回顾重点内容：

1）web服务软件：把本地资源文件共享给外部访问

2）tomcat基本使用

3）web应用（网站）目录结构

|-WebRoot 根目录

|- 普通静态网页文件（html、css。。。）

|-WEB-INF

|-classes 存放class文件。servlet类

|-lib 存放jar包

web.xml 文件。配置servlet

4）部署web应用三种方式

4.1）直接拷贝webapps

4.2）修改server.xml

在host标签中添加 <Context docBase="" path=""/>

4.3)直接在conf/Catalina/localhost目录添加xml文件

<Context docBase=""/>

5) 开发动态网页

servlet技术：使用java语言来开发动态网页的技术

开发servlet的步骤：

5.1）编写java类，继承HttpServlet，覆盖doGet方法

5.2）把servlet的class文件拷贝到项目的WEB-INF/classes目录下

5.3）在项目的WEB-INF/web.xml文件配置

<servlet>

<servlet-name></servlet-name>

<servlet-class></servlet-class>

</servlet>

<servlet-mapping>

<servlet-name></servlet-name>

<url-pattern>/hello</url-pattern>

</servlet-mapping>

5.4)访问servlet

http://localhost:8080/web/hello

# 1 http协议

## 1.1 概念

是对浏览器和服务器端数据传输格式的规范！

## 1.2 http协议内容

|  |
| --- |
| GET /day08/first HTTP/1.1  Host: localhost:8080  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-GB,en-US;q=0.8,zh-CN;q=0.6,zh;q=0.4,en;q=0.2  Accept-Encoding: gzip, deflate  Connection: keep-alive |

|  |
| --- |
| HTTP/1.1 200 OK  Server: Apache-Coyote/1.1  Content-Length: 24  Date: Mon, 08 Jun 2015 01:51:04 GMT |

# 2 Http请求

|  |
| --- |
| GET /day08/first HTTP/1.1 --请求行  Host: localhost:8080 --请求头（以键值对形式出现）  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-GB,en-US;q=0.8,zh-CN;q=0.6,zh;q=0.4,en;q=0.2  Accept-Encoding: gzip, deflate  Connection: keep-alive  --一个空行  name=eric&password=123456 --- 实体内容 |

## 2.1 请求行

GET /day08/first HTTP/1.1

http协议版本：

http1.0: 当浏览器和服务器连接之后，在一次连接中只能发出一个请求

http1.1:当浏览器和服务器连接之后，在一次连接中可以发出多次请求。（效率比1.0更高）

常见问题：

在一个html页面，在页面上有三张图片，访问这个html页面，问一共发出几次请求？

答案：4次

请求资源：请求服务器的哪个资源文件

/day08/first : URI： 统一资源描述符。可以描述任何文件，本地，互联网

http://localhsot:8080/day08/first：URL ： 统一资源定位符，互联网（基于http协议） ,是URI的子集。

请求方式（提交方式）：

http协议中的请求方式：GET、POST、HEAD、PUT、CONNECT。。。。

最常用的的请求方式： GET 和 POST

<form action="提交的地址" method="get/post">

</form>

GET：

1）提交的参数显示到地址栏

原理： get方式提交的参数会跟在请求行内容的URI后面。以？开始，每个参数使用=号分开,name=jacky,第二个参数之后使用&符合连接

|  |
| --- |
| GET /day08/testMethod.html?name=jacky&password=123456 HTTP/1.1  Host: localhost:8080  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-GB,en-US;q=0.8,zh-CN;q=0.6,zh;q=0.4,en;q=0.2  Accept-Encoding: gzip, deflate  Referer: http://localhost:8080/day08/testMethod.html  Connection: keep-alive |

2）不适合存放敏感数据

3）不超过1kb

POST：

1）提交的参数不会显示到地址栏

原理：post提交的参数跟在请求的实体内容中。第二个参数用&连接

|  |
| --- |
| POST /day08/testMethod.html HTTP/1.1  Host: localhost:8080  User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8  Accept-Language: en-GB,en-US;q=0.8,zh-CN;q=0.6,zh;q=0.4,en;q=0.2  Accept-Encoding: gzip, deflate  Referer: http://localhost:8080/day08/testMethod.html  Connection: keep-alive  name=jacky&password=123456 -实体内容 |

