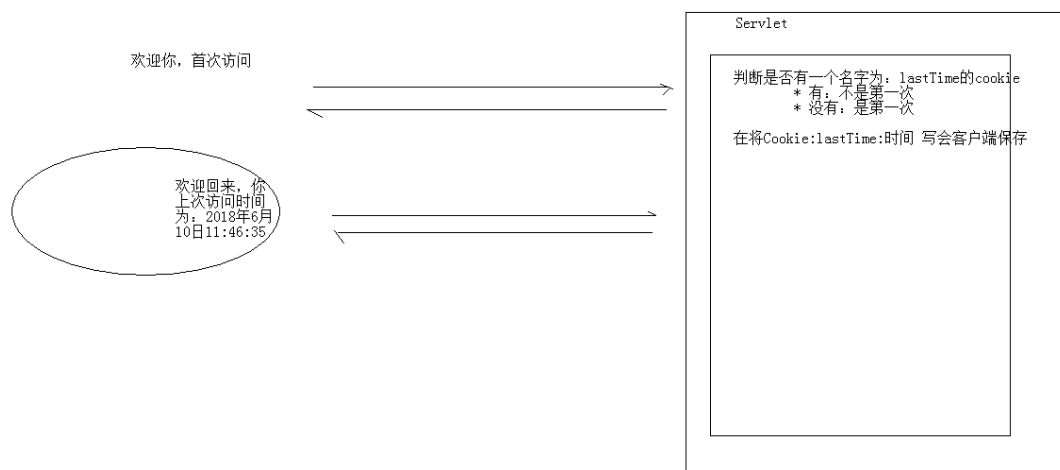


客户端请求cookiedemo1的资源，然后cookiedemo1响应后把cookie发送给客户端（使用的是api），发送的数据是msg=hello，响应头是set-cookie，set-cookie的值是msg=hello，http里响应头规定了如果浏览器收到了一个cookie头，浏览器会自动的将这个头里携带的数据msg=hello保存到客户端浏览器上，下次浏览器第二次再次发送请求的时候，它会将这个数据带过去的，即将数据放到请求的消息头里面，这个消息头叫做cookie，cookie头里面带的的数据是msg=hello，这都是浏览器自动做的事，带过去之后，在服务器里面来可以来获取这个请求头里面的数据，我们使用的是javaweb给我们封装好的api来做这个操作，不需要单独对这个请求头来操作，简化了开发步骤



会话技术

1. 会话技术，包括以下两个

1. **Cookie**
2. **Session**

注：我们先学**cookie**，再学**jsp**，最后再学**jsp**，因为**jsp**充当一个页面展示层的作用
资源有动态资源与静态资源，与**java**相关的动态资源有**servlet**与**jsp**

2. 会话：一次会话中包含多次请求和响应。

* 一次会话：浏览器第一次给服务器资源发送请求，会话建立，直到有一方断开为止

3. 功能：在一次会话的范围内的多次请求间，共享数据

4. 方式：

1. 客户端会话技术：**Cookie**
2. 服务器端会话技术：**Session**

Cookie:

1. 概念：客户端会话技术，将数据保存到客户端

2. 快速入门：

* 使用步骤：

1. 创建**Cookie**对象，绑定数据
* `new Cookie(String name, String value)`
2. 发送**Cookie**对象
* `response.addCookie(Cookie cookie)`
3. 获取**Cookie**，拿到数据
* `Cookie[] request.getCookies()`

注：不管是发送**cookie**还是获取**cookie**，最终它们的体现形式都是以**http**的协议来进行数据的交互，通过**http**协议的请求与完成

3. 实现原理

* 基于响应头**set-cookie**和请求头**cookie**实现

4. **cookie**的细节

1. 一次可不可以发送多个**cookie**?

* 可以

* 可以创建多个**Cookie**对象，使用**response**调用多次**addCookie**方法发送**cookie**即可。

2. **cookie**在浏览器中保存多长时间?

1. 默认情况下, 当浏览器关闭后 (浏览器关闭后内存会被释放), **Cookie**数据被销毁 (相当于**cookie**保存在浏览器的内存中)

2. 持久化存储 (即使浏览器被关闭后, **cookie**数据也会被保存在浏览器的内存中):

* **setMaxAge(int seconds)** (这个方法是**cookie**对象的一个方法)

1. 正数: 将**Cookie**数据写到硬盘的文件中。持久化存储。并指定**cookie**存活时间, 时间到后, **cookie**文件自动失效

2. 负数: 默认值。即**cookie**保存在浏览器的内存中, 浏览器关闭后**cookie**会被释放

3. 零: 删除**cookie**信息。即**cookie**即不存在于内存, 也不存在于硬盘

3. **cookie**能不能存中文?

* 在**tomcat 8** 之后, **cookie**中文数据与特殊字符 (比如空格) 建议使用URL编码存储, URL解码解析

4. **cookie**共享问题?

1. 假设在一个**tomcat**服务器中, 部署了多个**web**项目, 那么在哪些**web**项目中**cookie**能不能共享?

* 默认情况下**cookie**不能共享

* **setPath(String path)**: 设置**cookie**的获取范围。默认情况下, 设置当前的虚拟目录

* 如果要共享 (让当前服务器下 (指都是部署在**tomcat**服务器的8080端口下的项目) 部署的所有项目共享**Cookie**信息), 则可以将**path**设置为"/"

2. 不同的**tomcat**服务器间**cookie**共享问题?

不同过的服务器间也是可以共享**cookie**的, 但不用过多的去了解

比如百度官网的新闻与贴吧其实是属于不同服务器的, 但他们都得共享

cookie资源

<http://tieba.baidu.com/index.html>

<http://news.baidu.com/index.html>

baidu.com是一级域名, **tieba**是二级域名, **news**也是二级域名

* **setDomain(String path)**: 如果设置一级域名相同, 那么多个服务器之间**cookie**可以共享

* **setDomain(".baidu.com")**, 那么**tieba.baidu.com**和**news.baidu.com**中**cookie**可以共享

5. **Cookie**的特点和作用

1. **cookie**存储数据在客户端浏览器

2. 浏览器对于单个**cookie** 的大小一般限制是4kb 以及 对同一个域名下的总**cookie**数量也有限制 (20个)

*注: 不同的浏览器限制对于单个**cookie** 的大小不一样

* 作用:

1. **cookie**一般用于存出少量的不太敏感的数据

2. 在不登录的情况下, 完成服务器对客户端的身份识别

6. 案例：记住上一次访问时间

1. 需求：

1. 访问一个Servlet，如果是第一次访问，则提示：您好，欢迎您首次访问。
2. 如果不是第一次访问，则提示：欢迎回来，您上次访问时间为：显示时间字符串

2. 分析：

1. 可以采用Cookie来完成
2. 在服务器中的Servlet判断是否有一个名为lastTime的cookie

1. 有：不是第一次访问

1. 响应数据：欢迎回来，您上次访问时间为：2018年6月10日

11:50:20

2. 写回Cookie: lastTime=2018年6月10日11:50:01

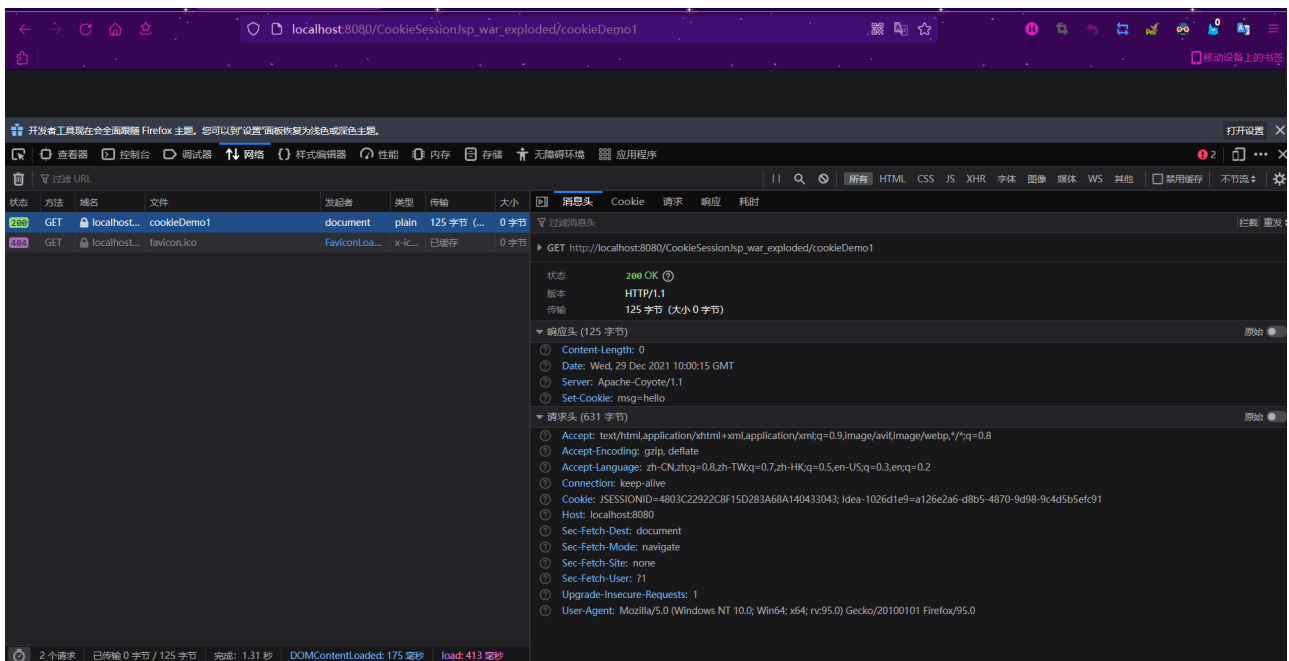
2. 没有：是第一次访问

1. 响应数据：您好，欢迎您首次访问
2. 写回Cookie: lastTime=2018年6月10日11:50:01

CookieSessionJsp项目

浏览器查看

- 打开firefox后，按f12，然后浏览器输入
localhost:8080/CookieSessionJsp_war_exploded/cookieDemo1,然后按回车,第一次发送请求，进行请求头与响应头的查看，响应头里会带一个set-cookie头，idea的控制台里无任何输出

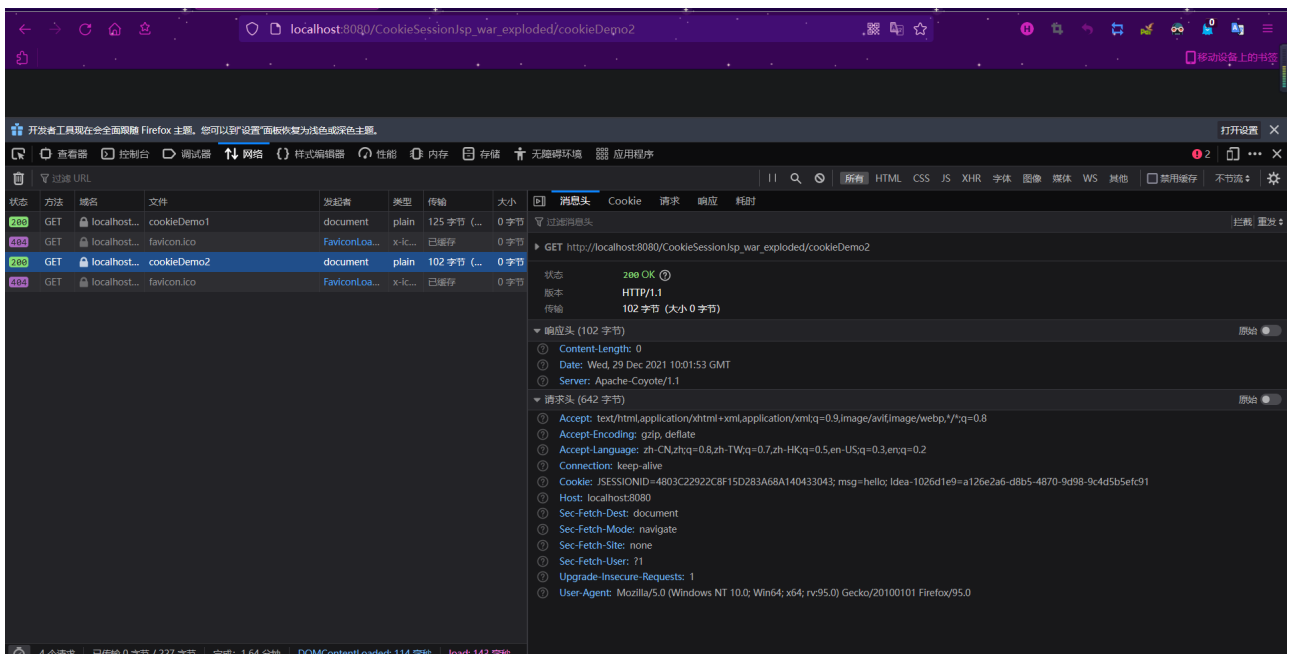


- 关闭f12，再打开发f12，然后浏览器输入
localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车，发现
请求头里有一个cookie头，响应头里什么都没有，控制台里输出以下：

