

会话过程要解决的问题：保持整个各个客户端自己的数据  
保存会话技术：

## cookie对象

- 1, cookie是客户端技术，程序把每个用户的数据以cookie的形式写给用户各自的浏览器，当用户使用浏览器去访问服务器中的web资源时，就会带着自己各自的数据去，这样，web资源处理的就是用户各自的数据
- 2, httpsession是服务器端技术，采用这个技术，服务器在运行时可以为每一个用户的浏览器创建一个其独享的httpsession对象，用户session为用户浏览器独享，所以用户在访问服务器的web资源时，可以把各自的数据放在各自的session中，当用户去访问服务器中的其他web时，其他的web资源再从用户各自的session中取出数据为用户服务

cookie在javax中的servlet中的http

javax.servlet.http.Cookie类用于创建一个cookie，response接口中定义了一个addCookie方法，它用于在其响应头中添加一个相应的set-cookie头字段，

cookie不能频繁的使用

cookie只能存储字符串

## cookie细节

- ①一个cookie只能标识一种信息，它至少含有一个标识该信息的名称（name）和设置值（value）
- ②一个web站点可以给一个web浏览器发送多个cookie，一个web浏览器也可以存储多个web站点的cookie
- ③浏览器一般只允许存放300个cookie，每个站点最多存放20个cookie，每个cookie的大小限制在4kb
- ④删除cookie时，path必须一致，否则不会删除
- ⑤如果创建一个cookie。并将它发送到浏览器，默认情况下它是一个会话级别的cookie，即存储在浏览器的内存中，用户退出浏览器之后即被删除，若希望浏览器将该cookie存储在磁盘上，则需要使用maxAge，并给出一个以秒为单位的时间，将最大时效设置为0则是命令浏览器删除该cookie

## 属性:

name: 名称不能唯一确定一个cookie, 路径不同则不同

value: 不能写中文

path: 默认值是写cookie的那个程序的访问路径

比如: `http://localhost:8080/newS/servlet/test6`中写ccookie

path就是newS/servlet/看当前创建cookie的资源servlet文件路径

客户端在访问服务器另外资源时, 根据访问的路径来决定是否带cookie到服务器

当前访问的路径如果是以cookie的path开头的路径, 浏览器就带, 否则就不带

maxAge: cookie的缓存时间, 默认值是-1 (默认存储在浏览器的内存中) 单位是秒

负数: cookie的数据存储在浏览器的缓存中

0: 删除, 路径要保持一致, 否则可能被删错

正数: 缓存 (持久化到磁盘上) 的时间

## //获取客户端保存的最后访问时间

```
Cookie cookie[] = request.getCookies();//获得客户端所有的cookie
对象

for(int i = 0; cookie != null && i < cookie.length; i++)
{
    //判断当前cookie中的name是否是想要的cookie对象
    if (("lastAccessTime").equals(cookie[i].getName())) {
        //如果是想要的cookie对象, 则把cookie对象中的值取出
        long l = Long.parseLong(cookie[i].getValue());
        out.write("你的最后访问时间为: " + new
Date().toLocaleString());
    }
}

out.print("<a href =
'" + request.getContextPath() + "/servlet/clear' >clear</a>");
//创建cookie, 并把信息写回客户端
Cookie ck = new
Cookie("lastAccessTime", System.currentTimeMillis() + "");
//设置cookie的有效时间, 单位是秒
ck.setMaxAge(60*5); //设置保存时间为五分钟
```

```
//设置cookie的path, 表示在当前项目下都可以进行访问cookie中的值
ck.setPath("/");
//把cookie信息写在客户端
response.addCookie(ck);
```

### 删除cookie对象

```
//创建cookie对象
Cookie ck = new Cookie("lastAccessTime","");
ck.setPath("/");//设置被删除cookie的path, 否则可能会删错对象
ck.setMaxAge(0);//相当于删除
response.addCookie(ck);//将ck写回客户端
```

cookie对象属性补充:

servlet需要做的事情:

- ①获取表单数据
- ②处理业务逻辑
- ③分发转向

### 实现登录的时候保存用户名:

#### login页面:

```
response.setContentType("text/html;charset=utf-8");
request.setCharacterEncoding("utf-8");
PrintWriter out = response.getWriter();

//得到客户端保存的cookie数据
String nameString = "";
String checkString = "";
Cookie [] ck = request.getCookies();
for(int i = 0;ck!=null&&i<ck.length;i++)
{
    if ("username".equals(ck[i].getName())) {
        nameString = ck[i].getValue();
        checkString = "checked='checked' ";
    }
}

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

```

        out.print("<form action =
'"+request.getContextPath()+"/servlet/dologin' method = 'post'>");
        out.print("用户名<input type = 'text' name = 'username'
value=' "+nameString+" ' /></br>");
        out.print("密码<input type = 'password' name = 'pass' /></br>");
        out.print("记住用户名<input type = 'checkbox' name =
'remember' "+checkString+" /></br>");
        out.print("<input type = 'submit' name = 'submit' value = '提交' />
</br>");

        out.print("</form>");

```

### dologin页面:

```

request.setCharacterEncoding("utf-8");
response.setContentType("text/html;charset=utf-8");
PrintWriter out = response.getWriter();
response.setCharacterEncoding("utf-8");
//获取表单数据
String nameString = request.getParameter("username");
String pasString = request.getParameter("pass");
String rember = request.getParameter("remember");
Cookie ck = new Cookie("username",nameString);
//处理业务逻辑
//分发转向
if ("tom".equals(nameString) && "123".equals(pasString)) {
    if (rember!=null) {
        ck.setPath("/");
        ck.setMaxAge(Integer.MAX_VALUE); //设置cookie的有效保存时间为最大值
    }
    else {
        ck.setMaxAge(0); //删除cookie
    }
    response.addCookie(ck); //写回客户端
    out.print("登录成功! ");
}

```

```
else {  
    out.print("登录失败！");  
    //设置俩秒钟重新登录  
    response.setHeader("refresh",  
"2;url="+request.getContextPath()+"/servlet/login");  
}
```