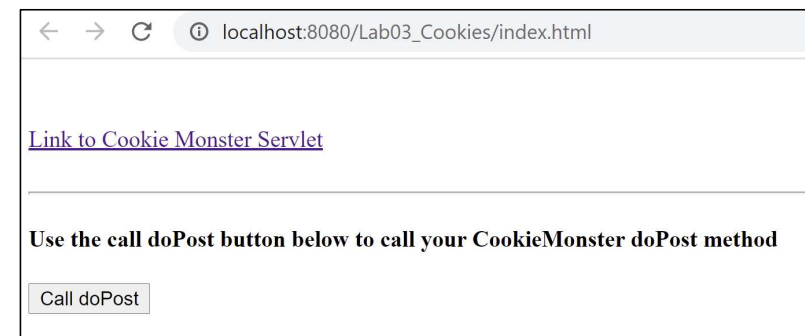


Lab 3 - Cookies

- Import the Lab03_Cookies_STARTER project
- Rename the project to “Lab03_Cookies”
- Examine the index.html file under WebContent
 - There are two links
 - The first will invoke the doGet method
 - The button will invoke the doPost method

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 <br><br>
9 <a href="http://localhost:8080/Lab2_3_Cookies/CookieMonster">Link to Cookie Monster Servlet</a>
10
11 <br><br><hr>
12 <h4>Use the call doPost button below to call your CookieMonster doPost method</h4>
13 <form method="post" action="CookieMonster">
14 <input type="submit" value="Call doPost" >
15 </form>
16
17
18 </body>
19 </html>
```



Lab 3 – Cookies

Review the link and the form to ensure you understand the methods they'll invoke

```
9 <a href="http://localhost:8080/Lab03_Cookies/CookieMonster">Link to Cookie Monster Servlet</a>
10
11 <br><br><hr>
12 <h4>Use the call doPost button below to call your CookieMonster doPost method</h4>
13 <form method="post" action="CookieMonster">
14 <input type="submit" value="Call doPost" >
15 </form>
```

Create a servlet, `com.servlet.example.CookieMonster`

- In `doGet`
 - Create a new `Cookie` object with the name "cookie1" and the value "myCookieValue"
 - Add the cookie to the response
 - Leave the default auto-generated response (see next line). Just so you know the page was served
`response.getWriter().append("Served at: ").append(request.getContextPath());`

Lab 3 – Cookies

Run your application and click on “Link to Cookie Monster Servlet”

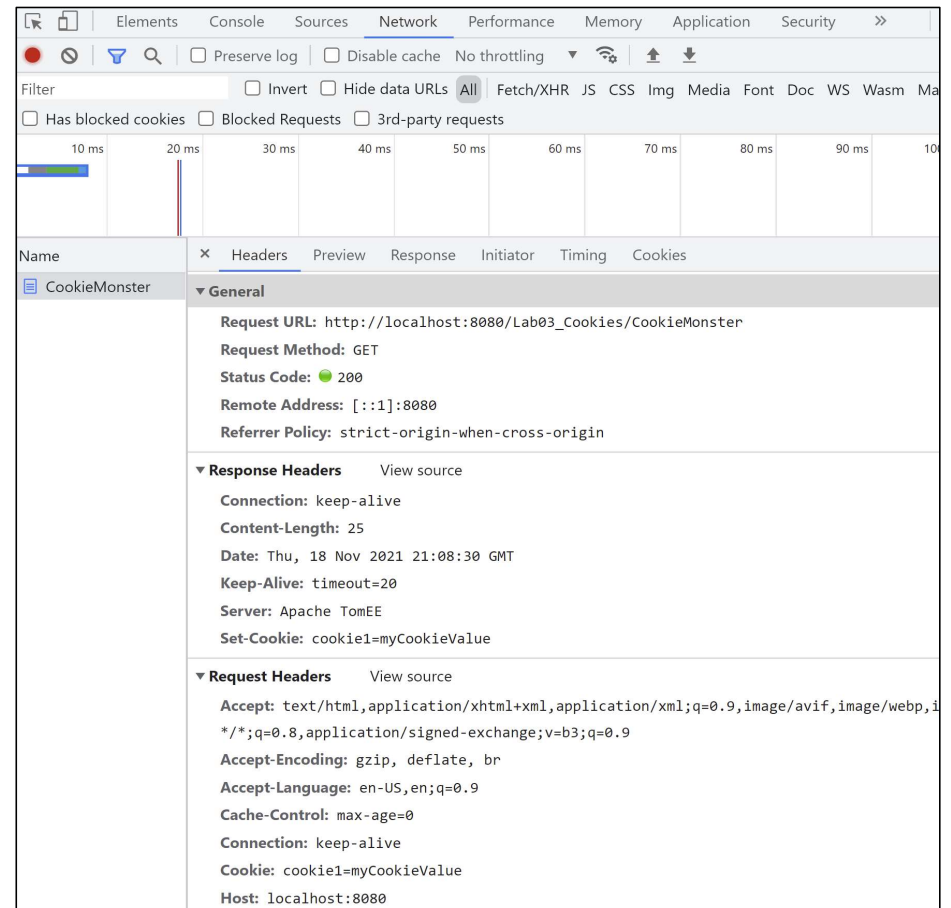


Lab 3 – Cookies

Examine the request and response headers

- You can use the TCP/IP monitor in Eclipse
- Alternately, if you open the page in Chrome, you can
 - Click the F12 key
 - Click on the “Network” tab
 - If you don’t see the name Cookie Monster, do Ctrl + R
 - Click on the name Cookie Monster

Examine the cookie in the response
and request Headers



The screenshot shows the Chrome DevTools Network tab with the 'CookieMonster' request selected. The 'Headers' sub-tab is active, displaying the following information:

- General**
 - Request URL: `http://localhost:8080/Lab03_Cookies/CookieMonster`
 - Request Method: `GET`
 - Status Code: `200`
 - Remote Address: `::1:8080`
 - Referrer Policy: `strict-origin-when-cross-origin`
- Response Headers**
 - Connection: `keep-alive`
 - Content-Length: `25`
 - Date: `Thu, 18 Nov 2021 21:08:30 GMT`
 - Keep-Alive: `timeout=20`
 - Server: `Apache TomEE`
 - Set-Cookie: `cookie1=myCookieValue`
- Request Headers**
 - Accept: `text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9`
 - Accept-Encoding: `gzip, deflate, br`
 - Accept-Language: `en-US,en;q=0.9`
 - Cache-Control: `max-age=0`
 - Connection: `keep-alive`
 - Cookie: `cookie1=myCookieValue`
 - Host: `localhost:8080`

Lab 3 – Cookies

Override doPost to do the following

- Get all the cookies from the request object
- Loop through the cookies and print out the value to the console
- Run the application again, this time click on the doPost button

Your output should be similar to this

