## 基本概念

### 1.1、WEB开发的相关知识

　　WEB，在英语中web即表示网页的意思，它用于表示Internet主机上供外界访问的资源。  
　　Internet上供外界访问的Web资源分为：

1. **静态web资源（如html 页面）：指web页面中供人们浏览的数据始终是不变。**
2. **动态web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**。

　　静态web资源开发技术：Html  
　　常用动态web资源开发技术：JSP/Servlet、ASP、PHP等  
　　在Java中，动态web资源开发技术统称为Javaweb。

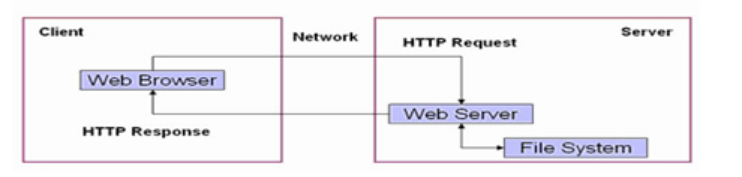
### 1.2、WEB应用程序

　　WEB应用程序指供浏览器访问的程序，通常也简称为web应用。例如有a.html 、b.html…..多个web资源，这多个web资源用于对外提供服务，此时应把这多个web资源放在一个目录中，以组成一个web应用（或web应用程序）  
　　一个web应用由多个静态web资源和动态web资源组成，如:html、css、js文件，Jsp文件、java程序、支持jar包、配置文件等等。  
　　**Web应用开发好后，若想供外界访问，需要把web应用所在目录交给web服务器管理，这个过程称之为虚似目录的映射**

### 1.3、WEB发展史

　　WEB发展的两个阶段：静态、动态

### 1.4、静态WEB

　　\*htm、\*html，这些是网页的后缀，如果现在在一个服务器上直接读取这些内容，那么意味着是把这些网页的内容通过网络服务器展现给用户。整个静态WEB操作的过程图如下：

　　在静态WEB程序中，客户端使用WEB浏览器（IE、FireFox等）经过网络(Network)连接到服务器上，使用HTTP协议发起一个请求（Request），告诉服务器我现在需要得到哪个页面，所有的请求交给WEB服务器，之后WEB服务器根据用户的需要，从文件系统（存放了所有静态页面的磁盘）取出内容。之后通过WEB服务器返回给客户端，客户端接收到内容之后经过浏览器渲染解析，得到显示的效果。

**静态WEB中存在以下几个缺点：**

**1、Web页面中的内容无法动态更新，所有的用户每时每刻看见的内容和最终效果都是一样的。**

　　为了可以让静态的WEB的显示更加好看，可以加入了JavaScript以完成一些页面上的显示特效，但是这些特效都是在客户端上借助于浏览器展现给用户的，所以在服务器上本身并没有任何的变化。

　　实现静态WEB客户端动态效果的手段：

* JavaScript
* VBScript

　　在实际的开发中JavaScript使用得最多。

**2、静态WEB无法连接数据库，无法实现和用户的交互。**

　　使用数据库保存数据是现在大多数系统的选择，因为数据库中可以方便地管理数据，增删改查操作可以使用标准的SQL语句完成。

### 1.5、动态WEB

　　所谓的动态不是指页面会动，主要的特性的是：“WEB的页面展示效果因时因人而变”，而且动态WEB具有交互性，WEB的页面的内容可以动态更新。整个动态WEB操作的过程图如下：

　　动态WEB中，程序依然使用客户端和服务端，客户端依然使用浏览器（IE、FireFox等），通过网络(Network)连接到服务器上，使用HTTP协议发起请求（Request），现在的所有请求都先经过一个WEB Server Plugin（服务器插件）来处理，此插件用于区分是请求的是静态资源(\*.htm或者是\*.htm)还是动态资源。

　　如果WEB Server Plugin发现客户端请求的是静态资源(\*.htm或者是\*.htm)，则将请求直接转交给WEB服务器，之后WEB服务器从文件系统中取出内容，发送回客户端浏览器进行解析执行。

