WEB，在英语中表示网页的意思，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资源分为静态资源和动态资源。

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。

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。

常用动态web资源开发技术：JSP/Servlet、ASP.net、PHP等

在Java中，动态web资源开发技术统称为Java web开发。

C/S架构

B/S架构 浏览器/服务端

Web应用

JSP+Servlet

# 一 tomcat基本使用

学习web开发，需要先安装一台web服务器，然后再在web服务器中提供相应的web资源，供用户使用浏览器访问。

## 1 常见的web服务软件

WebLogic：Oracle公司，收费，全面支持javaEE规范

JBoss：RedHat ，收费，全面支持javaEE规范

Tomcat：Apache，免费，部分支持javaEE规范

## 2 下载安装

获取Tomcat安装程序包

tar.gz文件是Linux操作系统下的压缩版本

exe文件是Windows系统下的安装版本

zip文件是Windows系统下的压缩版本

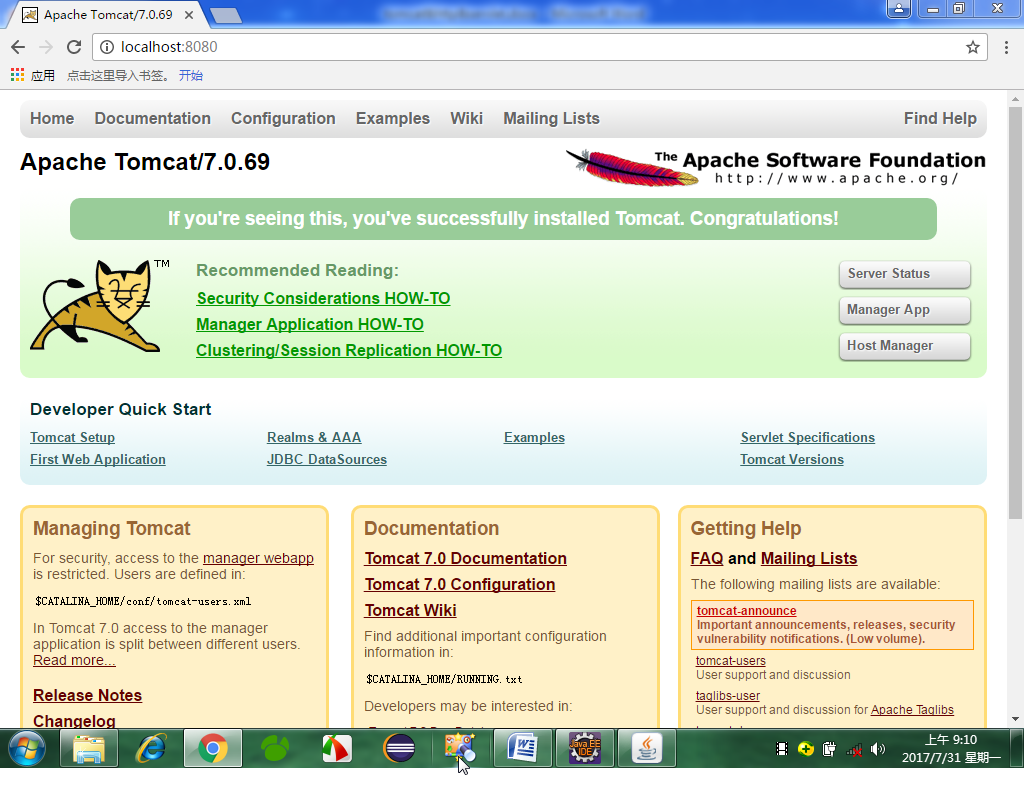
本教学过程中使用zip版的tomcat8，将其解压后放在合适的目录下，路径中不要包含中文

