МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ "БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ" КАФЕДРА ИИТ

ОТЧЁТ

по лабораторной работе №4 «Cookies»

Выполнил:

Студент 2 курса группы ПО-9 Харитонович Захар Сергеевич

Проверил:

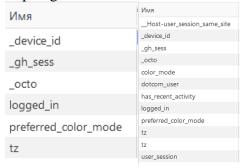
Михняев А. Л.

Ход работы

Задание:

- 1. Используя куки необходимо сделать счётчик уникальных посещений сайта.
- 2. Если пользователь сайта не уникальный (уже заходил с этого адреса), то необходимо отобразить ему дату последнего посещения сайта и браузер, с которого он заходил.
- 3. Если пользователь уникальный то поприветствовать его и записать в текстовый файл: дату посещения сайта, ір адрес, браузер.
- 4. Получить все атрибуты ServletContext, HttpSession, HttpServletRequest и вывести их на экран.
- 5. Создать репозиторий Git в который записать все лабораторные работы. Посмотреть какие куки и какие сессии создаются при входе в аккаунт Git. Что будет если их удалить или поменять значения?

При удалении этих куки произойдёт выход из аккаунта. https://github.com/ZakharTS/web_lab



Код программы:

Cookies.java:

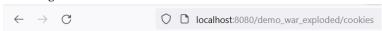
```
package com.example.demo;
import jakarta.servlet.Servlet;
import jakarta.servlet.ServletContext;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.*;
import java.io.IOException;
import java.io.PrintWriter;
import java.io.Writer;
import java.lang.reflect.Array;
import java.time.Instant;
import java.time.LocalDateTime;
import java.util.Arrays;
import java.util.Enumeration;
import java.util.concurrent.atomic.AtomicInteger;
@WebServlet(name = "cookies", value = "/cookies")
public class Cookies extends HttpServlet {
   private static final String BROWSER = "userBrowser";
   private static final String VISITED = "userVisited";
   private static final String IP ADDRESS = "userIP";
   private static final String DATE = "lastVisit";
    private static final String PATH = "/demo war exploded/cookies";
   private static final AtomicInteger counter = new AtomicInteger();
                                                         HttpServletResponse
            void
                    doGet(HttpServletRequest request,
    public
response) throws IOException {
        response.setContentType("text/html");
        Cookie[] cookies = request.getCookies();
        if (cookies != null) {
                         (Arrays.stream(cookies).anyMatch((cookie
            if
                                                                             ->
cookie.getName().equals(VISITED)))) {
```

```
Cookie browser = getCookie(cookies, BROWSER);
                Cookie ip = getCookie(cookies, IP ADDRESS);
                Cookie date = getCookie(cookies, DATE);
                addCookies(request, response);
                try (PrintWriter writer = response.getWriter()) {
                   writer.write("<h1>Welcome back!</h1>");
                   writer.write("<h4>Your last visit:</h4>");
                   writer.write("<b>IP: </b>" + ip.getValue() + "<br>");
                   writer.write("<b>Browser: </b>" + browser.getValue() +
"<br>");
                   writer.write("<b>Last visit date: </b>" + date.getValue()
+ "<br>>");
                   writer.write("<b>ServletContext:</b><br>"
this.getServletContext().ORDERED LIBS + "<br>"
                            + this.getServletContext().TEMPDIR + "<br>");
                   writer.write("<b>HttpSession: </b><br>");
                   Enumeration<String>
                                                   attributeNames
request.getSession().getAttributeNames();
                   if (attributeNames != null) {
                        while (attributeNames.hasMoreElements()) {
                            String attribute = attributeNames.nextElement();
                            writer.write(attribute
                                                     + ": "
request.getSession().getAttribute(attribute));
                            writer.write("<br>");
                   writer.write("<b>HttpServletRequest: </b><br>");
                   attributeNames = request.getAttributeNames();
                   if (attributeNames != null) {
                        while (attributeNames.hasMoreElements()) {
                           String attribute = attributeNames.nextElement();
                            writer.write(attribute
request.getAttribute(attribute));
                            writer.write("<br>");
                   }
                } catch (Exception e) {
                   System.out.println(e);
                }
            } else {
                addCookies(request, response);
                try (PrintWriter writer = response.getWriter()) {
                                                       User!</h1><h3>Creating
                   writer.write("<h1>Welcome,
cookies...</h3>");
                } catch (Exception e) {
                   System.out.println(e);
           }
        } else {
           addCookies(request, response);
            try (PrintWriter writer = response.getWriter()) {
               writer.write("<h1>Welcome,
                                                   User!</h1><br><h3>Creating
cookies...</h3>");
            } catch (Exception e) {
               System.out.println(e);
            }
        }
   private static Cookie getCookie(Cookie[] cookies, String name) {
        if (cookies == null) return null;
        try {
```

```
Arrays.stream(cookies).filter(cookie
                                                                           ->
           return
cookie.getName().equals(name)).findFirst().get();
       } catch (Exception e) {
           System.out.println(e); return null;
   }
   private
               static
                         void
                                   addCookies(HttpServletRequest
HttpServletResponse response) {
       response.addCookie(createCookie(BROWSER,
                                                    request.getHeader("user-
agent").split(("\\s"))[0], PATH));
       response.addCookie(createCookie(IP ADDRESS,
getClientIpAddress(request), PATH));
       response.addCookie(createCookie(DATE,
                                                   Instant.now().toString(),
PATH));
       response.addCookie(createCookie(VISITED, "true", PATH));
   }
   private static Cookie createCookie(String name, String value, String
path) {
       Cookie cookie = new Cookie (name, value);
       cookie.setPath(path);
       cookie.setHttpOnly(true);
       cookie.setMaxAge(-1);
       return cookie;
   private static final String[] HEADERS TO TRY = {"X-Forwarded-For",
                                                      "HTTP X FORWARDED FOR",
"Proxy-Client-IP",
                    "WL-Proxy-Client-IP",
                        "HTTP_X_CLUSTER_CLIENT_IP",
                                                      "HTTP CLIENT IP",
"HTTP X FORWARDED",
"HTTP FORWARDED FOR", "HTTP FORWARDED", "HTTP VIA", "REMOTE ADDR"};
   private static String getClientIpAddress(HttpServletRequest request) {
       for (String header : HEADERS TO TRY) {
           String ip = request.getHeader(header);
                         ! =
                                     && ip.length()
                 (ip
                               null
                                                                           & &
!"unknown".equalsIgnoreCase(ip)) {
               return ip;
       return request.getRemoteAddr();
   }
Результат работы:
                   localhost:8080/demo_war_exploded/cookies
```

Welcome, User!

Creating cookies...



Welcome back!

Your last visit:

IP: 127.0.0.1 Browser: Mozilla/5.0

Last visit date: 2023-05-19T10:12:44.443328200Z

ServletContext:

jakarta.servlet.context.orderedLibs jakarta.servlet.context.tempdir HttpSession: HttpServletRequest: