

Cookie

1、--Cookie是什么？作用？Cookie保存位置？

-字面意思：

翻译就是曲奇饼干，也是Cookie文件的意思

-作用：

Cookie可以保存的一个会话状态，使会话状态保存在客户端上

Cookie被清除或者失效后，这个会话状态就没有了

-保存位置：

Cookie是保存在客户端浏览器上的，有两个位置：

保存在浏览器的缓存中，是即时的，浏览器关闭后这种Cookie就会被清除

保存在客户端的硬盘文件中，有保存期限，在该期限内关闭浏览器Cookie不会被清除，除非手动删除或过期失效

2、--哪些开发中有Cookie？

只要是web开发，是B/S架构系统，基于HTTP协议的开发都有Cookie的存在

所以说Cookie不只是存在于javaweb中

Cookie这种机制是HTTP协议规定的。也就是W3C制定的

3、--我们所熟悉Cookie实现的功能？

-在京东购物网上未登录添加商品进购物车的状态保留在客户端上

-实现十天免登录

...

4、--在Java中，Cookie是一种类class，存在于Servlet规范中，所以new出Cookie对象

需要注意的是javax.servlet.http.Cookie这个类中，是没有无参的构造方法的。

其构造方法需要传两个String类型的参数，两参数含义是name和value。所以说new出的Cookie对象由name和value两部

分组成

5、--怎么创建Cookie？

```
Cookie cookie = new Cookie(String name,String value);
```

6、--服务器可以一次性向客户端发送多个Cookie

7、--默认情况下，服务器发送到浏览器的Cookie被保存在浏览器的缓存中，浏览器不关闭，Cookie一直存在且有效，浏览器 被关闭后Cookie立马被清除

8、--那么存在于浏览器缓存或者是其硬盘文件中的Cookie，什么时候会被浏览器发送给服务器？

-浏览器要提交这些保存的Cookie，跟请求路径有关系

-请求路径与Cookie紧密关联

-不同的请求路径可以发送不同的Cookie

9、--默认情况下，被保存的Cookie和哪些路径绑定？

我们用这样的请求路径：

--

/idea_servlet_http_cookie_11_war_exploded/test/createAndSendCookieToBrowser

将Cookie从服务器发送给客户端

这个被浏览器保存的Cookie默认和test/路径绑定在一起：

就是说这个Cookie存在还在有效时间，发送test/路径的请求：

--http://IP:port/idea_servlet_http_cookie_11_war_exploded/test/

或者这样

--http://IP:port/idea_servlet_http_cookie_11_war_exploded/test/a

Cookie一定会被提交给服务器

下面没有test/路径的就不会提交Cookie:

--http://IP:port/idea_servlet_http_cookie_11_war_exploded/test1/a

和这样的

--http://IP:port/idea_servlet_http_cookie_11_war_exploded/test

经过测试:

--/idea_servlet_http_cookie_11_war_exploded/a

将Cookie从服务器发送给客户端

这个被浏览器保存的Cookie默认和idea_servlet_http_cookie_11_war_exploded/路径绑定在一起

10、--发送的Cookie是可以在创建该Cookie后进行请求路径绑定的

假设在java程序中执行了这样的程序:

```
cookie.setPath("idea_servlet_http_cookie_11_war_exploded/king");
```

这个Cookie就会和"idea_servlet_http_cookie_11_war_exploded/king"路径绑定在一起

11、--默认情况下,如果在创建Cookie后没有设置其有效时长,该Cookie被发送到客户端浏览器后是被存储在浏览器的缓存当中的,这样的Cookie是和浏览器共存亡的,浏览器被关闭就会被清除。当我们设置了Cookie的有效时间后, Cookie就会保存到浏览器的硬盘文件中,在有效时间内,如果不手动删除这个Cookie,这个Cookie不会受浏览器的关闭影响。有效期过后,该Cookie就失效了。

设置方法如下:

```
public void setMaxAge(int expiry) {  
    this.maxAge = expiry;  
}
```

需要注意的是:

设置的单位是秒seconds

Cookie有效时间 expiry = 0 表示Cookie直接被删除

Cookie有效时间 expiry < 0 表示Cookie不被存储

Cookie有效时间 expiry > 0 表示Cookie被存储到硬盘文件中

--存储在浏览器客户端的硬盘文件路径是打开浏览器的工具->Internet选项->常规->设置-查看文件

12、--当浏览器发送Cookie给服务器,服务器怎么接收呢?

需要用到HttpServletRequest接口中的getCookies()方法

```
Cookie[] cookies = request.getCookies();
```

```
Cookie[] cookies = request.getCookies();  
if (cookies != null){  
    for (Cookie cookie:  
        cookies) {  
        System.out.println(cookie.getName()+" = "+cookie.getValue());  
    }  
}
```

13、--浏览器是可以禁用Cookie的

在Internet选项中选择隐私进行设置,这是IE浏览器的设置。

意思是服务器发送过来的Cookie被浏览器拒收了

这只是浏览器不接收Cookie,服务器该发送的还是会发送。

创建web项目idea-servlet-servlet-cookie-11进行服务器创建Cookie对象然后向浏览器发送Cookie对象

web.xml文件:

```
<?xml version="1.0" encoding="UTF-8"?>  
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"  
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

        xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
        version="4.0">
        <!--设置欢迎页面，-->
        <welcome-file-list>
            <welcome-file>html/cookie.html</welcome-file>
        </welcome-file-list>

        <servlet>
            <servlet-name>cookie</servlet-name>
            <servlet-class>com.servlet.http.CreateAndSendCookieToBrowser</servlet-
class>
        </servlet>
        <servlet-mapping>
            <servlet-name>cookie</servlet-name>
            <url-pattern>/test/createAndSendCookieToBrowser</url-pattern>
        </servlet-mapping>
    </web-app>

```

cookie.html文件

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>cookie page</title>
</head>
<body>
    <a
href="/idea_servlet_http_cookie_11_war_exploded/test/creatAndSendCookieToBrowser
">服务器创建Cookie并发送到浏览器1</a>
</body>
</html>

```

CreateAndSendCookieToBrowser.java

```

package com.servlet.http;

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;

public class CreateAndSendCookieToBrowser extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

        //创建Cookie对象
        Cookie cookie1 = new Cookie("username", "zhangsan");
        Cookie cookie2 = new Cookie("password", "123");

        //绑定Cookie的请求路径
    }
}

```

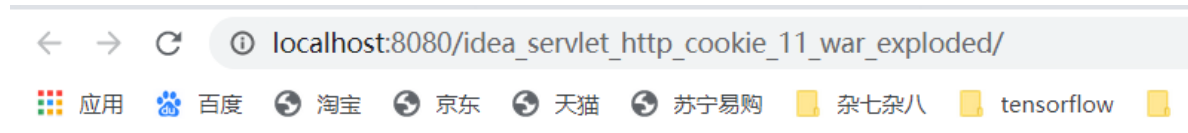
```

        cookie1.setPath(request.getContextPath()+"/king");
        cookie2.setPath(request.getContextPath()+"/king");

        //将Cookie发送给浏览器客户端，到浏览器肯定是响应，所以用response
        response.addCookie(cookie1);
        response.addCookie(cookie2);
    }
}