JSESSIONID:DC9B2D19148D84E0D140810A95A1FAF0

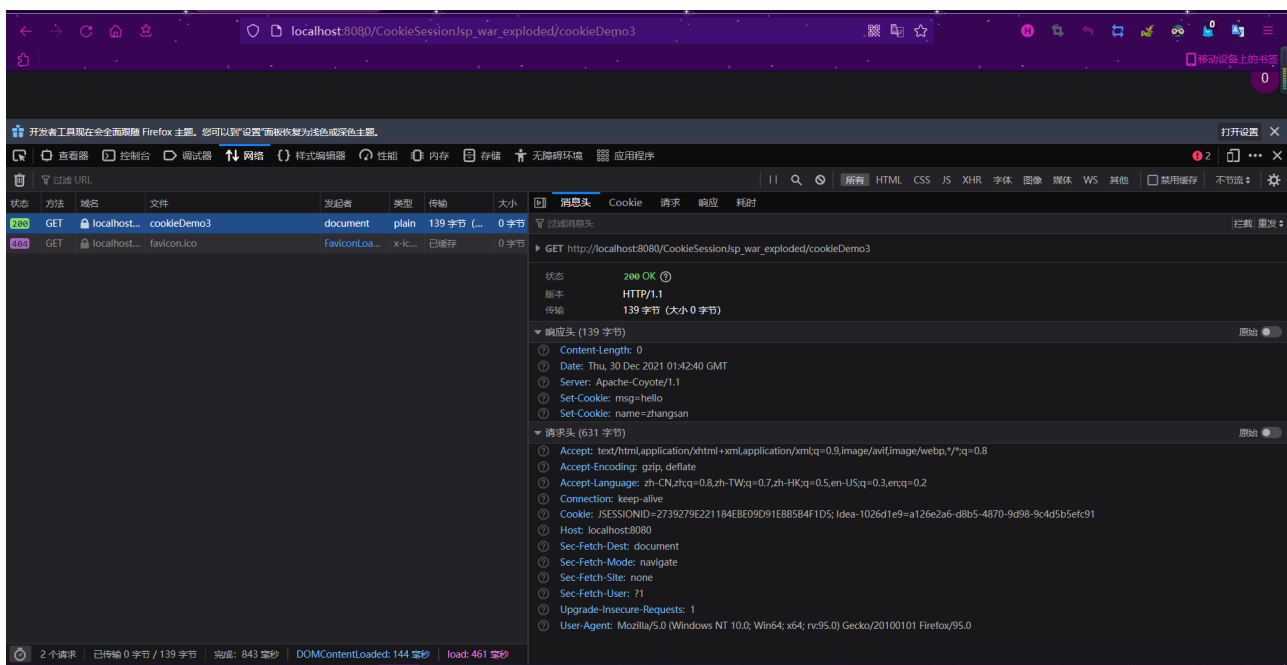
msg:hello

Idea-1026d1e9:a126e2a6-d8b5-4870-9d98-9c4d5b5efc91



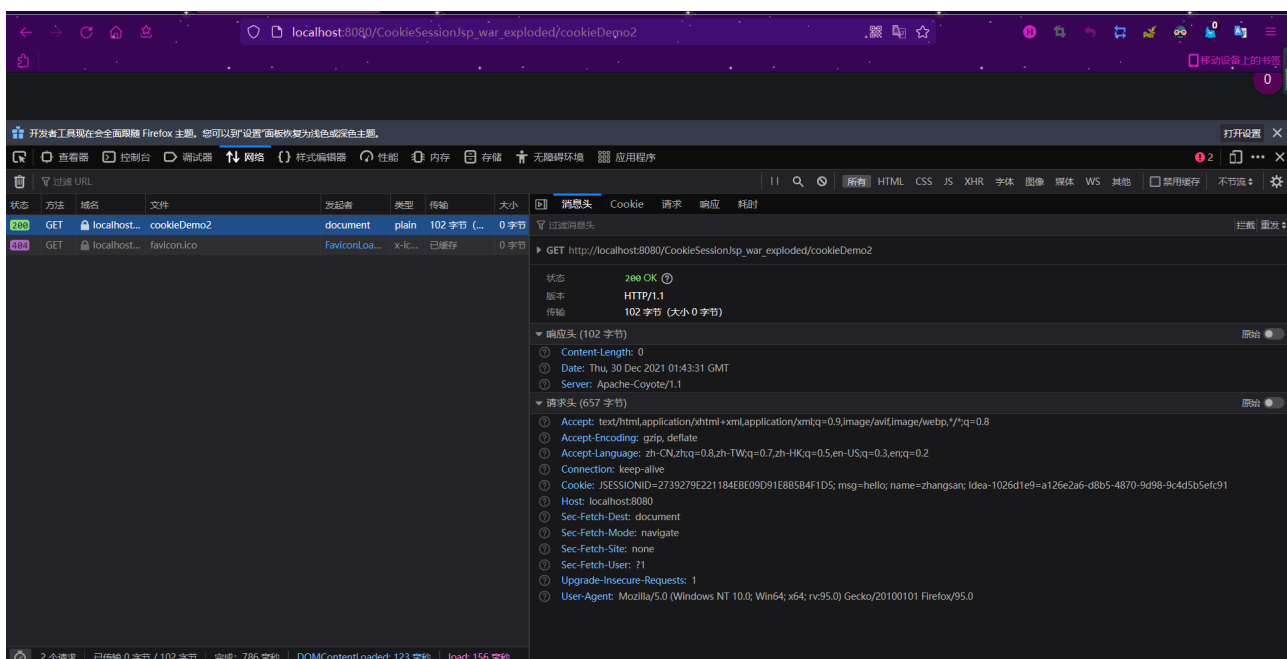
一次可以发送多个cookie

- 关闭tomcat后再启动tomcat后关闭f12，关闭浏览器，再打开发f12，然后浏览器
输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo3,然后按回车，
发现响应头里有两个Set-Cookie； idea的控制台没有任何输出

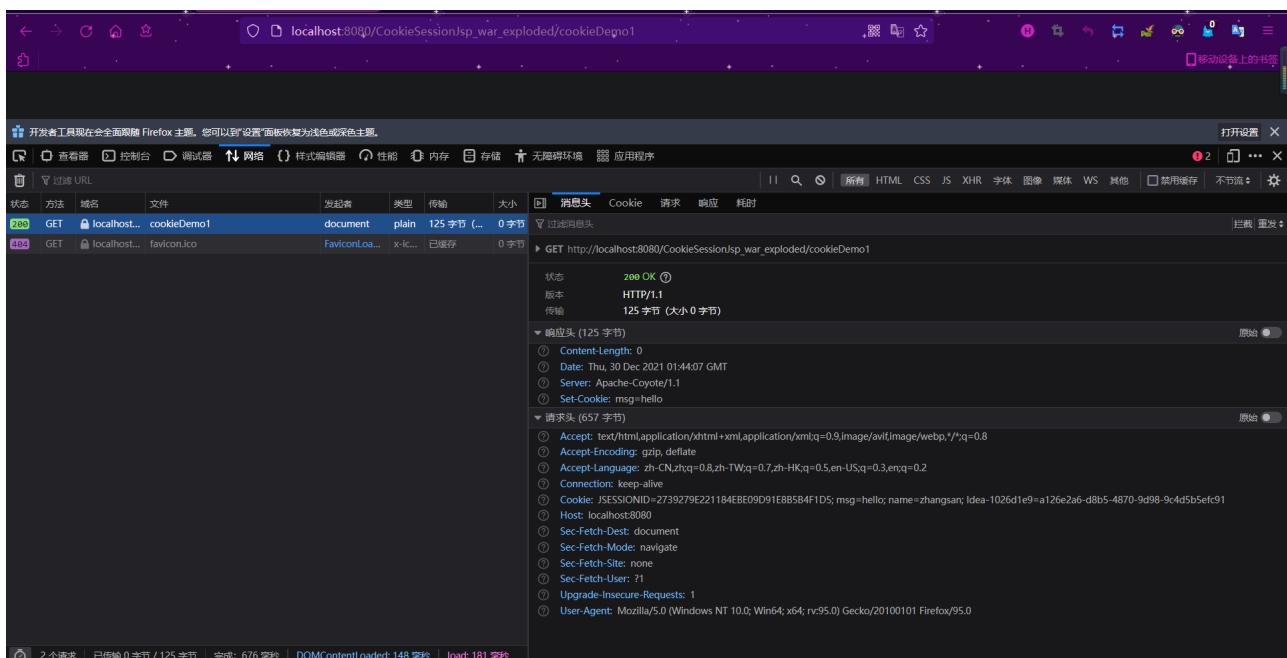


关闭f12，再打开发f12，然后浏览器输入
localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车，发发现请求头
里有两个Cookie（注：不是只有访demo2才能带过去cookie，而是在demo2里面我们获取了
这些cookie信息而已）；idea的控制台输出以下：

```
JSESSIONID:6A4A245BC0C3D71EF3B49EE5BB282158  
msg:hello  
name:zhangsan  
Idea-1026d1e9:a126e2a6-d8b5-4870-9d98-9c4d5b5efc91
```



关闭f12，再打开发f12，然后浏览器输入
localhost:8080/CookieSessionJsp_war_exploded/cookieDemo1（访问demo1也会带过去
cookie，只不过demo1我们没有区获取cookie，所以看不到控制台的打印效果）,然后按回
车；idea的控制台没有任何输出



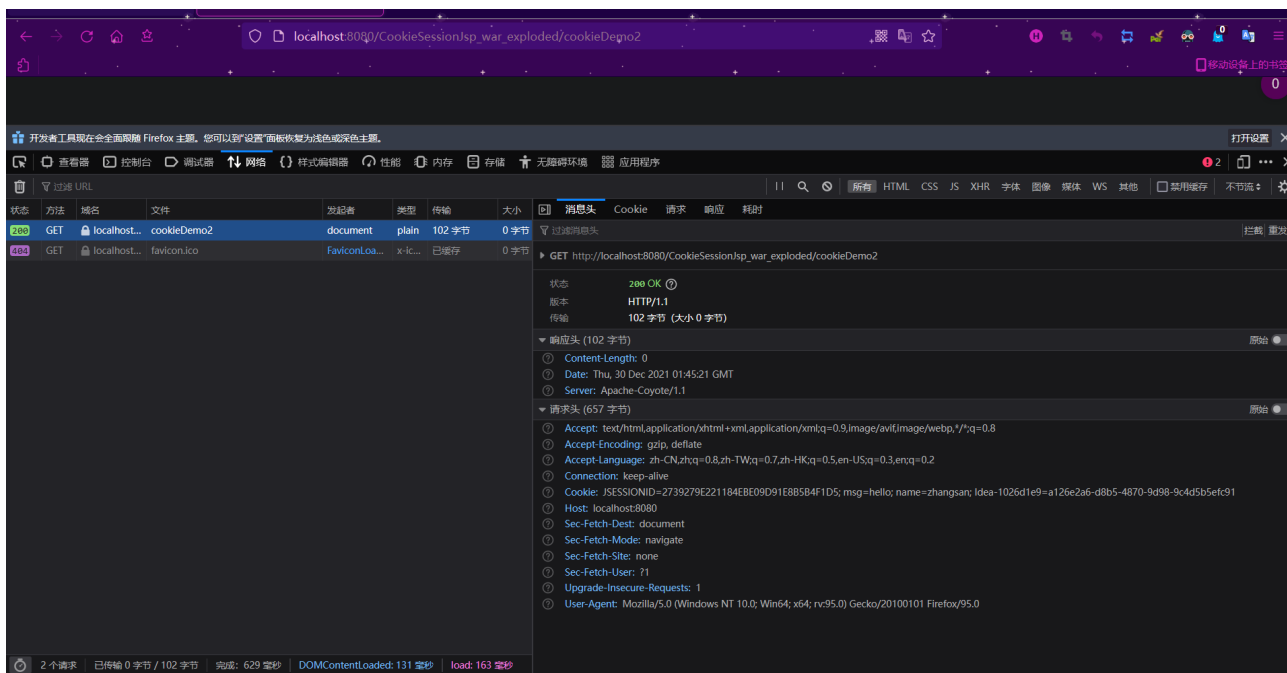
关闭f12，再打开发f12，然后浏览器输入
localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车idea的控制台输出以下(与上面的输出一样):