## 3 启动关闭

启动：Tomcat安装目录/bin/stratup.bat

关闭：Tomcat安装目录/bin/shutdown.bat

启动tomcat服务后，在浏览器输入<http://localhost:8080/>，如果显示如下界面说明服务器启动正常



## 4 常见问题

1) JAVA\_HOME

没有JAVA\_HOME环境变量，启动tomcat服务器，闪退

2) 8080端口占用

Tomcat使用8080端口

修改tomcat端口或者关闭占用8080端口的程序

修改端口号位置：

<Connector connectionTimeout="**20000**" port="**8080**" protocol="**HTTP/1.1**" redirectPort="**8443**" />

3)CATALINA\_HOME引起的问题

安装版的tomcat会自动增加CATALINA\_HOME环境变量，如果卸载tomcat，不会卸载该环境变量。再次安装tomcat时，如果和以前的路径不一样，启动tomcat就会出问题

解决方案：直接删除该环境变量

## 5 tomcat主要目录结构

Bin ：tomcat相关的命令和脚本文件

Conf ：tomcat服务器相关的配合文件，server.xml核心配置

Lib ：运行tomcat需要的jar文件

Webapps：存放web应用，即供外界访问的web资源

Work：tomcat工作目录，存放jsp生成的class文件

# 二 web应用开发

## 1 web应用目录结构

Web应用目录

--页面相关的资源，如html、jsp、js、css等文件**外界可以直接访问**

--META-INF

--WEB-INF**该目录下的文件外界无法非法直接访问，由web服务器负责调用**

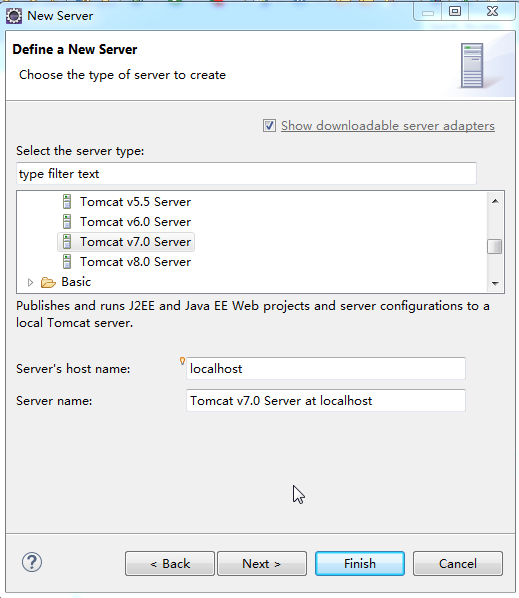
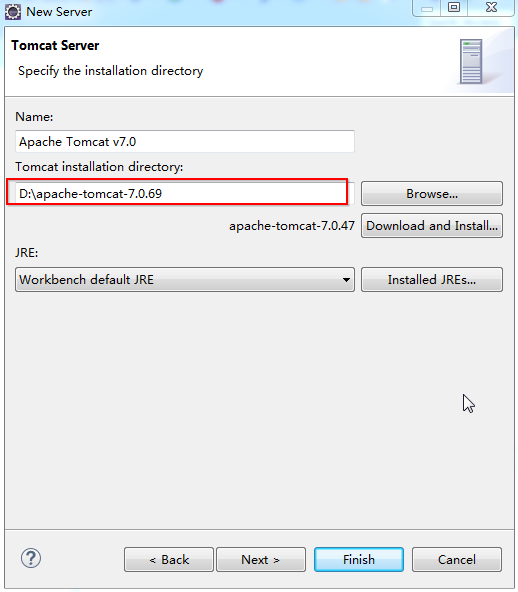
--classes：web应用程序的字节码文件