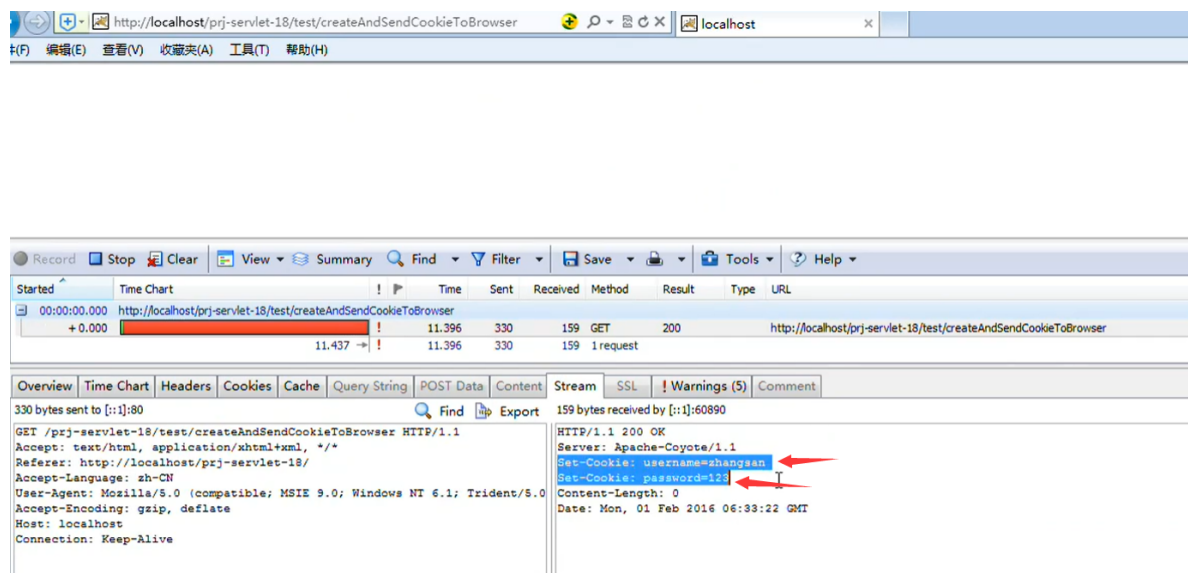
```

部署项目启动服务器，跳转到欢迎页面如下：



[服务器创建Cookie并发送到浏览器1](#)

打开HttpWatch，开始监测，点击链接，监测结果如下



可以看到响应协议中，有两个Cookie对象被响应而发送了过来，此时这两个Cookie被默认保存到了浏览器的缓存中

再测试Cookie和请求路径的绑定，下面截图中是课程中的，部署的项目名不一样。

上面我们用http://localhost:8080/idea_servlet_http_cookie_11_war_exploded/test/createAndSendCookieToBrowser这样的请求路径给客户端浏览器发送了Cookie，该Cookie被保存到浏览器中

此时我们用这样的路径：http://localhost:8080/idea_servlet_http_cookie_11_war_exploded/test/a再次请求，截图的监测结果如下