JSESSIONID:6A4A245BC0C3D71EF3B49EE5BB282158

msg:hello

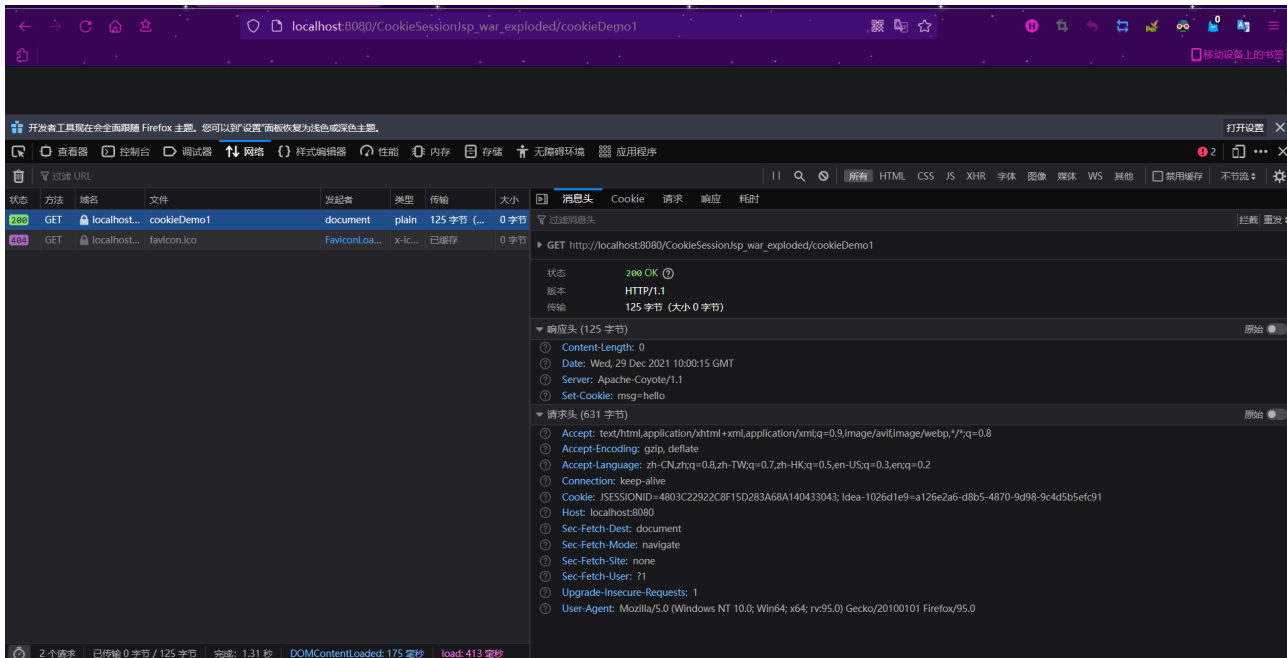
name:zhangsang

Idea-1026d1e9:a126e2a6-d8b5-4870-9d98-9c4d5b5efc91



浏览器关闭后Cookie数据被销毁

- 关闭浏览器，关闭tomcat后再开启tomcat，然后浏览器输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo1,然后按回车idea的控制台无输出

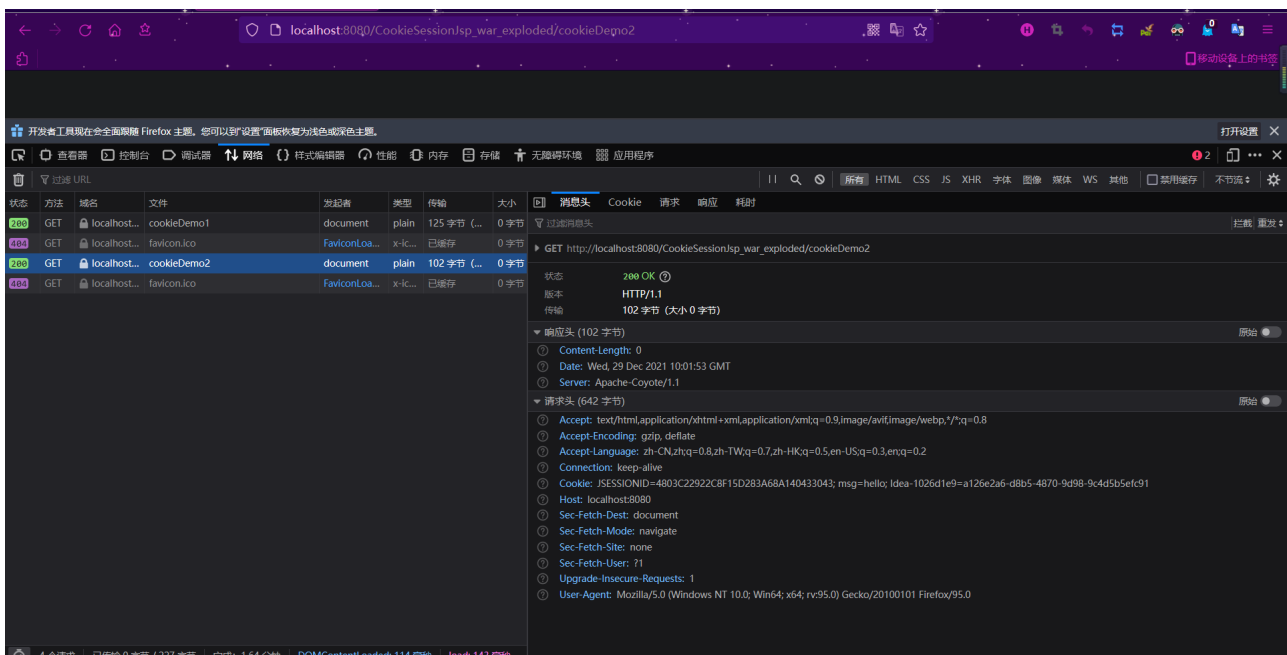


- 关闭f12，再打开发f12，然后浏览器输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车，idea控制台里输出以下：

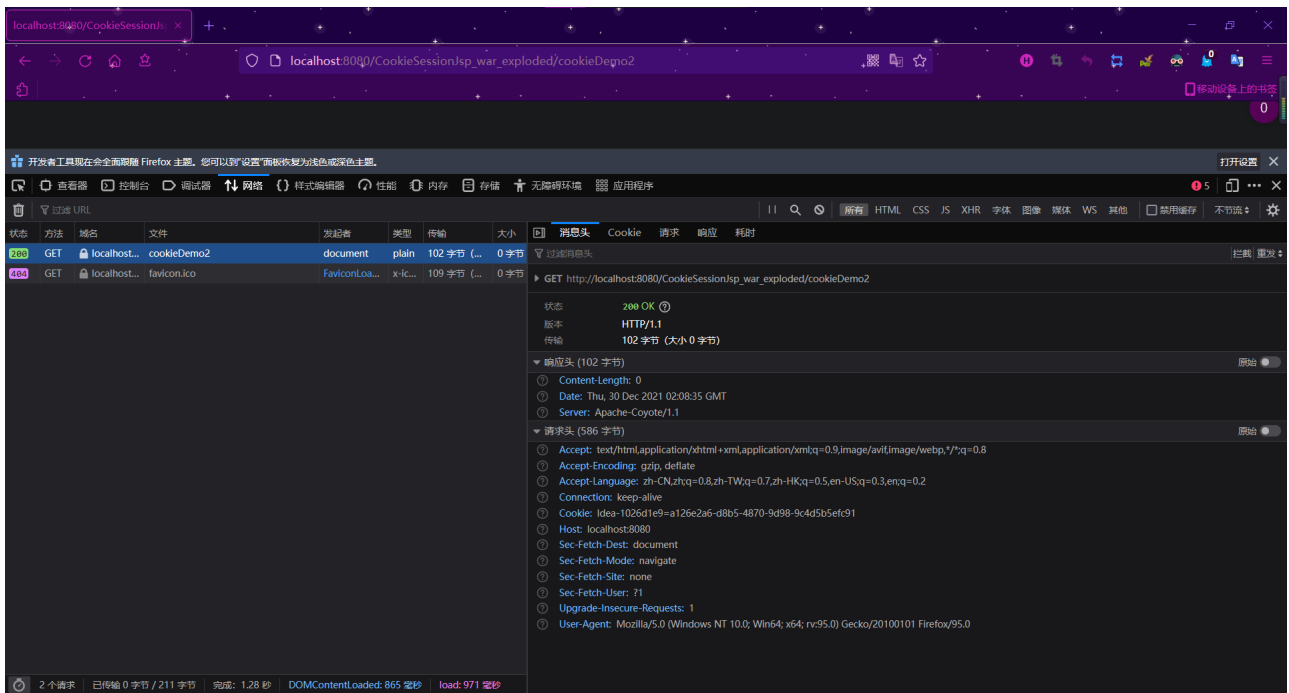
JSESSIONID:DC9B2D19148D84E0D140810A95A1FAF0

