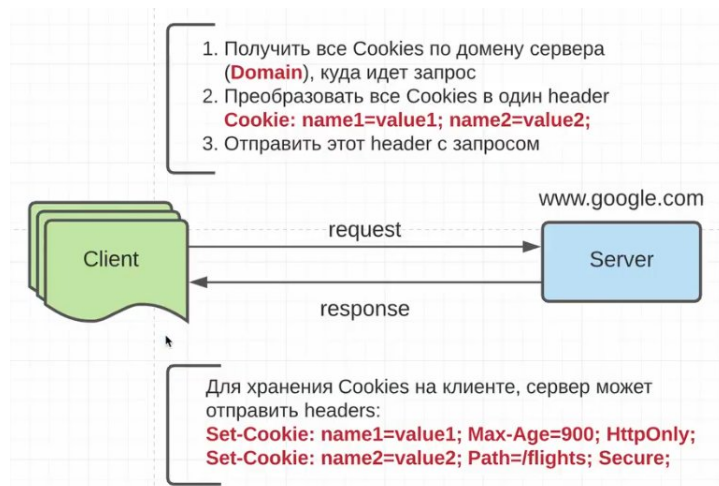
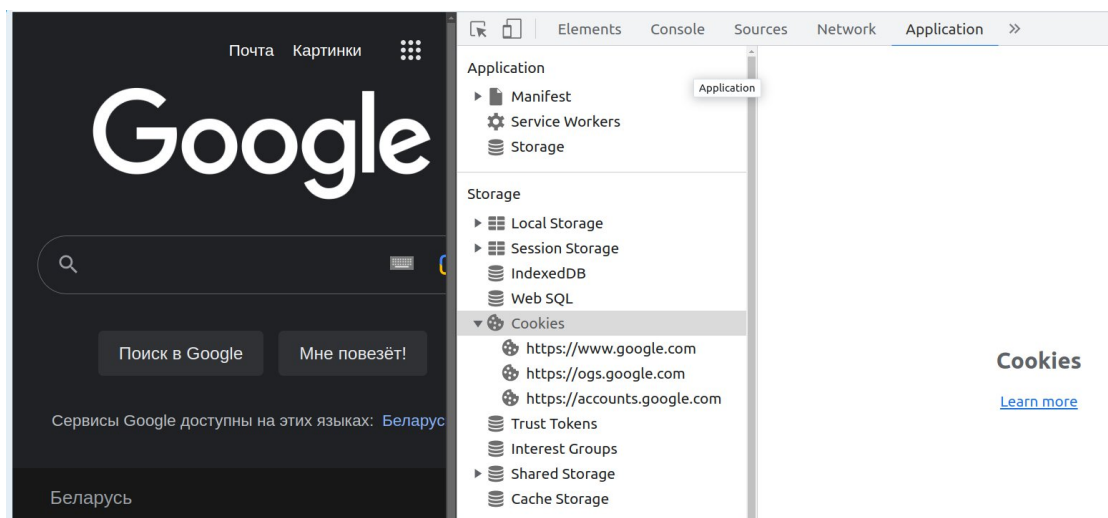


Часть 1. Теоретическая.



Cookies

https://developer.mozilla.org/ru/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server



Cookies

[Learn more](#)

Application

- Manifest
- Service Workers
- Storage
 - Local Storage
 - Session Storage
 - IndexedDB
 - Web SQL
 - Cookies**
 - <https://www.google.com>
 - <https://ogs.google.com>
 - <https://accounts.google.com>
 - Trust Tokens
 - Interest Groups
 - Shared Storage
 - Cache Storage

