Servlet Cookie基本概念和使用方法

Cookie 介绍

Cookie 是一种在网站和应用程序中用于存储用户信息的小型文本文件。当用户访问一个网站或应用程序时，该网站或应用程序会将一个包含用户信息的 Cookie 发送到用户的浏览器。浏览器会将该 Cookie 存储在用户的计算机上，并在以后的访问中将该 Cookie 发送回网站或应用程序。

虽然 Cookie 对于提供个性化体验和方便用户来说非常有用，但它们也引发了一些隐私和安全问题。例如，第三方 Cookie 可以用于跟踪用户在多个网站上的活动，可能会侵犯用户的隐私。出于隐私和安全的考虑，现代浏览器通常允许用户控制哪些 Cookie 被接受和存储，并提供了清除 Cookie 的选项。

Cookie 主要有两种类型：会话 Cookie 和持久 Cookie。

- 会话 Cookie：这些 Cookie 在用户关闭浏览器时会被删除。它们主要用于跟踪用户在当前会话中的活动，如用户在网站上浏览的页面、添加到购物车的商品等。
- 持久 Cookie：这些 Cookie 在用户关闭浏览器后仍然存在，可以在指定的时间段内保留。它们用于存储长期的用户信息，如用户的登录凭据、偏好设置等。

Cookie使用步骤

1. 创建一个Cookie对象：

```java
Cookie cookie = new Cookie("cookieName", "cookieValue");
```

2. （可选）设置Cookie的属性：

```java
cookie.setMaxAge(3600); // 设置Cookie的生存期（以秒为单位），在这个示例中为1小时
cookie.setPath("/"); // 设置Cookie适用的路径，这里是根路径，表示对整个应用程序可见
cookie.setDomain("example.com"); // 设置Cookie适用的域名，例如example.com
cookie.setSecure(true); // 设置Cookie只在通过HTTPS连接时传输
```

3. 将Cookie添加到HTTP响应中：

这将在HTTP响应的头部添加一个Set-Cookie标头，告诉客户端保存该Cookie。

```java
response.addCookie(cookie);
```

4. 在后续的请求中访问Cookie：

你可以通过**request.getCookies()**方法获取请求中的所有Cookie对象，并遍历它们以访问每个Cookie的名称和值。

```java
Cookie[] cookies = request.getCookies(); // 获取所有的Cookie对象

if (cookies != null) {
    for (Cookie cookie : cookies) {
        String name = cookie.getName();
        String value = cookie.getValue();
        // 处理Cookie数据
    }
}
```

使用Servlet和Cookie实现客户端存储的登录功能示例：

LoginServlet类

Java | 复制代码

```java
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.Cookie;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

import java.io.IOException;
import java.io.PrintWriter;

@WebServlet("/login")

public class LoginServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;

    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        // 检查用户名和密码是否有效（在此处添加验证逻辑）

        if (isValidUser(username, password)) {
            // 创建Cookie对象
            Cookie userCookie = new Cookie("username", username);
            // 设置Cookie的生命周期（这里设置为1小时）
            userCookie.setMaxAge(60 * 60);
            // 将Cookie添加到响应中
            response.addCookie(userCookie);

            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.println("<html>");
            out.println("<body>");
            out.println("<h3>欢迎, " + username + "!</h3>");
            out.println("</body>");
            out.println("</html>");
        } else {
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.println("<html>");
            out.println("<body>");
            out.println("<h3>登录失败，请检查用户名和密码。</h3>");
            out.println("</body>");
```

```
            out.println("</html>");
        }
    }

    private boolean isValidUser(String username, String password) {
        // 在此处进行用户名和密码的验证，可以连接数据库或使用硬编码的方式进行验证
        // 返回true表示验证通过，返回false表示验证失败
        // 这里只是一个示例，实际应用中应该使用更安全的验证方式
        return "A".equals(username) && "123".equals(password);
    }
}
```

index.jsp

HTML | 复制代码

```html
<html>
  <head>
    <title>登录界面</title>
    <style>
      body {
        background-color: #f1f1f1;
        font-family: Arial, sans-serif;
      }

      .container {
        width: 300px;
        margin: 0 auto;
        margin-top: 100px;
        background-color: #ffffff;
        padding: 20px;
        border-radius: 5px;
        box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
      }

      .container h2 {
        text-align: center;
      }

      .container input[type="text"],
      .container input[type="password"] {
        width: 100%;
        padding: 10px;
        margin-bottom: 20px;
        border: 1px solid #ccc;
        border-radius: 4px;
      }

      .container input[type="submit"] {
        width: 100%;
        padding: 10px;
        background-color: #4CAF50;
        border: none;
        color: #fff;
        border-radius: 4px;
        cursor: pointer;
      }

      .container input[type="submit"]:hover {
        background-color: #45a049;
      }
```

```
    </style>
  </head>
  <body>
    <div class="container">
      <h2>登录</h2>

      <form action="/login" method="post">
        <label for="username">用户名:</label>
        <input type="text" id="username" name="username" required><br><br>

        <label for="password">密码:</label>
        <input type="password" id="password" name="password" required><br>
<br>

        <input type="submit" value="Login">
      </form>
    </div>
  </body>
</html>
```

删除Cookie

不设置有效期，关闭浏览器，自动失效；

设置有效期时间为 0 ；

浏览器中查看Cookie的方法

1. 谷歌浏览器：
    - 打开Chrome浏览器，并导航到您感兴趣的网站。
    - 点击右上角的菜单图标（三个垂直线点），选择“更多工具”。
    - 在下拉菜单中选择“开发者工具”。
    - 在开发者工具窗口中，选择“应用”选项卡。
    - 在左侧导航栏中，展开“存储”，然后点击“Cookies”。
    - 在右边的面板中，您将看到该网站设置的 Cookie 列表。
2. 微软浏览器：
    - 打开Edge浏览器，并导航到您感兴趣的网站。
    - 点击右上角的菜单图标（三个水平点）。
    - 在下拉菜单中选择“更多工具”。

- 在弹出的菜单中选择“开发人员工具”。
- 在开发者工具窗口中，选择“应用”选项卡。
- 在左侧导航栏中，展开“存储”，然后点击“Cookies”。
- 在右边的面板中，您将看到该网站设置的 Cookie 列表。