msg:hello

Idea-1026d1e9:a126e2a6-d8b5-4870-9d98-9c4d5b5efc91

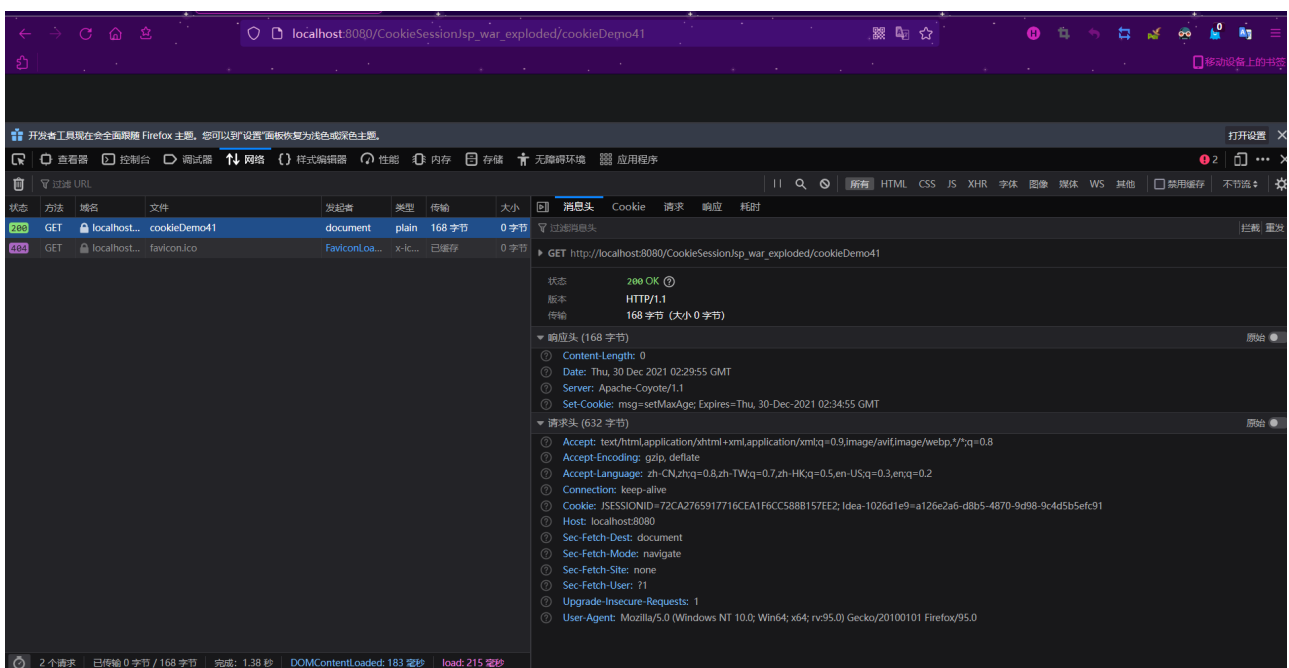


- 关闭浏览器，再打开发f12，然后浏览器输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车，idea控制台里无任何输出：



持久化储存（正数）

- 关闭浏览器，关闭tomcat后再开启tomcat，再打开f12，然后浏览器输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo41,然后按回车，idea控制台里无任何输出：