--lib：存放运行web程序需要的jar文件

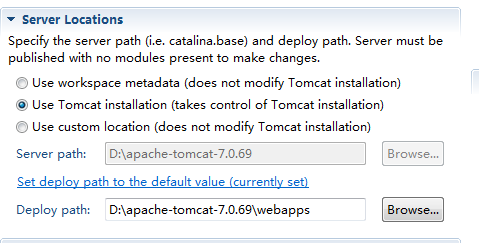
--web.xml: web应用的配置文件

## 2 Servlet体验

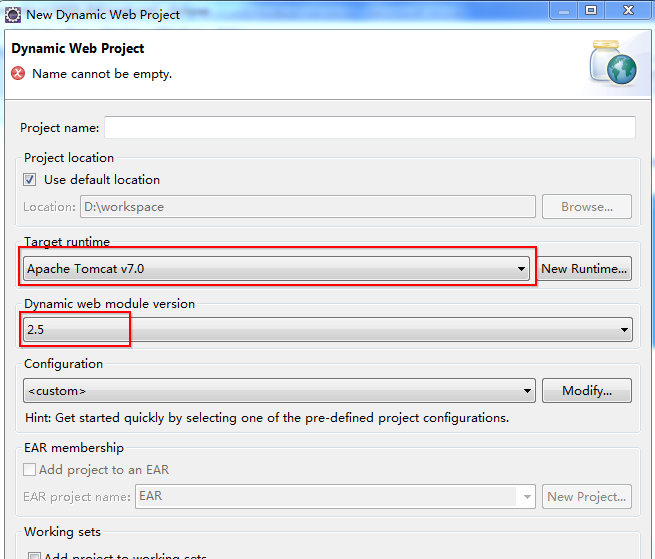
1）新建server

修改web应用的发布目录：



2） 新建web项目



3） 新建Servlet类

4） 运行

Web应用程序要发布到tomcat服务器下

<http://tomcat服务器/web应用名称/>资源名称

http://localhost:8080/W09\_Servlet1/HelloServlet

# 三 Http入门

客户端连上web服务器后，若想获得web服务器中的某个web资源，需遵守一定的通讯格式，HTTP协议用于定义客户端与web服务器通迅的格式

HTTP是hypertext transfer protocol（超文本传输协议）的简写，它是TCP/IP协议的一个应用层协议，用于定义WEB浏览器与WEB服务器之间交换数据的过程。

HTTP协议的版本：HTTP/1.0、HTTP/1.1

## 1 Http请求

客户端连上服务器后，向服务器请求某个web资源，称之为客户端向服务器发送了一个HTTP请求。一个完整的HTTP请求包括如下内容：一个请求行、若干请求头、以及实体内容，其中的一些请求头和实体内容都是可选的，请求头和实体内容之间要用空行隔开。

