1. web服务器软件

（一）概述

接受用户的请求，处理请求并作出相应。在web服务器软件中，可以部署web项目。让用户通过浏览器来访问这些项目。

（二）Java相关的web服务器软件

1. WebLogic

大型的JavaEE服务器，支持所有的JavaEE规范，oracle公司产品

2. WebSphere

大型的JavaEE服务器，支持所有的JavaEE规范，IBM公司产品

3. JBOSS

大型的JavaEE服务器，支持所有的JavaEE规范，JBOSS公司产品

4. Tomcat

中小型的JavaEE服务器，仅支持少量的JavaEE规范，如servlet/jsp，由Apache基金开源

1. Tomcat
2. Tomcat的使用
3. 安装

将压缩包解压即可，注意在那晃目录不要有中文和空格

1. 启动

bin/startup.bat，双击该批处理文件。

1. 访问

在浏览器中输入：

<http://localhost:8080>访问自己的

在浏览器中输入：

<http://ip:8080>访问ip地址下的

1. 问题

启动时可能由于8080端口号已经被占用而报错，我们需要找到占用的端口号，以及对应的进程，并杀死该进程。

或者修改Tomcat自身的端口号，在conf/server.xml文件中修改默认端口号。一般将tomcat的默认端口号改为80，因为80端口号是http协议的默认端口号，在访问tomcat时，可以不用输入端口号。

1. 项目部署

Tomcat项目部署的方式：  
（1）直接将项目放到webapps目录下即可。可以将项目打包成一个war包，再将war包放置到webapps目录下，war包会自动解压缩。删除项目时直接删除war包即可。

（2）在conf\Catalina\localhost创建任意名称的xml文件，在文件中编写<Context docBase=”D::\\Java” />

docBase:项目存放的路径

虚拟路径：访问项目文件的虚拟目录名为xml文件的名称，如xml文件为a.xml，则该项目的虚拟路径就为D:\\a，通过这个路径来访问该项目

1. Java动态项目结构

项目名称

WEB-INF

web.xml：该项目的核心配置文件

classers目录：字节码文件

lib目录：防止项目依赖的jar包

index.jsp文件为初始显示文件，要想即时地显示添加的文件，需要在Run-Edit Configuration中修改 “On update action”为 Update resources和 “On frame deactivation” 为 Update resources。

1. Servlet
2. 概述

server applet，运行在服务器端的小程序。当浏览器访问服务器端时，服务器端需要通过Java类来实现动态的逻辑，这个Java类要想被Tomcat识别，就要遵守一定的规则，即接口规范，这个接口就是Servlet。该接口定义了Java类被浏览器访问到（Tomcat识别）的规则。

1. 使用
2. 定义一个类，实现Servlet接口
3. 实现接口中的抽象方法
4. 配置Servlet
5. 配置Servlet

所谓的配置Servlet，是配置Servlet对应的Java类以及URL。浏览器要通过URL访问服务器，因此我们要提前设置好Servlet的URL以及Java类。配置Servlet是在web.xml中进行配置。

设置Java类，servlet-name为servlet的名字，servlet-class为该servlet对应的Java类，需要写全类名，即从项目名开始。

<servlet>

<servlet-name>demo1</servlet-name>

<servlet-class>cn.web.servlet.ServletDemo</servlet-class>

</servlet>

配置URL，url-pattern为资源位置

<servlet-mapping>

<servlet-name>demo1</servlet-name>

<url-pattern>/demo</url-pattern>

</servlet-mappint>

1. Servlet执行原理
2. 当服务器接收到客户端浏览器的请求后，会解析请求URL路径，获取访问的Servlet的资源路径。
3. 查找web.xml文件，是否有对应的<url-pattern>标签体内容
4. 有对应的url，则可以通过servlet-name找到servlet-class的全类名
5. 找到全类名后，Tomcat会将字节码文件加载进内存，并且创建其对象。
6. 调用该类的方法
7. Servlet的方法
8. init方法

创建对象，只执行一次。可以通过配置文件来设置servlet的创建时机，默认情况下是第一次被访问时，Servlet被创建。

在<servlet>标签下配置：  
第一次被访问时创建，设置<load-on-startup>的值为负数

<load-on-startup>-5</load-on-startup>

在服务器启动时创建，设置<load-on-startup>的值为0或者正整数：  
<load-on-startup>0</load-on-startup>

1. service方法

提供服务，可执行多次。每次访问Servlet时，service方法都会被调用一次。

1. destroy方法

销毁对象，只执行一次。Servlet对象被销毁时执行，当服务器关闭时，先执行destroy方法，用于释放资源，再销毁Servlet对象。

1. 注解配置

在Servlet3.0之后，支持了注解配置，不需要用web.xml进行配置了。同样是定义类实现servlet接口，在类上使用@WebServlet注解，进行配置：

@WebServlet(“servlet资源路径”)

1. 注意事项

Tomcat真正访问的是 Tomcat部署的Web项目，对应着web目录下的所有资源，WEB-INF目录下的资源是不能被浏览器直接访问的。

1. Servlet的体系结构
2. 结构

GenericServlet抽象类继承Servlet接口，HttpServlet抽象类又继承了GenericServlet抽象类。

Servlet接口

GenericServlet 抽象类

HttpServlet 抽象类

1. GenericServlet抽象类

将Servlet接口中的其他方法做了默认的空实现，只将service()方法作为抽象方法，将来在定义Servlet类时，可以继承GenericServlet抽象类，实现service()方法即可，其他的方法若有需要依然可以重写。

1. HttpServlet抽象类

继承了GenericServlet抽象类，该类是对http协议的一种封装，简化操作。我们要做的是定义类继承HttpServlet，并重写doGet() / doPost()方法。

1. Servlet相关配置
2. urlpattern

servlet的访问路径，一个servlet可以定义多个访问路径：

@WebServlet({“/a”, “/b”, “/c”})

路径的定义规则：  
/xx ：单层路径

/xx/xx : 多层路径，目录结构

\*.a : \* 为通配符，后面的a也为自定义的扩展名