### 课程介绍：

目前大家对于浏览器和服务器的交互模式，以及请求的处理都有了理解，并且也能够进行请求的处理。本节课围绕整个流程再次进行技术优化，重点学习 Cookie 技术，此技术的应用面是非常广的。Session

Cookie 学习：

问题：

HTTP 协议是没有记忆功能的，一次请求结束后，相关数据会被销毁。如果第二次的请求需要使用相同的请求数据怎么办呢？

难道是让用户再次请求书写吗?

解决：

使用 Cookie 技术

|  |
| --- |
|  |

解释：

Cookie 技术其实是浏览器端的数据存储技术，解决了同一个工程下不同请求需要使用相同的请求数据的问题。我们把请求需要共享的请求数据，存储在浏览器端，避免用户进行重复的书写请求数据。但是哪些数据需要使用 Cookie 技术存储起来是一个主观问题，需要在后台进行响应的时候来告诉浏览器，有些数据其他请求还会使用，需要存储起来。

特点：

浏览器端的数据存储技术

适合少量数据

键值对

不安全

注意：一个Cookie对象存储一条数据，多条数据，可以创建多个Cookie对象进行存储。

作用：Cookie 技术解决不同请求发送之间的数据共享问题。

|  |
| --- |
| **package** com.serlvet;  **import** com.pojo.User; **import** com.service.UserServieImpl;  **import** javax.servlet.ServletException; **import** javax.servlet.http.Cookie; **import** javax.servlet.http.HttpServlet; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException; **import** java.net.URLEncoder;  **public class** LoginServlet **extends** HttpServlet {  UserServieImpl **usi**;  **public** LoginServlet(){  **usi** = **new** UserServieImpl();  }  @Override  **protected void** service(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  resp.setContentType(**"text/html;charset=utf-8"**);  String uname =req.getParameter(**"uname"**);  String pwd =req.getParameter(**"pwd"**);  *//调用service* User user = **usi**.login(uname,pwd);  **if**(user != **null**){  System.***out***.println(**"登录了"**);  *//创建Cookie对象 // Cookie c = new Cookie("user", user.getUname());  //旧版本web中cookie是中文的时候要进行url传输编码处理URLEncoder.encode(user.getUname(),"utf-8")* Cookie c = **new** Cookie(**"user"**, URLEncoder.*encode*(user.getUname(),**"utf-8"**));  Cookie c2 = **new** Cookie(**"pwd"**,user.getPwd());  *//重定向,跳转页面* resp.addCookie(c);*//把Cookie对象放入resp对象,响应到浏览器* resp.addCookie(c2);*//把Cookie对象放入resp对象,响应到浏览器* resp.sendRedirect(**"Success"**);  }**else**{  resp.getWriter().write(**"<font color='red'>用户或密码有误!</font>"**);  }  } } |
| **package** com.serlvet;  **import** com.pojo.User;  **import** javax.servlet.ServletException; **import** javax.servlet.http.Cookie; **import** javax.servlet.http.HttpServlet; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException; **import** java.net.URLDecoder;  **public class** Success **extends** HttpServlet {  @Override  **protected void** service(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {   resp.setContentType(**"text/html;charset=utf-8"**);  Cookie[] cookies = req.getCookies();  String value =**null**;  String pwd =**null**;  **for**(Cookie c:cookies){  **if**(c.getName().equals(**"user"**)){ *// value = c.getValue();  //解码url传输的数据* value = URLDecoder.*decode*(c.getValue(),**"utf-8"**);  }  **if**(c.getName().equals(**"pwd"**)){  pwd = c.getValue();  }  }  resp.getWriter().write(**"欢迎"**+value+**"登录,XX管理系统"**);  resp.getWriter().write(pwd);  } } |