Background Services

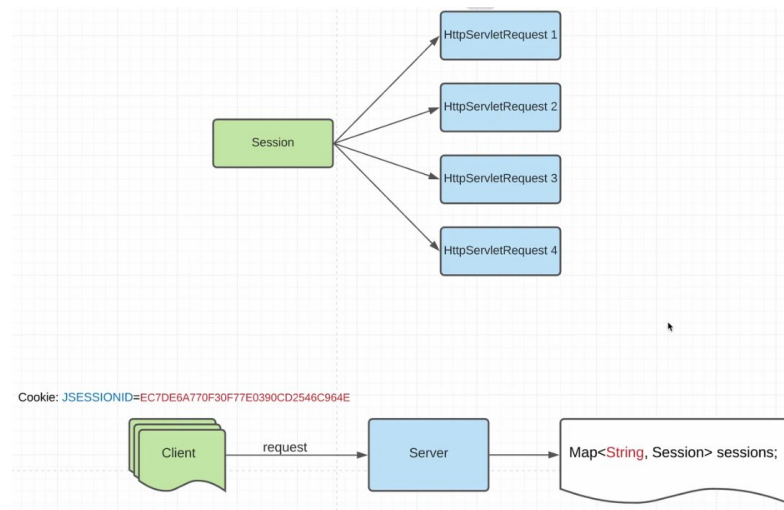
- Back/forward cache
- Background Fetch
- Background Sync

Filter

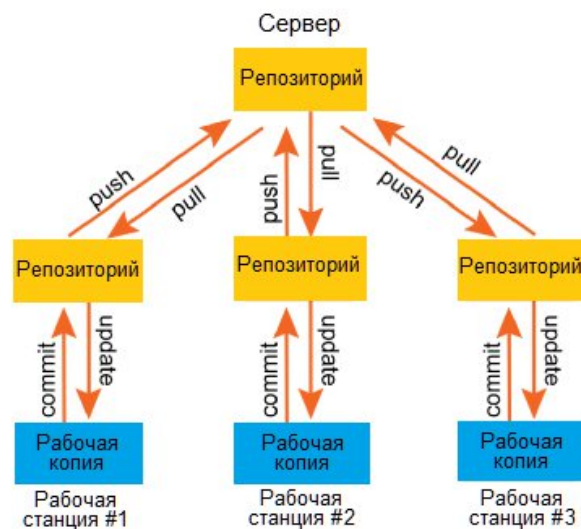
Only show cookies with an issue

Name	Value	Do...	Path	Exp...	Size	Htt...	Sec...	Sa...	Par...	Pr...
__Secure-3PSIDCC	AP8dLxAPeQ_c96xgRRFz-xldrbG...	.go...	/	202...	91	✓	✓	None		High
SSID	AQEPaFvTE446RAKN3	.go...	/	202...	21	✓	✓			High
__Secure-1PAPISID	9YMen-Alg2cJVRi3/AAjh9yi_57wV...	.go...	/	202...	51		✓			High
HSID	A8CRa3aq73nOU4sHG	.go...	/	202...	21	✓				High
__Secure-3PSID	VAhFJZnbITHaAG4KA3SR4qBciDa...	.go...	/	202...	85	✓	✓	None		High
__Secure-1PSID	VAhFJZnbITHaAG4KA3SR4qBciDa...	.go...	/	202...	85	✓	✓			High
SID	VAhFJZnbITHaAG4KA3SR4qBciDa...	.go...	/	202...	74					High
SAPISID	9YMen-Alg2cJVRi3/AAjh9yi_57wV...	.go...	/	202...	41		✓			High
APISID	Sk1qvBRlgM4hKEnH/AHySQtmCA...	.go...	/	202...	40					High
__Secure-1PSIDCC	AP8dLbyj1Ms-o1Vk8J1fQFJyW_Fh...	.go...	/	202...	92	✓	✓			High
SIDCC	AP8dLtzvtO9cUbsta_IDvyZskj-1Zt...	.go...	/	202...	80					High
__Secure-3PAPISID	9YMen-Alg2cJVRi3/AAjh9yi_57wV...	.go...	/	202...	51		✓	None		High
_ym_uid	1663152377814998219	.go...	/	202...	26		✓	None		Me...
1P_JAR	2023-04-16-07	.go...	/	202...	19		✓	None		Me...
AEC	AUEFqZfewA2jYqwOc_SEQndWW...	.go...	/	202...	61	✓	✓	Lax		Me...
NID	511=BkP8d6Y2hrOe_ECZUF2j...	.go...	/	202...	178	✓				Me...
DV	EwWIRdpyhAESQAgxTl7ANXygv...	ww...	/	202...	33					Me...
vm_d	1681219149	.oo...	/	202...	15		✓	None		Me...

Select a cookie to preview its value

Сессии

<https://developer.mozilla.org/ru/docs/Web/HTTP/Session>

GIT**Распределённая система контроля версий**

<https://practicum.yandex.ru/blog/chto-takoe-github-kak-on-rabotaet/>

<https://docs.github.com/ru/pages/getting-started-with-github-pages/about-github-pages>

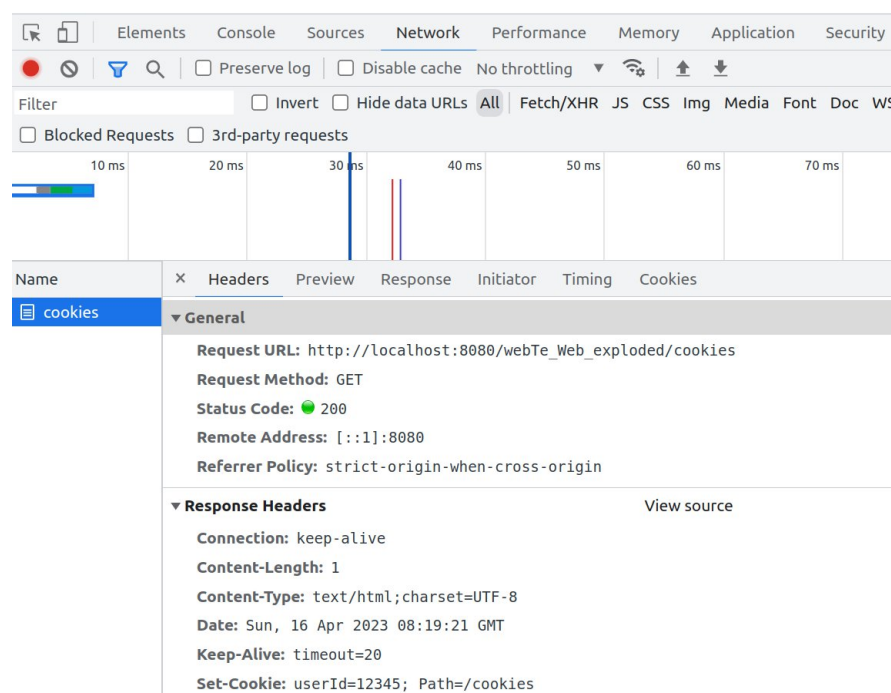
Часть 2. Практическая.

