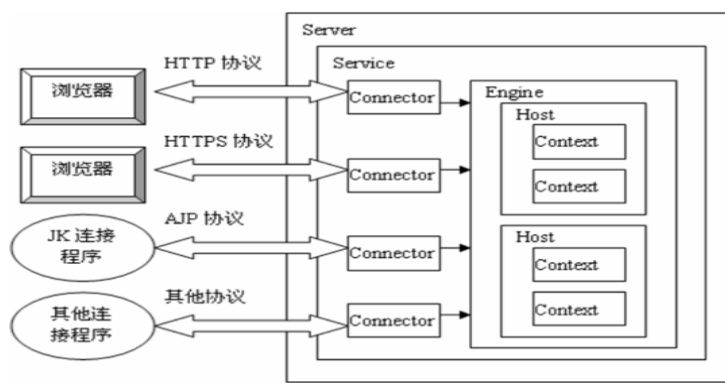
<Context docBase="D:\myAPP"/>

- 文件名 myAPP 为访问的虚拟目录;
 - docBase 为访问的真实目录;
- 注: 配置完成后不需要重启。

访问: <http://localhost:8080/myAPP>

1.6.2 Tomcat 体系结构

- Server 表示整个 tomcat 服务器;
- 一个服务器对应多个服务, 每个服务由 Connector 配置, 可以配置不同的端口和协议;
- 服务会调用 tomcat 引擎(engine)对请求做出处理, 引擎管理 host。
- 一个 host 表示一个主机, 负责调用对应的程序, 处理具体的请求, 发回相应。
- 一个 host 可以映射多个不同的目录。



1.7 配置默认端口、默认应用、默认主页

- A、把 server.xml 中 <Connector port="80" protocol="HTTP/1.1" connectionTimeout="20000" redirectPort="8443" />

浏览器 http 默认端口: 80

- B、默认应用:

默认应用: 把配置的虚拟目录(conf\Catalina\localhost 中的文件)的配置文件名改为 ROOT.xml 即可。

- C、默认主页

修改当前应用 web.xml, 添加以下内容:

```
<?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_3_0.xsd"
  version="3.0">

  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>1.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>

</web-app>
```