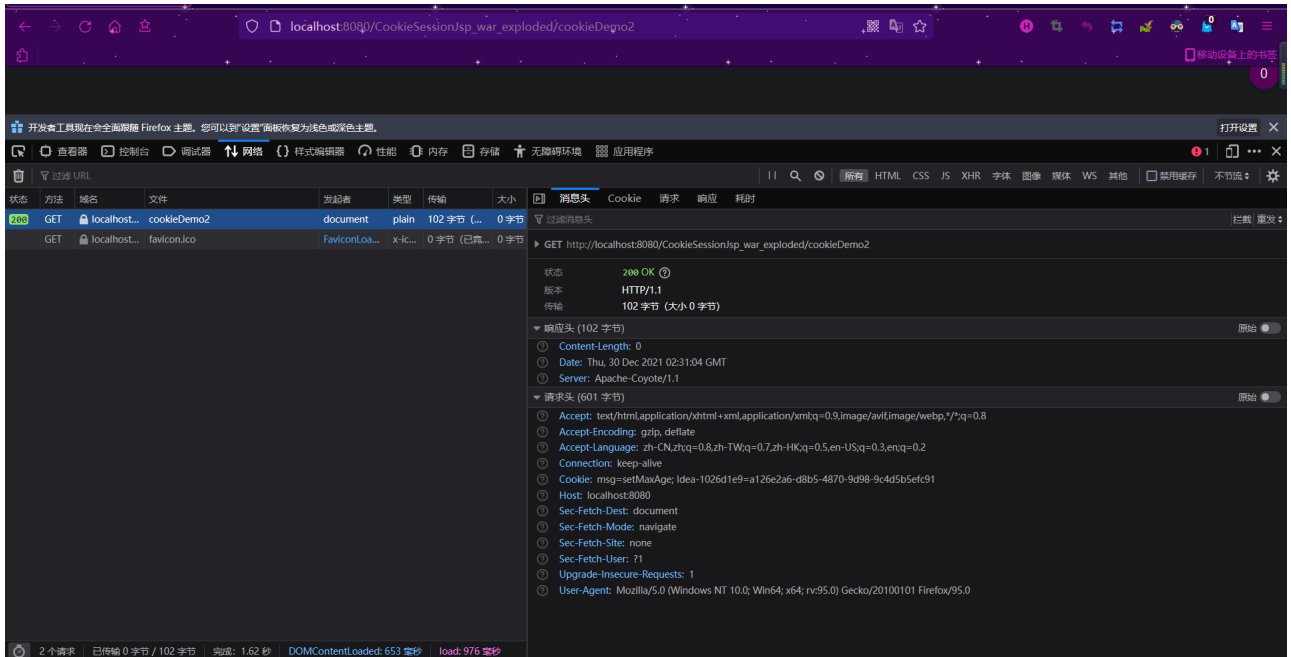


- 关闭浏览器，再打开f12，然后浏览器输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车，idea

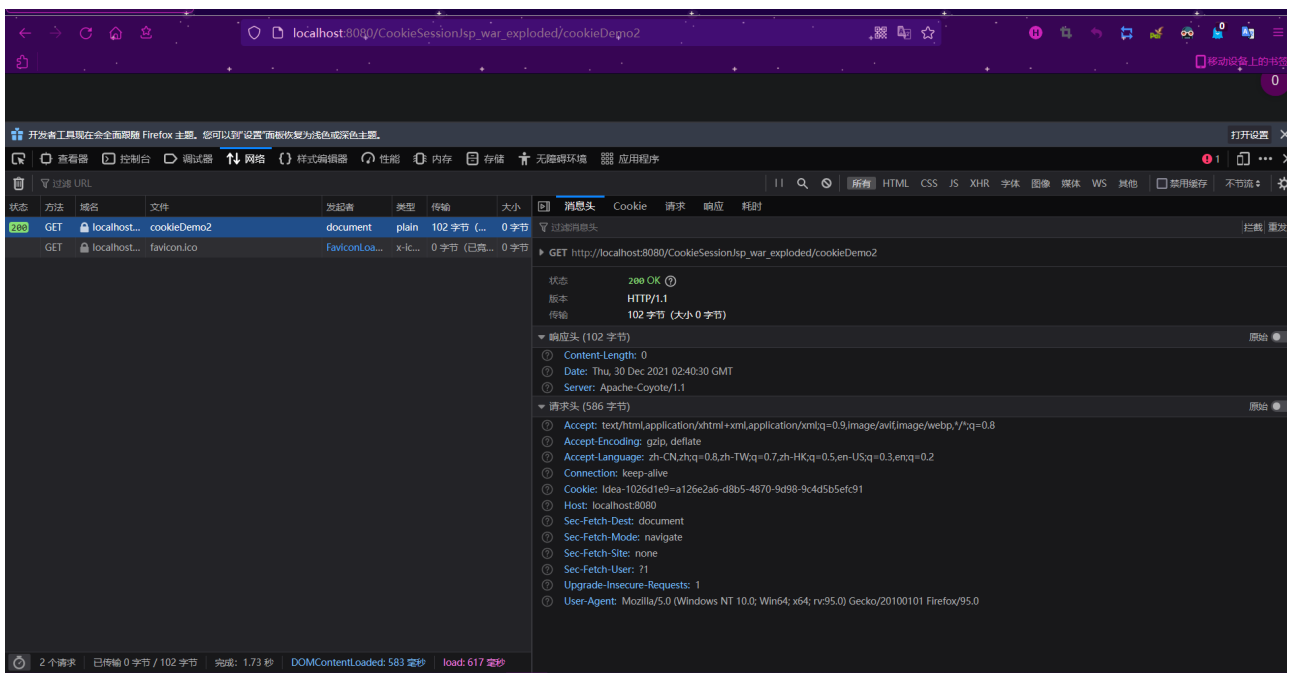
控制台里输出：

msg:setMaxAge

Idea-1026d1e9:a126e2a6-d8b5-4870-9d98-9c4d5b5efc91

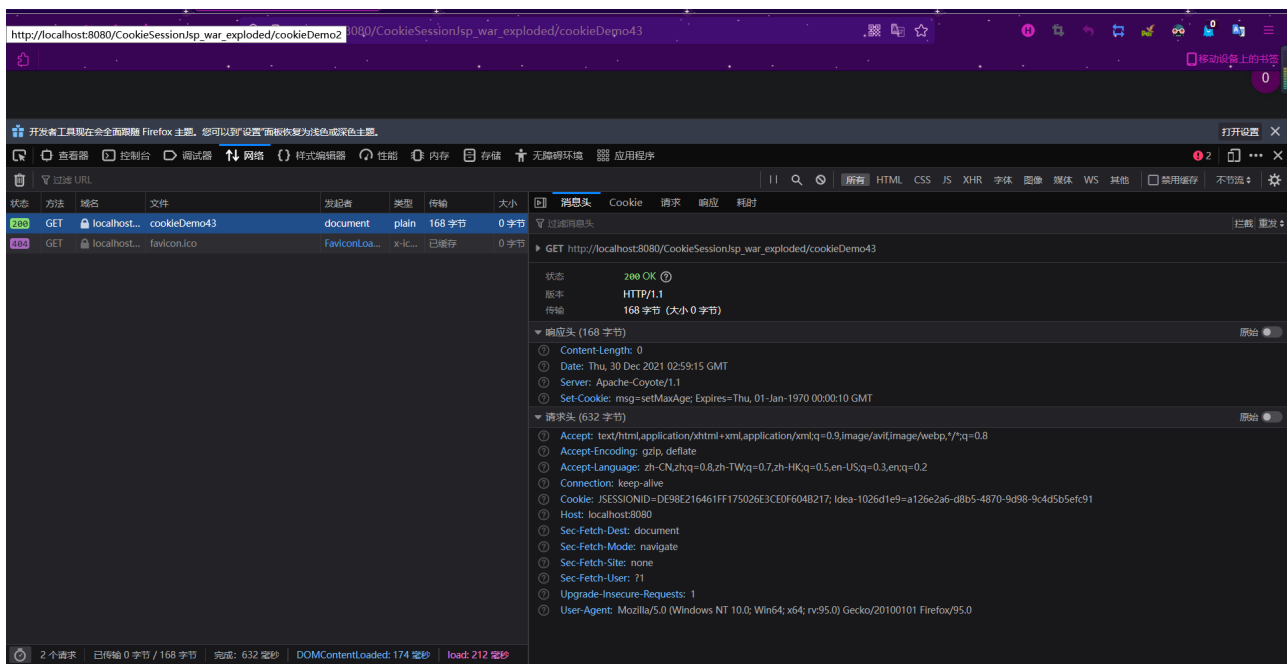


- 关闭浏览器，等300秒后，再打开f12，然后浏览器输入
localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车，idea
控制台无输出



持久化储存（0）

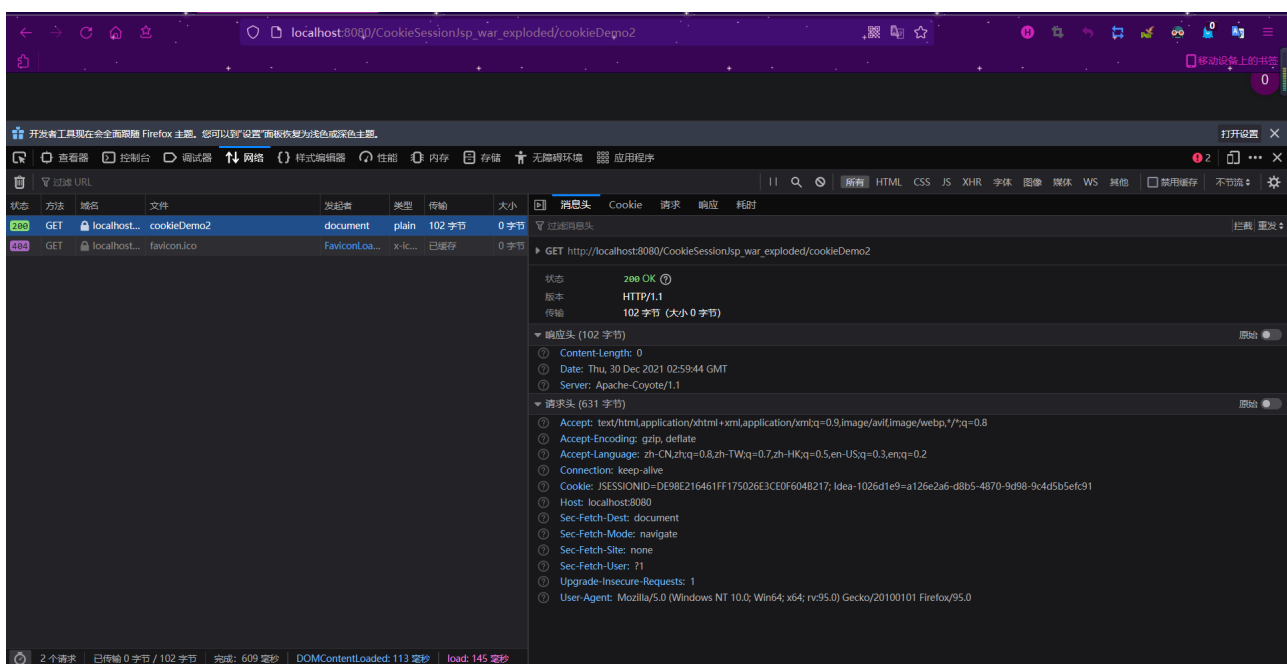
- 关闭浏览器，关闭tomcat后再开启tomcat，再打开f12，然后浏览器输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo43,然后按回车，idea控制台里无任何输出：



- 关闭发f12，再打开f12，然后浏览器输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车，idea控制台输出：

