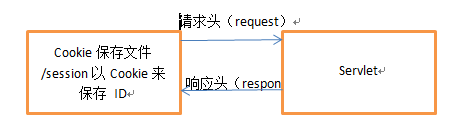
## 创建servlet小型程序服务器：

每个Cookie或者Session保存文件中都有一连串的数字，表示ID，在发送给Session在请求头中发送给服务器中，而Servlet有用同样响应头的发送给客户端找到相应的Cookie或Session



前提是用创建文件，由于不需要工具是使用这个来创建servlet

1. 继承与HttpServlet方法
2. 重写doGet或doPost方法（HttpServlet有七种方法可以重写）
3. 配置servlet，就是web.xml文件，告示服务器什么样的url地址访问当前的servlet
4. 写入你需要的代码
5. 编译好的class文件放到项目的/WEB-INF/classs文件夹下

不过要注意是：servle不可以是裸体类

## 真正的项目开发，直接创建servlet

Servlet的生命周期：

生命周期：当第一次访问servlet时，调用他的构造方法先创建对象，后才调用inin方法，初始化功能以读取文件的配置，知道URL地址才能读取ServletConf，然后根据请求方式调用相应的server方法，此后servlet不再调用构造方法，因为servlet是单例的，也不再调用init方法，服务器只读取一次url地址，只有重新加载时，才会调用detroy方法。

什么是单例：只需要创建一次对象。

什么地方用到单例：当多个线共用一个实例，那么就使用单例，且单例会减少内存消耗

单例例子：class P{

private static P p = new P();

private P(){}

public static P getStem(){

return p;

}

}

private static P p;

static {//使用static语句块，当一个类被加载jvm时执行，说明在servlet不能用这样语句块，应为servlet自动加载，会得到多个实例

P p = new P();

}

生命周期例子：

public dateServlet() {

System.out.println("-----------------cons-----------------");

}

@Override

public void destroy() {

System.out.println("-----------------destroy-----------------");

}

@Override

public void init() throws ServletException {

System.out.println("-----------------init-----------------");

}

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

System.out.println("-----------------get-----------------");

}

-----------------cons-----------------

-----------------init-----------------

-----------------get-----------------

七月 08, 2012 9:15:14 上午 org.apache.catalina.core.StandardContext reload

信息: Reloading Context with name [/servlet] has started

-----------------destroy-----------------

Init()

有初始化功能，初始化配置文件，也就是读取web.xml中ServletConfig，从而知道url地址

注意：有两个init方法，一个有参数，一个没有。

public void init(ServletConfig config) throws ServletException {

各种配置来读取URL地址

this.init();//用这个重写init（无参）不用super.init(config);

}

public void init(ServletConfig config) throws ServletException {

随意编写

}

当不写init()时，系统默认有参数的init()

Destroy()

加载是调用

Web.xml

配置：<servlet>

<servlet-name>Shows</servlet-name>

<servlet-class>com.wu.Shows</servlet-class>

<load-on-startup>n</load-on-startup>//当数字为负数是，在服务器开始期间运行，n>0时在开启服务器后运行，n的数字表示不同的顺序，愈大愈后

</servlet>

<servlet-mapping>

<servlet-name>c</servlet-name>

<url-pattern>/welcome</url-pattern>

</servlet-mapping>

## Cookie Session application

http：本身是无状态的协议

就是说服务器根本不可能知道你在之前的页面做过什么，其实在真正意思上来讲，在一个登入界面，登入后，只有改页面才知道你已经登入了，在访问其他的页面别的页面根本不知道登入情况

所以就需要Cookie，session，application记录状态，以不断的刷新来使用session

### 1.Cookie

1. 保存在客服端操作系统中的文件
2. 保存在浏览器中，没写cookie.setMaxAge(时间)当关闭浏览器中时没有了缓存了，也就没有保存
3. 保存在操作系统中，以文件形式保存，设置时间cooki.setMaxAge(3000)）
4. 只能保存字符串，以文件形式保存
5. Cookie是名—值对的形式

### 2.Session

1. 保存在服务器内存中
2. 可以保存java任何类型数据

### 3.创建Cookie实例：

Cookie cookie = new Cookie(“name”,”value”);//创建对象，自己来创建对象的（名——值）

其中name中只能保存一个值，会覆盖值

cookie.setMaxAge(3000);//设置时间，不会保存在浏览器中

response.addCookie(cookie);//保存cookie存入的值

读取cookie数据

Cookie[] cookie = request.getCookie();//接收cookie中保存的数据，就算不创建对象，浏览器中也会自动创建对象，所以需要保存什么，用指令也可以保存

PrintWrite out = request.getWriter();//打印到浏览器

Response.sendRedirect(“./show”);//跳转地址，如果是.java 文件则不需要后缀名了，html就需要 /项目的根路径，但不是实际的路径

Cookie c1 = new Cookie("name","value");

c1.setMaxAge(3000);

response.addCookie(c1);

Cookie c2 = new Cookie("age","19");

c2.setMaxAge(3000);

response.addCookie(c2);

Cookie c3 = new Cookie("whiter","bai");

c3.setMaxAge(3000);

response.addCookie(c3);

response.sendRedirect("./showTest");

得到Cookie保存的值：

PrintWriter out = response.getWriter();

Cookie[] cookie = request.getCookies();

if(cookie!=null){

for(Cookie c : cookie){

out.println("name="+c.getName()+"---->value---"+c.getValue());

}

}

else{

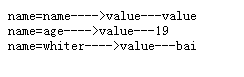
out.print("没有传入值");

}

out.flush();

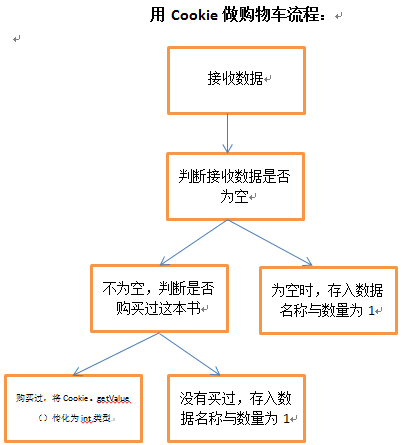
out.close();

答案：



## 4.购物车

做项目的时候需要将想法转化流程



1. 接受数据，用 String[] str = request.ParameterValues(“books”);接收数据name
2. 拿到客户端的所有的cookie，Cookie[] cookie = request.getCookies();接收数据name value
3. 判断是否购买过这个商品

if(str[i].equals(cookies[j].getName()))

1. 购买后将保存在cookie中的值传化为int类型
2. 没有购买时，返回value为“1”

接收参数，以及叠加数目

String[] str = request.getParameterValues("books");

Cookie[] cookies = request.getCookies();

if(str!=null || cookies!=null){

for(int i=0; i<str.length;i++){

for(int j=0 ; j<cookies.length;j++){

if(str[i].equals(cookies[j].getName())){

int num = Integer.parseInt(cookies[j].getValue())+1;

Cookie cookie = new Cookie(str[i],num+"");

response.addCookie(cookie);

// response.sendRedirect("./index.html");

// return;

}else{

Cookie cookie = new Cookie(cookies[j].getValue(),"1");

response.addCookie(cookie);

}

}

}

}else{

for(int i=0 ;i<cookies.length;i++){

Cookie cookie1 = new Cookie(cookies[i].getValue(),"1");

response.addCookie(cookie1);

}

}

response.sendRedirect("./index.html");

付账：

Cookie[] cookie = request.getCookies();

response.setCharacterEncoding("gb2312");

PrintWriter out = response.getWriter();

if(cookie == null){

out.print("您没有购买任何商品，谢谢！！！");

}

else{

out.println("购买的商品有");

for(int i=0 ; i<cookie.length; i++){

out.println(cookie[i].getName()+":"+cookie[i].getValue()+"件");

}

}

out.flush();

out.close();

Cookie 是在第一次访问后被创建，就是说当第一次进入的时候没有创建cookie中的对象，保存cookie对象了，所以必须判断cookie 是否为空，也必须把Cookie的数量保持为1；

购物车的写的包装类：

在写购物车出现问题解决情况：

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String[] computers = request.getParameterValues("computer");

for(int i=0;i<computers.length;i++){

System.out.println(computers[i]);