　　如果WEB Server Plugin发现客户端请求的是动态资源（\*.jsp、\*.asp/\*.aspx、\*.php），则先将请求转交给WEB Container(WEB容器)，在WEB Container中连接数据库，从数据库中取出数据等一系列操作后动态拼凑页面的展示内容，拼凑页面的展示内容后，把所有的展示内容交给WEB服务器，之后通过WEB服务器将内容发送回客户端浏览器进行解析执行。

### 1.6、动态WEB应用的实现手段

　　动态WEB现在的实现手段非常多，较为常见的有以下几种：

* Microsoft ASP、ASP.NET
* PHP
* JAVA Servlet/JSP

#### 1、Microsoft ASP、ASP.NET

　　微软公司动态WEB开发是比较早的，而且最早在国内最流行的是ASP。ASP就是在HTML语言之中增加了VB脚本，但是标准的开发应用应该是使用ASP+COM，但是实际情况来看，在开发ASP的时候基本上都在一个页面中写上成百上千的代码，页面代码极其混乱。

　　ASP本身有开发平台的限制：Windows+IIS+SQL Server/Access，ASP只能运行在Windows操作系统上，ASP现在基本上已经淘汰，现在基本上都是使用ASP.NET进行开发，ASP.NET在性能有了很大的改善，而且开发迅速，但是依然受限于平台。ASP.NET中主要是使用C#语言。

#### 2.PHP

　　PHP开发速度很快，功能强大，跨平台(平台指的就是运行的操作系统)，而且代码也简单。

#### 3.Servlet/JSP

　　这是SUN公司(SUN现在已经被Oracle公司收购)主推的B/S架构的实现语言，是基于JAVA语言发展起来的，因为JAVA语言足够简单，而且很干净。

　　Servlet/JSP技术的性能也是非常高的，不受平台的限制，各个平台基本上都可以使用。而且在运行中是使用多线程的处理方式，所以性能非常高。

　　SUN公司最早推出的WEB技术推出的是Servlet程序，Servlet程序本身使用的时候有一些问题，所有的程序是采用JAVA代码+HTML的方式编写的，即，要使用JAVA输出语句，一行一行地输出所有的HTML代码，之后，SUN公司受到了ASP的启发，发展出了JSP(Java Server Page)，JSP某些代码的编写效果与ASP是非常相似的。这样可以很方便地使一些ASP程序员转向JSP的学习，加大市场的竞争力度。

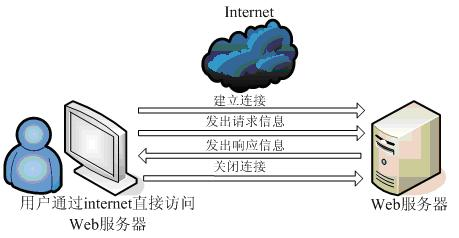
## 二、WEB服务器

### 2.1、WEB服务器简介

　　1、Web服务器是指驻留于因特网上某种类型计算机的程序，是可以向发出请求的浏览器提供文档的**程序**。当Web浏览器（客户端）连到服务器上并请求文件时，服务器将处理该请求并将文件反馈到该浏览器上，附带的信息会告诉浏览器如何查看该文件（即文件类型）。



　　2、服务器是一种被动程序：只有当Internet上运行在其他计算机中的浏览器发出请求时，服务器才会响应。



# WEB环境搭建

## 1.Web服务器

* Web服务器主要用来接收客户端发送的请求和响应客户端请求。
* 常见的JavaWeb服务器：
* **Tomcat（Apache）**：当前应用最广的JavaWeb服务器；
* JBoss（Redhat红帽）：支持JavaEE，应用比较广EJB容器 –> SSH轻量级的框架代替
* GlassFish（Orcale）：Oracle开发JavaWeb服务器，应用不是很广；
* Resin（Caucho）：支持JavaEE，应用越来越广；
* Weblogic（Orcale）：要钱的！支持JavaEE，适合大型项目；
* Websphere（IBM）：要钱的！支持JavaEE，适合大型项目；