JSESSIONID:DE98E216461FF175026E3CE0F604B217

Idea-1026d1e9:a126e2a6-d8b5-4870-9d98-9c4d5b5efc91

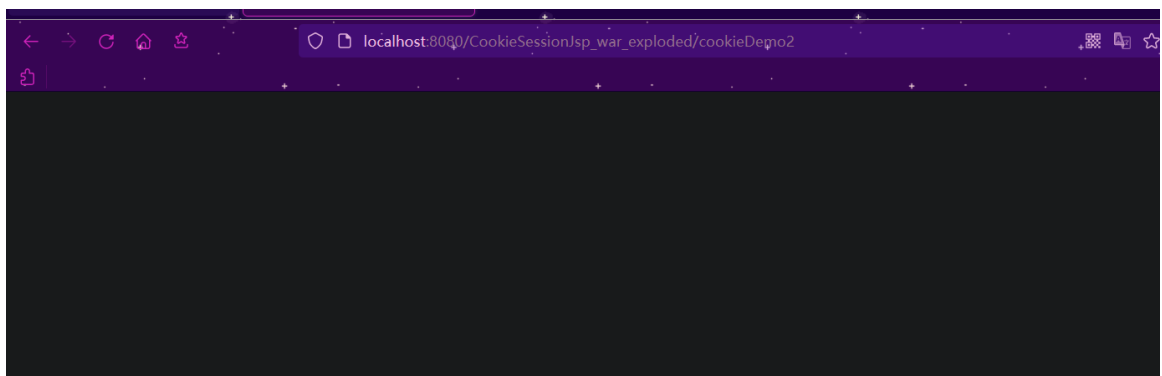
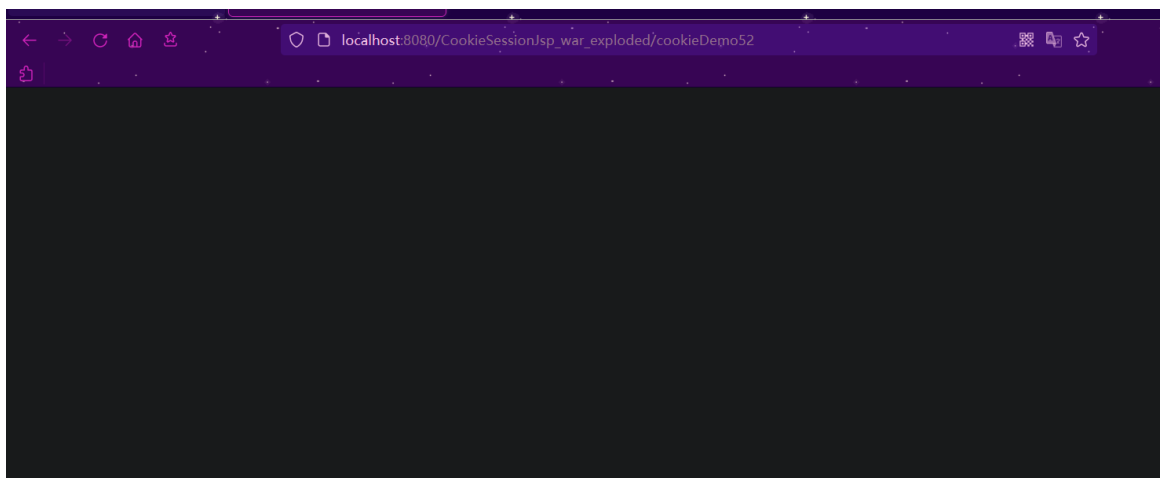


存储中文(编码后)

- 关闭浏览器，关闭tomcat后再开启tomcat，然后浏览器输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo52,然后按回车，网页还是白色的，且还停留在cookieDemo52网页，发现控制台，没有输出

然后浏览器再输入localhost:8080/CookieSessionJsp_war_exploded/cookieDemo2,然后按回车，网页还是白色的，且还停留在cookieDemo2网页，发现控制台，输出以下：

```
lastTime:2021%E5%B9%B412%E6%9C%8830%E6%97%A5+15%3A55%3A17
JSESSIONID:0DEE20D06315130107E482923B7CC1A5
msg:%E4%BD%A0%E5%A5%BD
Idea-1026d1e9:a126e2a6-d8b5-4870-9d98-9c4d5b5efc91
```



```

@WebServlet("/cookieDemo52")
public class CookieDemo52 extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        //URL编码
        String str = "你好";
        str = URLEncoder.encode(str, "utf-8");
        Cookie c1 = new Cookie("msg", str);

        response.addCookie(c1); // "你好"可以被正常的存储
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        this.doPost(request, response);
    }
}

```

存储中文（未编码）

- 关闭发f12，再打开f12，然后浏览器输入
localhost:8080/CookieSessionJsp_war_exploded/cookieDemo51,然后按回车，浏览器发现出错，控制台没有输出：

```

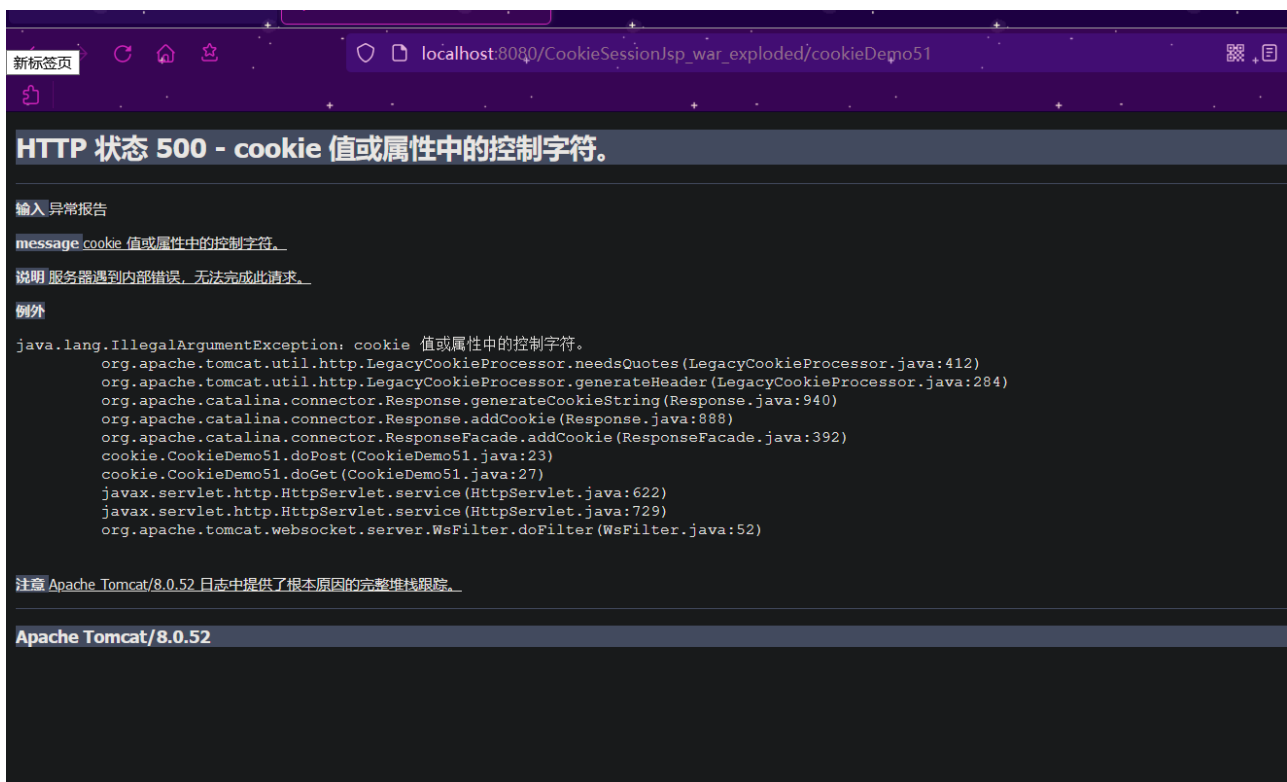
/*
 *
 */
@WebServlet("/cookieDemo51")
public class CookieDemo51 extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        Cookie c1 = new Cookie("msg", "你好");

        response.addCookie(c1); // "你好"可以被正常的存储
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        this.doPost(request, response);
    }
}

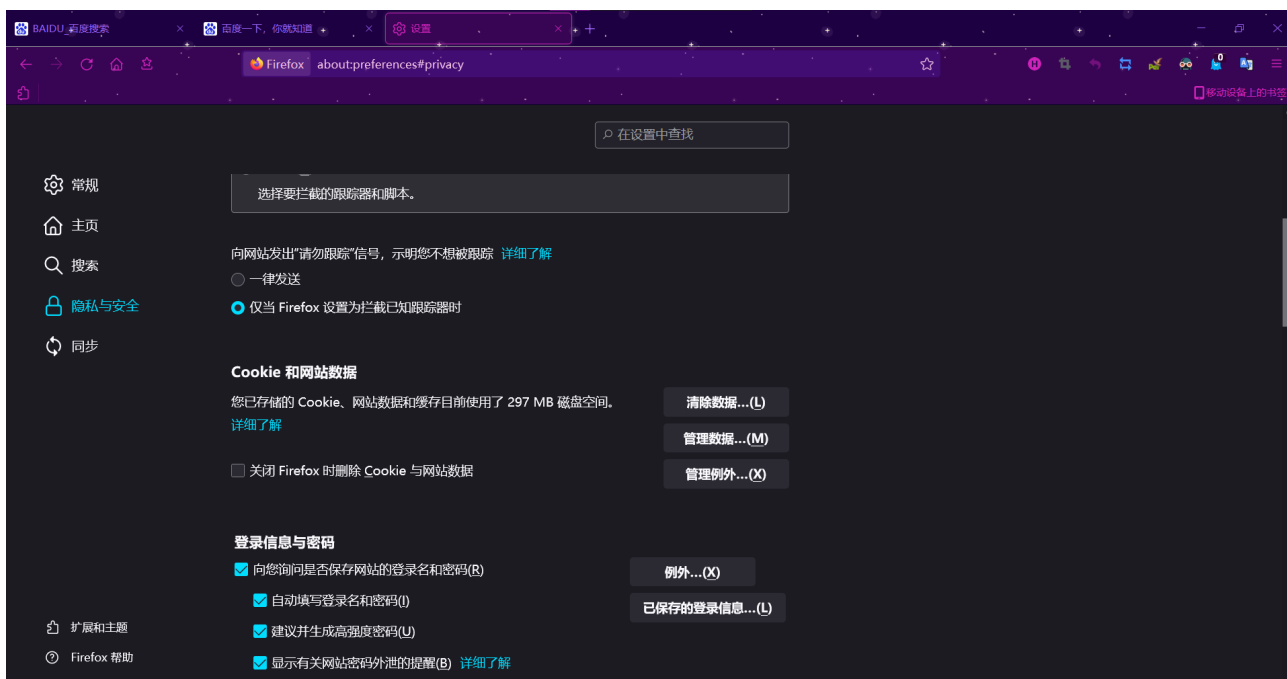
```