如下所示 ：

GET /W09\_Servlet1/HelloServlet HTTP/1.1 --请求行

Host: localhost:8080 -- 请求头 数据是key-value形式

Connection: keep-alive

Cache-Control: max-age=0

Upgrade-Insecure-Requests: 1

User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.116 Safari/537.36

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8

Accept-Encoding: gzip, deflate, sdch

Accept-Language: zh-CN,zh;q=0.8

空行

Post方式提交的数据 --请求体

## 2 请求方式

请求行中的GET称之为请求方式，请求方式有：*POST、GET、HEAD、OPTIONS、DELETE、PUT等*

常用的有：POST、GET

不管POST或GET，都用于向服务器请求某个WEB资源，这两种方式的区别主要表现在数据传递上，客户端通过这两种方式都可以带一些数据给服务器。

1）get请求

如请求方式为GET方式，则可以在请求的URL地址后以?的形式带上交给服务器的数据，多个数据之间以&进行分隔。如：

http://localhost:8080/W09\_Servlet1/HelloServlet?name=zhangsan&age=12

GET方式的特点：在URL地址后附带的参数是有限制的，其数据容量不能超过1K

2）post请求

如请求方式为POST方式，则可以在请求的实体内容中向服务器发送数据。

http://localhost:8080/W09\_Servlet1/HelloServlet

提交的数据放在请求体中：

name=zhangsan&age=13

Post方式的特点：*传送的数据量无限制。*

## 3 Http响应

一个HTTP响应代表服务器向客户端回送的数据，它包括：

一个响应行、若干响应头、以及实体内容 ，其中的一些响应头和实体内容都是可选的，响应头和实体内容之间要用空行隔开。

HTTP/1.1 200 OK --响应行

Server: Apache-Coyote/1.1 --响应头

Content-Length: 24

Date: Mon, 31 Jul 2017 02:57:09 GMT

空行

服务端返回浏览器显示的数据 --响应体

## 4 响应行

格式： *HTTP版本号　状态码　原因叙述*

举例：*HTTP/1.1 200 OK*

状态码用于表示服务器对请求的处理结果，它是一个三位的十进制数。响应状态码分为5类，如下所示：

|  |  |
| --- | --- |
| 100～199 | 表示成功接收请求，要求客户端继续提交下一次请求才能完成整个处理过程 |
| 200～299 | 表示成功接收请求并已完成整个处理过程，常用200 |
| 300～399 | 为完成请求，客户需进一步细化请求。例如，请求的资源已经移动一个新地址，常用302、307和304 |
| 400～499 | 客户端的请求有错误，常用404 |
| 500～599 | 服务器端出现错误，常用 500 |

# 四 Servlet开发

进行java web应用程序开发的技术

Web服务器收到客户端的http请求，会针对每一次请求，分别创建一个用于代表请求的request对象、和代表响应的response对象。

request和response对象即然代表请求和响应，如果要获取客户机提交过来的数据，只需要找request对象。要向客户端输出数据，只需要找response对象。

## 1 HttpServletRequest

HttpServletRequest对象代表客户端的请求，当客户端通过HTTP协议访问服务器时，HTTP请求头中的所有信息都封装在这个对象中，开发人员通过这个对象中的方法，可以获得请求相关信息。

1)获取请求参数

//读取get方式提交的数据

//获取请求参数

//如果参数中有中文，name=%E5%BC%A0%E4%B8%89&password=123

String queryString = request.getQueryString();

//中文乱码问题

queryString = URLDecoder.*decode*(queryString, "utf-8");

System.***out***.println(queryString);

//读取post方式提交的数据

ServletInputStream inputStream = request.getInputStream();

**byte**[] buff = **new** **byte**[1024];

**int** len = inputStream.read(buff);

String string = **new** String(buff, 0, len);

string = URLDecoder.*decode*(string, "utf-8");

System.***out***.println(string);

统一方式：

//根据表单中输入控件的name属性值，获取表单参数

//get和post方式都可以用该方法获取表单参数

//tomcat8下，get方式下，使用getParameter方法，中文不会乱码

String name = request.getParameter("name");

String pwd = request.getParameter("password");

System.***out***.println(name);

System.***out***.println(pwd);

2)post提交方式下中文乱码处理

//post提交方式，解决中文乱码的方案

//tomcat内部使用了iso-8859-1的编码

request.setCharacterEncoding("utf-8");

## 2 HttpServletResponse

HttpServletResponse对象服务器的响应。该对象中封装了向客户端发送数据、发送响应头，发送响应状态码的方法

1)返回响应内容

//获取打印流的对象

PrintWriter writer = response.getWriter();

//writer.write("hello");

//writer.write("<html><head><title>hello</title></head><body><font color='red'>hello world</font></body></html>");

String html = "<html>";

html += "<head>";

html += "</head>";

html += "<body>";

html += " 你好";

html += "</body>";

html += "</html>";

writer.write(html);

writer.close();

2)中文乱码处理

response.setContentType("text/html;charset=utf-8");

3)重定向

response.setStatus(302);

response.setHeader("location", "/W06\_Servlet2/index.html");

4)刷新

//定时刷新，每隔一秒刷新一次

//response.setHeader("refresh", "1");

//指定时间（秒）后，跳转到另外的界面

response.setHeader("refresh", "3;url=/W06\_Servlet2/index.html");

## 3 Servlet注册

例如，访问：<http://localhost:8080/W09_Servlet3/hello>

Tomcat服务器截取到访问的资源名称/hello,然后从web.xml的<url-pattern>中找匹配的数据，找到后，再查找相关的servlet-name；然后从<servlet>节点中找匹配的servlet-name，找到后，就找到了对应的servlet-class，最后创建对象，执行其中的方法