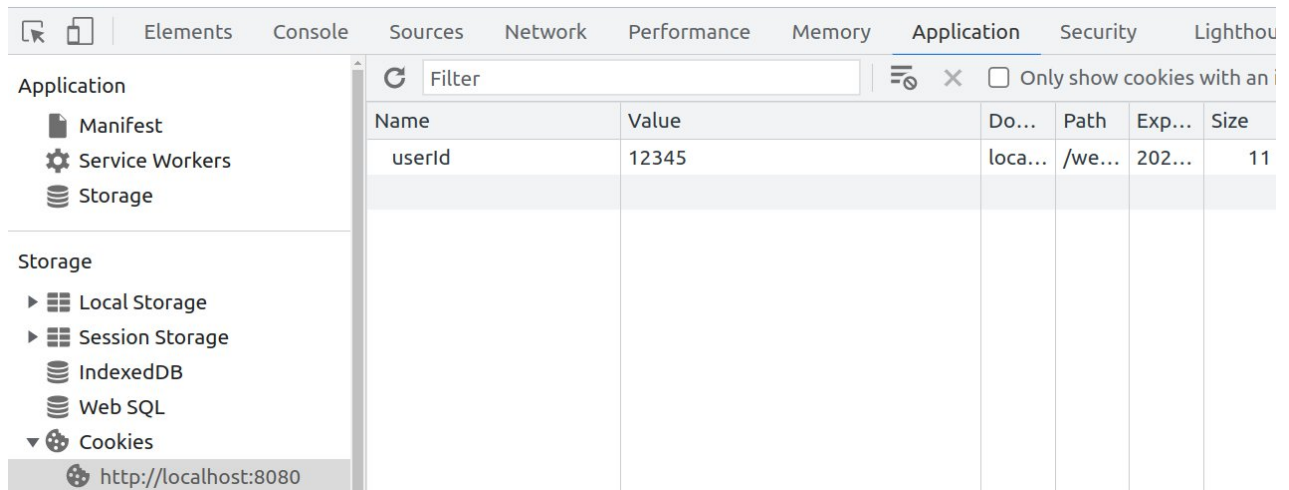
Создадим сервлет для создания cookies.

```
package WT04;

import ...
@WebServlet("/cookies")
public class CookieServlet extends HttpServlet {
    2 usages
    private static final String UNIQ_ID = "userId";
    2 usages
    private static final AtomicInteger counter = new AtomicInteger();
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        var cookies :Cookie[] = req.getCookies();
        if (cookies == null || Arrays.stream(cookies).stream<Cookie>
            .filter(cookie -> UNIQ_ID.equals(cookie.getName()))
            .findFirst() Optional<Cookie>
            .isEmpty()){
            var cookie = new Cookie(UNIQ_ID, value: "12345");
            cookie.setPath("/cookies");
            cookie.setMaxAge(-1);
            resp.addCookie(cookie);
            counter.incrementAndGet();
        }
        resp.setContentType("text/html");
        try (var writer :PrintWriter = resp.getWriter()) {
            writer.write(counter.get());
        }
    }
}
```

проверим, что сервлет работает, в браузере запускаем режим разработчика, после чего делаем запрос к серверу по указанному нами url. В request видим передаваемое значение, которое проверяем во вкладке application.





Было бы не лишним запустить код сервлета в режиме debug и смотрим какие фазы выполняются, а какие не выполняются при повторном запросе. Так же смотрим в браузер в режиме разработчика на значения request и response.


Задание:

Необходимо разработать простейший сервлет, в котором:

- < Получить список всех заголовков и вывести на экран их значения;
- < Переопределить метод doPost для отображения переданных переменных в виде: param=1234&value=test
(в Postman использовать эту вкладку)
- < Изменить сервлет работающий с передачей текстовых файлов для работы с бинарными, в частности отображения файлов изображений *.jpg
- < Разместить файл с изображением в директории на жёстком диске, в браузере отобразить его с помощью изменённого сервлета.

Часть 3. Контрольные вопросы.

По результатам выполнения лабораторной работы преподаватель надеется, что студент получил достаточные теоретические и практические знания и в состоянии ответить на следующие вопросы:

1. Что такое war-файл, его отличие от jar-файла?
2. Какова общая структура веб-проекта?
3. Что такое контейнер сервлета?
4. Каков жизненный цикл сервлета?
5. Что такое ServletContext, что он содержит?
6. Когда и сколько раз при (в какой фазе) выполнения сервлета вызывается метод `init()`?
7. Что такое параметры запроса, какой у них тип и как передать массив параметров по одному ключу?
8. В чём отличия значений передаваемых параметров в опциях Postman?

9. Какой метод в HTTP не является неизменяемым?
10. Что такое CGI отличие этой технологии от сервлетов?

Для защиты лабораторной работы необходимо:

- < Прочитать и понять теоретическую часть.
- < **Индивидуально** выполнить все задания (теоретические и практические) согласно варианта - если такой присутствует. Принимается первый уникальный код, все остальные признаются плагиатом или репликами первого и не учитываются.
- < Понимать, что делает каждая строка написанного кода.
- < Оформить отчёт, который должен быть распечатан или написан вручную на листах формата А4 (цвет листов не важен) и содержать титульный лист и ход выполнения работы. Выводы о результатах проделанной работы приветствуются.
* использование репозитория (git или аналоги) с открытым доступом приветствуется.

P.S. Для тех, кто не имеет доступа в сеть Интернет, все программы необходимые для выполнения данной лабораторной работы находятся в папке с заданием.