}//可得到对应name 如 acer

Cookie[] cookie = request.getCookies();

if(computers == null){

}

else if (cookie == null) {

for (int i = 0; i < computers.length; i++) {

Cookie cookies = new Cookie(computers[i], "1");

response.addCookie(cookies);

}

}else{

for (int i = 0; i < computers.length; i++) {

for (int j = 0; j < cookie.length; j++) {

if(computers[i].equals(cookie[j].getName())){

int num = Integer.parseInt(cookie[j].getValue())+1;

Cookie cookies = new Cookie(computers[i], num+"");

response.addCookie(cookies);

response.sendRedirect("./index.html");

return;

}

}

Cookie cookies = new Cookie(computers[i], "1");

response.addCookie(cookies);

}

}

response.sendRedirect("./index.html");

}

String[] computers = request.getParameterValues("computer");

for(int i=0;i<computers.length;i++){

System.out.println(computers[i]);

}//可得到对应name 如 acer

Cookie[] cookie = request.getCookies();

**for**(Cookie c : cookie)

System.*out*.println("name"+c.getName()+"values"+c.getValue());

得到的是：name Iphone values 1

Name hp values 6

name acer values 3

public void doGet(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

String[] books = request.getParameterValues("book");

Cookie[] cookies = request.getCookies();

if (books != null) {

if (cookies != null) {

for (int i = 0; i < books.length; i++) {

for (int j = 0; j < cookies.length; j++) {

if (books[i].equals(cookies[j].getName())) {

int num = Integer.parseInt(cookies[j].getValue()) + 1;

Cookie cookie = new Cookie(books[i], num + "");

response.addCookie(cookie);

response.sendRedirect("./index.html");

return;

}

}

Cookie cookie = new Cookie(books[i], "1");

response.addCookie(cookie);//

作用：第一次点击的时，不知道由于里面的循环，

}

}

// } else {

// for (int i = 0; i < books.length; i++) {

// Cookie cookie = new Cookie(books[i], "1");

// response.addCookie(cookie);

// }

// }

} else {

}

response.sendRedirect("./index.html");

}

### 5.Session

1.在服务器中创建，保存在服务器中

关闭浏览时，所保存的东西被清空，而一定不代表内存中的Session没有了，只有在规定的30分钟后才被清除

2.定义和创建session，用HttpSession得到getSession(true)

3.Session在30分钟后过期

4.登入界面是有Session写入状态，其中上网的时候，一般规定是1,2,3,5分钟过期时间，服务器不断的刷新才能使session不过期

5.Session 是由Cookie来实现的

如果禁用了Cookie情况下，依然要使用Session 那么必须使用URL参数中写的方法

<a href=”./show;jsession=<%=response.encodeURL(./show) %>>

1. Sesssion是被创建时，自动生成sessionId，则需要用Cookie把session存入浏览器中，在通过SessionId得到Session
2. 用setAttribute();存储，

void **setAttribute**([String](http://java.sun.com/j2se/1.5/docs/api/java/lang/String.html) name,[Object](http://java.sun.com/j2se/1.5/docs/api/java/lang/Object.html) value)

与Cookie不同之处最大时存储的位置不同，还以一个就是传入的值是Object的，就是因为Cookie只可以存入字符串得到字符串，而Session是java的任何java类型，所以……

1. getAttribute()取得，由于

得到：[Object](http://java.sun.com/j2se/1.5/docs/api/java/lang/Object.html) **getAttribute**([String](http://java.sun.com/j2se/1.5/docs/api/java/lang/String.html) name) Object类，所以要强制转换成相应的类型

HttpSession sessions = request.getSession(true);//用HttpSession获取定义session

sessions.setAttribute("UID", "123456");//存储其中信息

String uid = (String) sessions.getAttribute("UID");//得到Session存入的值

PrintWriter out = response.getWriter();

out.print(uid);//打印出来

### application（所有用户共享）

Context（运行的环境）

Context（容器）：

什么是容器：如果它可以管理某种对象的生命周期，她就称为容器

Tomcat：tomcat是servlet的容器运行的环境

ServletContext（Servlet的运行环境）可以找到所有的运行环境

1. ServletContext application = this.getServletContext(request.getSession().getServletConetext())
2. 作用是，访问量是需要