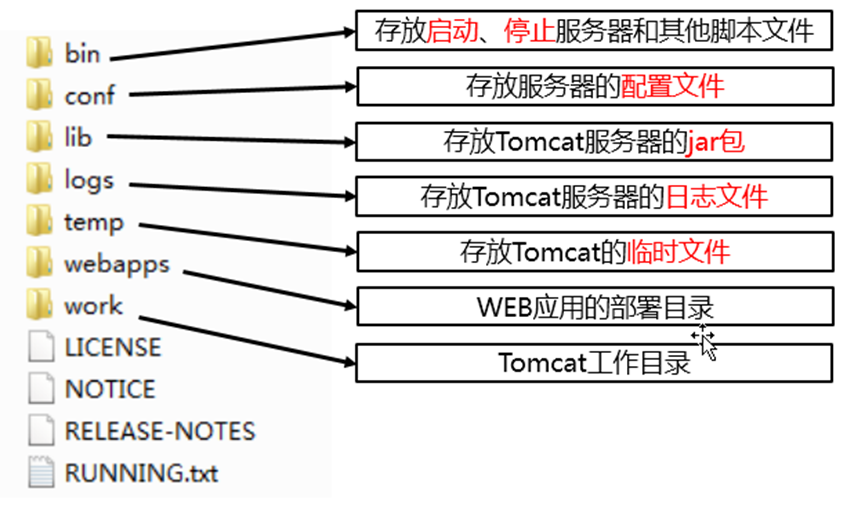
## 2.Tomcat服务器

### 2.1 Tomcat简介

* 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，成为目前比较流行的Web 应用服务器。

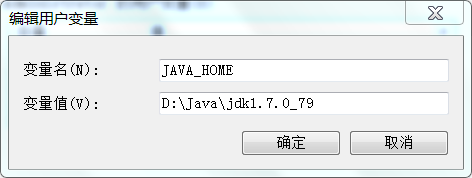
### 2.2 Tomcat安装、配置、启动

①解压apache-tomcat-8.5.13-windows-x64到**非中文无空格**目录中



* bin：该目录下存放的是二进制可执行文件，如果是安装版，那么这个目录下会有两个exe文件：tomcat6.exe、tomcat6w.exe，前者是在控制台下启动Tomcat，后者是弹出GUI窗口启动Tomcat；如果是解压版，那么会有startup.bat和shutdown.bat文件，startup.bat用来启动Tomcat，但需要先配置JAVA\_HOME环境变量才能启动，shutdawn.bat用来停止Tomcat；
* conf：这是一个非常非常重要的目录，这个目录下有四个最为重要的文件：
* server.xml：配置整个服务器信息。例如修改端口号
* tomcat-users.xml：存储tomcat用户的文件，这里保存的是tomcat的用户名及密码，以及用户的角色信息。可以按着该文件中的注释信息添加tomcat用户，然后就可以在Tomcat主页中进入Tomcat Manager页面了；
* web.xml：部署描述符文件，这个文件中注册了很多MIME类型，即文档类型。这些MIME类型是客户端与服务器之间说明文档类型的，如用户请求一个html网页，那么服务器还会告诉客户端浏览器响应的文档是text/html类型的，这就是一个MIME类型。客户端浏览器通过这个MIME类型就知道如何处理它了。当然是在浏览器中显示这个html文件了。但如果服务器响应的是一个exe文件，那么浏览器就不可能显示它，而是应该弹出下载窗口才对。MIME就是用来说明文档的内容是什么类型的！
* context.xml：对所有应用的统一配置，通常我们不会去配置它。
* lib：Tomcat的类库，里面是一大堆jar文件。如果需要添加Tomcat依赖的jar文件，可以把它放到这个目录中，当然也可以把应用依赖的jar文件放到这个目录中，这个目录中的jar所有项目都可以共享之，但这样你的应用放到其他Tomcat下时就不能再共享这个目录下的jar包了，所以建议只把Tomcat需要的jar包放到这个目录下；
* logs：这个目录中都是日志文件，记录了Tomcat启动和关闭的信息，如果启动Tomcat时有错误，那么异常也会记录在日志文件中。
* temp：存放Tomcat的临时文件，这个目录下的东西可以在停止Tomcat后删除！
* webapps：存放web项目的目录，其中每个文件夹都是一个项目；如果这个目录下已经存在了目录，那么都是tomcat自带的项目。其中ROOT是一个特殊的项目，在地址栏中没有给出项目目录时，对应的就是ROOT项目。<http://localhost:8080/examples，进入示例项目。其中examples>就是项目名，即文件夹的名字。
* work：运行时生成的文件，最终运行的文件都在这里。通过webapps中的项目生成的！可以把这个目录下的内容删除，再次运行时会生再次生成work目录。当客户端用户访问一个JSP文件时，Tomcat会通过JSP生成Java文件，然后再编译Java文件生成class文件，生成的java和class文件都会存放到这个目录下。
* LICENSE：许可证。
* NOTICE：说明文件。