2）适合敏感数据

3）提交的大小没有限制

## 2.2 请求头

|  |
| --- |
| Accept: text/html,image/\* --浏览器接受的数据类型  Accept-Charset: ISO-8859-1 --浏览器接受数据编码格式  Accept-Encoding: gzip,compress --浏览器接受的数据压缩格式  Accept-Language: en-us,zh- ---浏览器接受的语言  Host: www.it315.org:80 --请求发出的主机和端口（必须）  If-Modified-Since: Tue, 11 Jul 2000 18:23:51 GMT --浏览器缓存的最后修改时间  Referer: http://www.it315.org/index.jsp -- 当前请求来自于哪里 （防止非法链接）  User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0) --浏览器的类型  Cookie:name=eric ---浏览器保存的cookie数据  Connection: close/Keep-Alive ---浏览器和服务器的连接状态。close：关闭。keep-alive:保持连接  Date: Tue, 11 Jul 2000 18:23:51 GMT --请求发出的时间 |

## 2.3 案例- user-agent头 -- 获取浏览器类型

## 2.4 案例- referer头 --- 防止非法链接

第一次：下载资源 - > 下载页面 -> 打开广告页面（下载链接） -> 开始下载

第二次：直接点击下载链接 -> 跳转广告页面(下载链接) -> 开始下载

非法请求：

1）直接访问下载资源

2）不是从广告页面过来的请求

referer请求头只有从超链接过来的请求才有这个头

## 2.5 获取参数数据

get: 放在请求行的URI后面

post： 放在请求的实体内容

reuqest.getParameter("name") 获取一个值的参数

request.getParameterValue("name") 获取多个值的参数

request.getParameterNames() 获取所有参数名称

request.getParameterMap() 获取所有参数对象

解决get乱码问题：

手动解码

|  |
| --- |
| if("GET".equals(request.getMethod())){  name = new String(name.getBytes("iso-8859-1"),"utf-8");  } |

解决post乱码问题：

|  |
| --- |
| request.setCharacterEncoding("utf-8"); |

# 3 Http响应

|  |
| --- |
| HTTP/1.1 200 OK --响应行  Server: Apache-Coyote/1.1 --响应头  Content-Length: 24  Date: Mon, 08 Jun 2015 01:51:04 GMT  --一个空行  -实体内容 |

## 3.1 响应行

### http协议版本

### 状态码：当前服务器处理请求的状态（结果）

常见的状态码：

200： 请求处理完成，成功返回！

302： 需要浏览器进一步请求，才能完成。

404： 浏览器端的错误。没有找到资源。

500： 服务器端的错误。

### 状态描述：对状态码的文字补充

## 3.2 响应头

|  |
| --- |
| Location: http://www.it315.org/index.jsp --重定向的地址。结合302状态使用完成重定向的效果  Server:apache tomcat --服务器的类型  Content-Encoding: gzip --服务器发送给浏览器的数据压缩格式  Content-Length: 80 --服务器发送给浏览器的数据长度  Content-Language: zh-cn --服务器支持语言  Content-Type: text/html; charset=GB2312 --服务器发送给浏览器的数据类型和数据编码格式  Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT --服务器资源的最后修改时间  Refresh: 1;url=http://www.it315.org ---定时刷新或每隔n秒跳转资源  Content-Disposition: attachment; filename=aaa.zip -- 以下载方式打开资源  Transfer-Encoding: chunked  Set-Cookie:SS=Q0=5Lb\_nQ; path=/search -- 服务器发送给浏览器的cookie数据  Expires: -1 --通知浏览器不使用缓存  Cache-Control: no-cache  Pragma: no-cache  Connection: close/Keep-Alive 连接状态  Date: Tue, 11 Jul 2000 18:23:51 GMT 响应发出的时间 |

总结：

1）http协议： 对浏览器和服务器之间数据传输格式规定。

2）http请求：（浏览器->服务器）

2.1 请求格式

请求行（请求方式+请求资源+http协议）

请求头（user-agent/referer）

空行

实体内容

2.2 对开发获取请求数据：

HttpServletRequest对象：

获取的方法

3）http响应（服务器-》浏览器）

3.1 响应格式

响应行（http协议+状态码+描述）

响应头（location+refresh+content-type+content-diposition）

空行

实体内容

3.2 对开发修改响应数据

修改方法