查看 c o o k i e 数量

- 随便登录一个网站,进入设置，点击管理数据

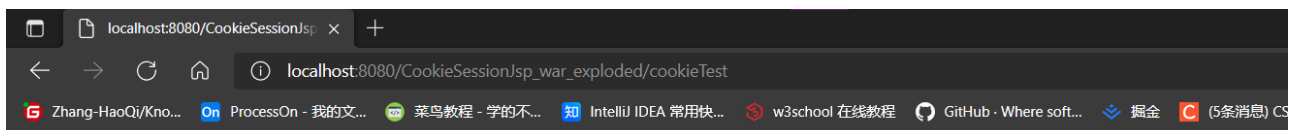
站点即域名，中间的数字就是指 c o o k i e 的数量（这里的都是硬盘里的持久化存储）





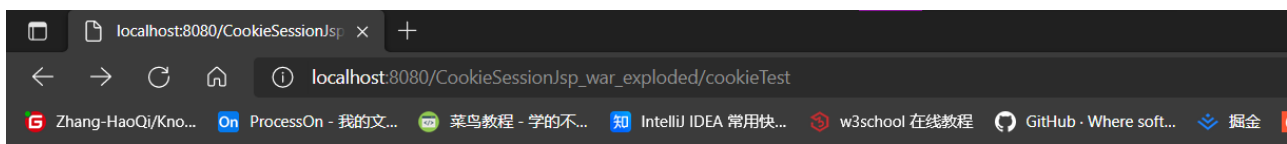
CookieTest案例

- 关闭mircsoft浏览器，再打开mircsoft浏览器，关闭tomcat再启动tomcat，登录 localhost:8080/CookieSessionJsp_war_exploded/cookieTest然后回车，查看网页



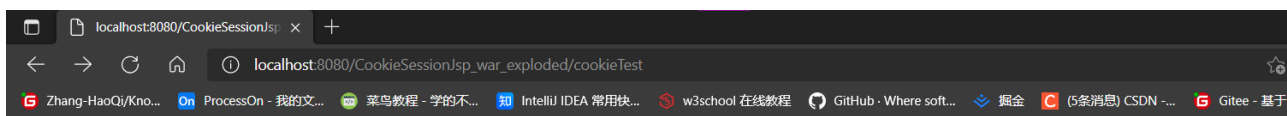
您好，欢迎您首次访问

- 第二次回车查看网页



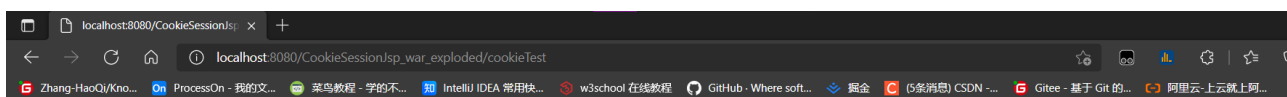
欢迎回来，您上次访问时间为:2021年12月31日 09:34:25

- 第三次回车查看网页



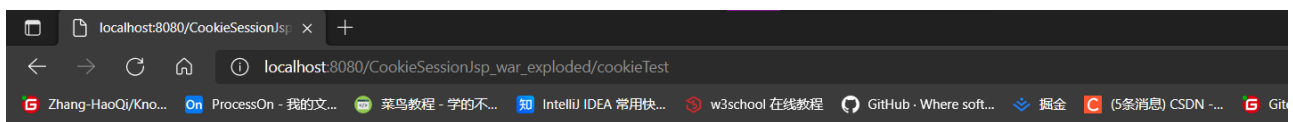
欢迎回来，您上次访问时间为:2021年12月31日 09:34:41

- 关闭mircsoft浏览器，再次输入网页，回车进行查看



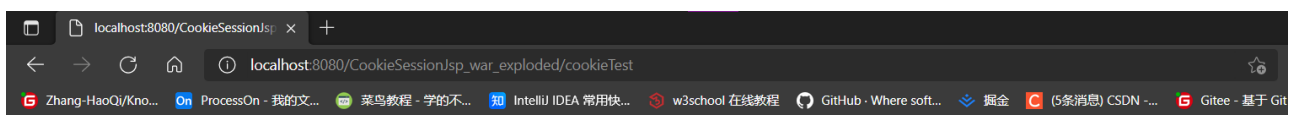
欢迎回来，您上次访问时间为:2021年12月31日 09:35:18

- 关闭tomcat，然后启动tomcat，输入后回车进行查看



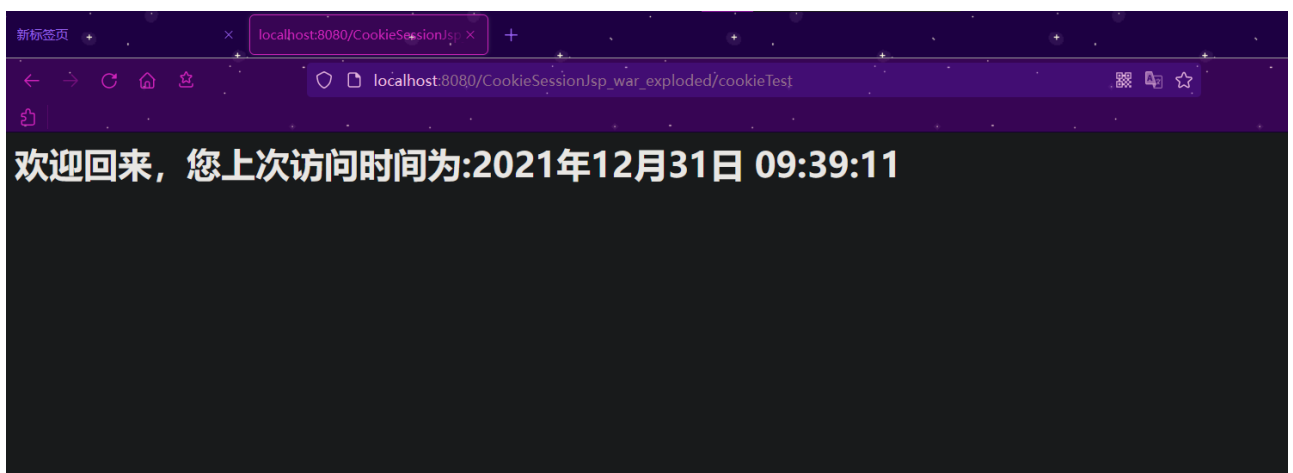
欢迎回来, 您上次访问时间为:2021年12月31日 09:38:09

- 关闭mircsoft浏览器, 然后打开mircsoft浏览器, 输入后回车进行查看



欢迎回来, 您上次访问时间为:2021年12月31日 09:38:40

- 关闭firefox, 然后打开firefox, 输入后回车进行查看



欢迎回来, 您上次访问时间为:2021年12月31日 09:39:11