②如果双击startup.bat后窗口一闪而过，请查看JAVA\_HOME是否配置正确



③新建环境变量CATALINA\_HOME=解压目录

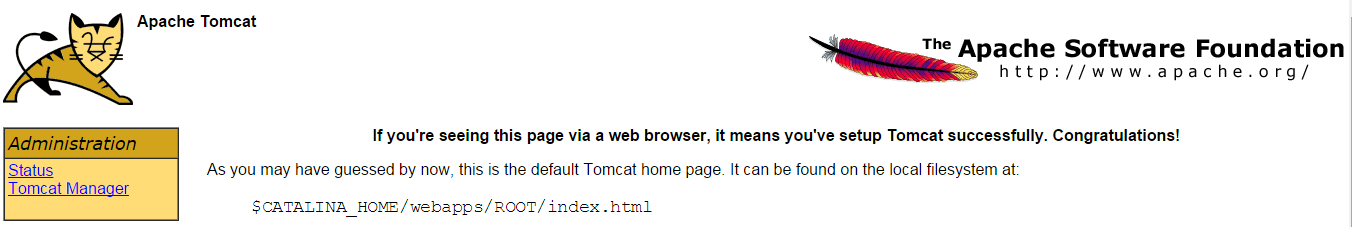


④在Path环境变量中加入Tomcat解压目录\bin目录



⑤在命令行中运行catalina run或者 startup启动Tomcat服务器，在浏览器地址栏访问如下地址进行测试

**http://localhost:8080**

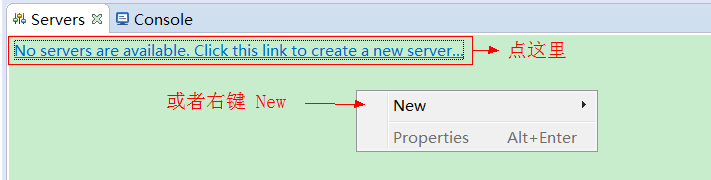


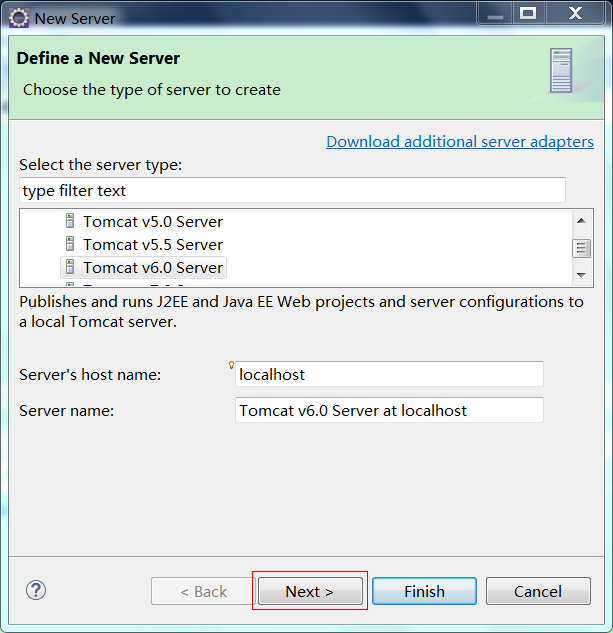
⑥如果启动失败，提示端口号被占用，则将默认的8080端口修改为其他未使用的值，例如10086等。