Cookie 数据存储：

默认情况下临时存储：

不设置 cookie 信息的存储时间，周期为一次会话，存储在浏览器的运行内存中，关闭浏览器即失效了

默认是临时存储：默认cookie存储之后，每次请求都会附带，除非设置有效路径

|  |
| --- |
| **package** com.serlvet;  **import** com.pojo.User; **import** com.service.UserServieImpl;  **import** javax.servlet.ServletException; **import** javax.servlet.http.Cookie; **import** javax.servlet.http.HttpServlet; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException; **import** java.net.URLEncoder;  **public class** LoginServlet **extends** HttpServlet {  UserServieImpl **usi**;  **public** LoginServlet(){  **usi** = **new** UserServieImpl();  }  @Override  **protected void** service(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  resp.setContentType(**"text/html;charset=utf-8"**);  String uname =req.getParameter(**"uname"**);  String pwd =req.getParameter(**"pwd"**);  *//调用service* User user = **usi**.login(uname,pwd);  **if**(user != **null**){  System.***out***.println(**"登录了"**);  *//创建Cookie对象 // Cookie c = new Cookie("user", user.getUname());  //旧版本web中cookie是中文的时候要进行url传输编码处理URLEncoder.encode(user.getUname(),"utf-8")* Cookie c = **new** Cookie(**"user"**, URLEncoder.*encode*(user.getUname(),**"utf-8"**));  Cookie c2 = **new** Cookie(**"pwd"**,user.getPwd());  c2.setPath(**"/SecondWeb/Success"**);*//设置携带cookie请求的路径,就会仅仅在访问设置的路径才携带此cookie  //重定向,跳转页面* resp.addCookie(c);*//把Cookie对象放入resp对象,响应到浏览器* resp.addCookie(c2);*//把Cookie对象放入resp对象,响应到浏览器* resp.sendRedirect(**"Success"**);  }**else**{  resp.getWriter().write(**"<font color='red'>用户或密码有误!</font>"**);  }  } } |

定时存储：

设置存储时间，周期为时间设置，存储在用户电脑硬盘中。在有效期内符合路径要求的请求都会附带该信息。

|  |
| --- |
| **package** com.serlvet;  **import** com.pojo.User; **import** com.service.UserServieImpl;  **import** javax.servlet.ServletException; **import** javax.servlet.http.Cookie; **import** javax.servlet.http.HttpServlet; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException; **import** java.net.URLEncoder;  **public class** LoginServlet **extends** HttpServlet {  UserServieImpl **usi**;  **public** LoginServlet(){  **usi** = **new** UserServieImpl();  }  @Override  **protected void** service(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  resp.setContentType(**"text/html;charset=utf-8"**);  String uname =req.getParameter(**"uname"**);  String pwd =req.getParameter(**"pwd"**);  *//调用service* User user = **usi**.login(uname,pwd);  **if**(user != **null**){  System.***out***.println(**"登录了"**);  *//创建Cookie对象 // Cookie c = new Cookie("user", user.getUname());  //旧版本web中cookie是中文的时候要进行url传输编码处理URLEncoder.encode(user.getUname(),"utf-8")* Cookie c = **new** Cookie(**"user"**, URLEncoder.*encode*(user.getUname(),**"utf-8"**));  Cookie c2 = **new** Cookie(**"pwd"**,user.getPwd());  c2.setMaxAge(10);*//给Cookie c2设置有效时间,单位是秒  //重定向,跳转页面* resp.addCookie(c);*//把Cookie对象放入resp对象,响应到浏览器* resp.addCookie(c2);*//把Cookie对象放入resp对象,响应到浏览器* resp.sendRedirect(**"Success"**);  }**else**{  resp.getWriter().write(**"<font color='red'>用户或密码有误!</font>"**);  }  } } |

Cookie 学习案例：三天免密码登录

|  |
| --- |
| **package** com.serlvet;  **import** com.pojo.User; **import** com.service.UserServieImpl;  **import** javax.servlet.ServletException; **import** javax.servlet.http.Cookie; **import** javax.servlet.http.HttpServlet; **import** javax.servlet.http.HttpServletRequest; **import** javax.servlet.http.HttpServletResponse; **import** java.io.IOException; **import** java.net.URLEncoder;  **public class** LoginServlet **extends** HttpServlet {  UserServieImpl **usi**;  **public** LoginServlet(){  **usi** = **new** UserServieImpl();  }  @Override  **protected void** service(HttpServletRequest req, HttpServletResponse resp) **throws** ServletException, IOException {  *//请求编码* req.setCharacterEncoding(**"utf-8"**);  *//响应编码* resp.setContentType(**"text/html;charset=utf-8"**);  String uname = req.getParameter(**"uname"**);  String pwd = req.getParameter(**"pwd"**);   *//免密登录* **if**(uname.equals(**""**) || pwd.equals(**""**)){ *//当用户名和密码输入框有一个是空的时候进免账户和密码登录逻辑  //从cookie中取账户和密码* Cookie[] cookies = req.getCookies();  **for**(Cookie c:cookies){  **if**(**"uname"**.equals(c.getName())){  uname = c.getValue();  }  **if**(**"pwd"**.equals(c.getName())){  pwd = c.getValue();  }  }  }  User user = **usi**.login(uname, pwd);  *//把用户名和密码放入cookie* Cookie cuname = **new** Cookie(**"uname"**,user.getUname());  Cookie cpwd = **new** Cookie(**"pwd"**,user.getPwd());  *//设置三天免登录* cuname.setMaxAge(3600\*24\*3);  cpwd.setMaxAge(3600\*24\*3);  *//把cookie放入响应对象中* resp.addCookie(cuname);  resp.addCookie(cpwd);  **if**(user != **null**){  *//跳转页面,登录转账使用重定向* resp.sendRedirect(**"Success"**);  }   } } |