```
</web-app>
```

← → ↻ ⓘ localhost:8080/idea_servlet_http_cookie_11_war_exploded/

应用 百度 淘宝 京东 天猫 苏宁易购 杂七杂八 tensorflow 工具

[服务器创建Cookie并发送到浏览器1](#)

[服务器创建Cookie并发送到浏览器2](#)

我们打开HttpWatch打开监测

将请求路径: http://localhost:8080/idea_servlet_http_cookie_11_war_exploded/a进行刷新

看到监测结果在中请求协议部分并没有发送Cookie

将请求路径改为: http://localhost:8080/idea_servlet_http_cookie_11_war_exploded/

请求协议中发送了Cookie。

将请求路径改为: http://localhost:8080/idea_servlet_http_cookie_11_war_exploded

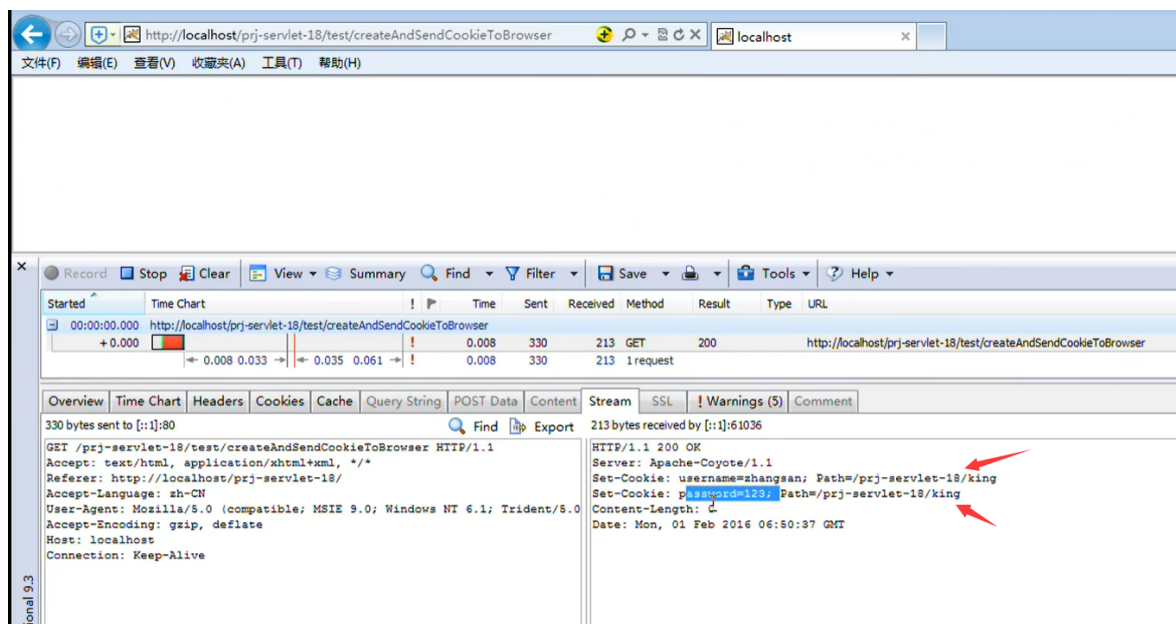
请求协议中没有发送Cookie。

所以说http://localhost:8080/idea_servlet_http_cookie_11_war_exploded/a路径发送的Cookie绑定的路径是:

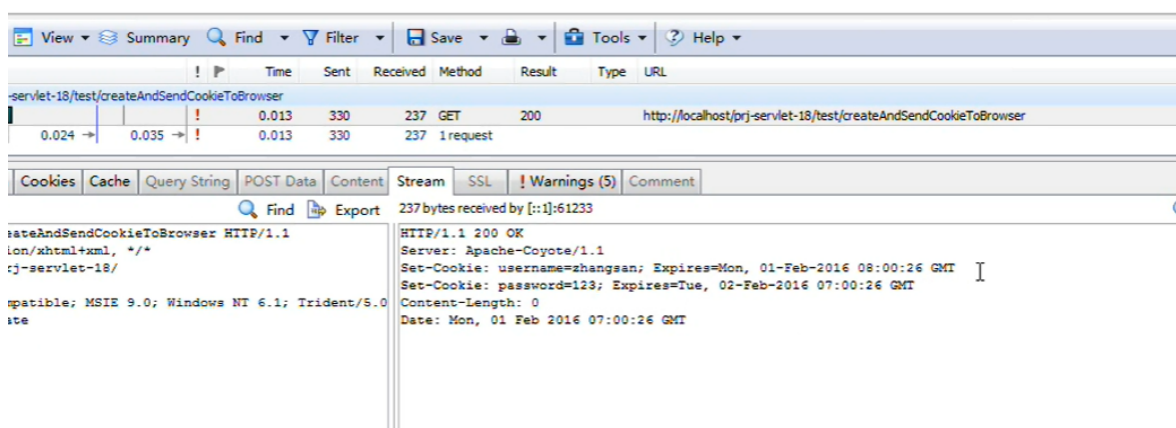
[idea_servlet_http_cookie_11_war_exploded/](http://localhost:8080/idea_servlet_http_cookie_11_war_exploded/)

默认的绑定路径规律像是发送这个Cookie的请求路径去掉最后面的路径名, 这只是猜测。

在绑定路径后我们通过监测请求路径, 从响应协议中可以看到Cookie所绑定的关联路径, 如下图:



设置有效时长后监测到的响应协议中看到Cookie的有效截至时间，如下图



下面测试浏览器提交Cookie到服务器，服务器接收

cookie.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>cookie page</title>
</head>
<body>
  <a
href="/idea_servlet_http_cookie_11_war_exploded/test/createAndSendCookieToBrowse
r">服务器创建Cookie并发送到浏览器1</a>
  <br>
  <a href="/idea_servlet_http_cookie_11_war_exploded/a">服务器创建Cookie并发送到
浏览器2</a>
  <br>
  <a
href="/idea_servlet_http_cookie_11_war_exploded/test/browserSendCookieToServer">
浏览器发送Cookie到服务器</a>
</body>
```



```
</html>
```