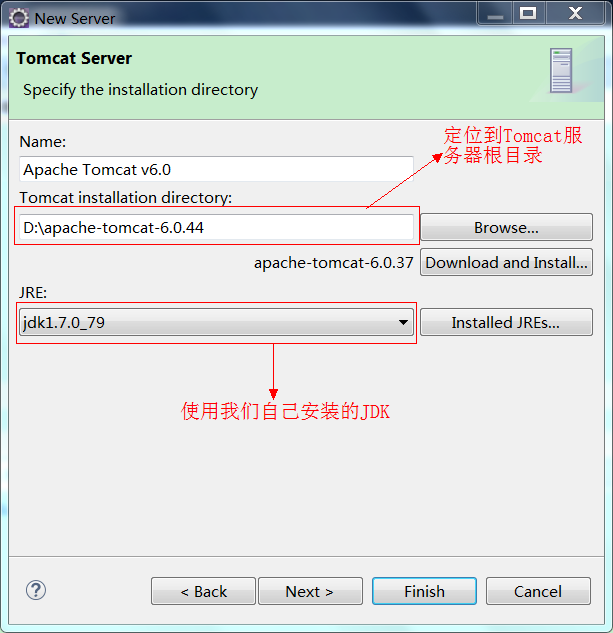
打开：解压目录\conf\server.xml，找到第一个Connector标签，修改port属性

### 2.3在Eclipse中创建Tomcat镜像

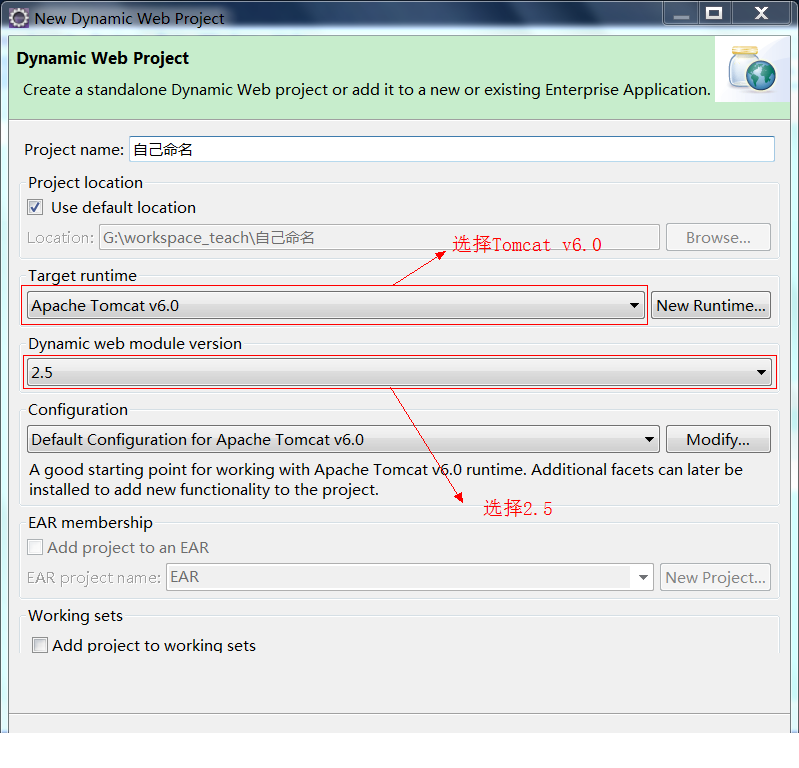
①在Servers视窗里点击超链接或在空白处右键→New







②创建**动态**Web工程进行测试



[1]在WebContent目录下创建index.jsp，加入如下代码

<%@page import="java.util.Date"%>

<%=new Date() %>

[2]在index.jsp上点右键：Run as→Run on Server查看运行结果

③说明：关联Tomcat镜像时，Eclipse会从本地Tomcat中复制信息及文件，之后二者的配置信息就没有关系了，其中任何一个的配置信息发生变化都不会自动同步到另外一个。