web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
    version="4.0">
    <!--设置欢迎页面，-->
    <welcome-file-list>
        <welcome-file>html/cookie.html</welcome-file>
    </welcome-file-list>

    <servlet>
        <servlet-name>cookie</servlet-name>
        <servlet-class>com.servlet.http.CreateAndSendCookieToBrowser</servlet-
class>
    </servlet>
    <servlet-mapping>
        <servlet-name>cookie</servlet-name>
        <url-pattern>/test/createAndSendCookieToBrowser</url-pattern>
        <url-pattern>/a</url-pattern>
    </servlet-mapping>

    <servlet>
        <servlet-name>bscts</servlet-name>
        <servlet-class>com.servlet.http.RecieveCookiesServlet</servlet-class>
    </servlet>
    <servlet-mapping>
        <servlet-name>bscts</servlet-name>
        <!--注意该路径中要添加test/与上面发送Cookie的路径绑定部分-->
        <url-pattern>/test/browserSendCookieToServer</url-pattern>
    </servlet-mapping>
</web-app>
```

RecieveCookiesServlet.java

```
package com.servlet.http;

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;

public class RecieveCookiesServlet extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {

        //从request对象中获取所有提交的Cookie，放进一个数组中
        Cookie[] cookies = request.getCookies();
```

```

        if (cookies != null){
            for (Cookie cookie:
                cookies) {
                System.out.println(cookie.getName()+" = "+cookie.getValue());
            }
        }
    }
}

```

CreateAndSendCookieToBrowser.java

```

package com.servlet.http;

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;

public class CreateAndSendCookieToBrowser extends HttpServlet {

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse
        response) throws ServletException, IOException {

        //创建Cookie对象
        Cookie cookie1 = new Cookie("username","zhangsan");
        Cookie cookie2 = new Cookie("password","123");

        //绑定Cookie的请求路径
        //      cookie1.setPath(request.getContextPath()+"/king");
        //      cookie2.setPath(request.getContextPath()+"/king");

        //设置Cookie有效时长
        cookie1.setMaxAge(60*60); //有效时长一个小时
        cookie2.setMaxAge(60*60*24); //有效时长一天

        //将Cookie发送给浏览器客户端，到浏览器肯定是响应，所以用response
        response.addCookie(cookie1);
        response.addCookie(cookie2);

    }
}

```

确保浏览器保存了Cookie，然后点击下图第三个链接，服务器窗口输出如下

← → ↻ ⓘ localhost:8080/idea_servlet_http_cookie_11_war_exploded/

应用 百度 淘宝 京东 天猫 苏宁易购 杂七杂八 tensorflow

[服务器创建Cookie并发送到浏览器1](#)

[服务器创建Cookie并发送到浏览器2](#)

[浏览器发送Cookie到服务器](#)

```
[2021-01-16 10:32:12,935] Artifact idea-servlet-http-cookie-11:war exploded: Deploy
username = zhangsan
password = 123
username = zhangsan
password = 123
Idea-d26a8271 = 2cd78248-20cb-4cc6-8552-456a1316158b
16-Jan-2021 22:32:22 511 信息 [Catalina-utility-1] org.apache.catalina.startup.Hov
```