**РОСЖЕЛДОР**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Ростовский государственный университет путей сообщения» (ФГБОУ ВО РГУПС)**

**Отчет по лабораторной работе №2**

**«Инструменты разработчика в браузере»**

**Предмет: Веб-программирование**

Выполнил: Лола Е.А. Преподаватель: Хлопин К.В.

Группа: АИБ-3-041

Ростов-на-Дону

2023 г.

**Цель работы**: В данной работе мы познакомимся с инструментами разработчика браузеров на примере Chrome. Научитесь использовать их для диагностики некоторых неполадок веб-ресурсов.

**1. Исследование заголовков и тела обычных запросов и их ответов.**

* Request URL - адрес, куда отправляется информация из формы, состоит из HTTP метода, URL, URI ресурса.
* Request Method - указывают, какое желаемое действие выполнится для данного ресурса.

Существуют методы: GET (запрашивает ресурс, используется для извлечения данных), HEAD (запрашивает ресурс также, как метод GET, но без тела ответа), POST (используется для отправки сущностей ресурсу., PUT, DELETE, CONNECT, OPTIONS, TRACE, PATCH.

* Status Code - указывает на успешность выполнения запроса.  
  Делится на 5 групп:  
  1. Информационные 100 - 199  
  2. Успешные - 200 - 299  
  3. Перенаправления 300 - 399  
  4. Клиентские ошибки 400 - 499  
  5. Серверные ошибки 500 - 599  
  Например, 200 - "Успешно", запрос успешно обработан;  
  301 - "Перемещен на постоянной основе", URI запрашиваемого ресурса был изменен.
* Remote Address - состоит из IP-адреса ресурса и порта. Обычно для подключения по HTTP порт - 80, а для HTTPS - 443. Например, в данном примере 5.255.255.80:443
* Referrer Policy - это HTTP Response Header. Он может содержать одно из значений:
  + ""
  + "no-referrer"
  + "no-referrer-when-downgrade"
  + "same-origin"
  + "same-origin"
  + "strict-origin"
  + "origin-when-cross-origin"
  + "strict-origin-when-cross-origin"
  + "unsafe-url"
* content-type - определяет Mime-тип ресурса - строка, отправляемая вместе с файлом, которая указывает тип файла. (например, передаваемый аудиофайл может быть помечен как audio/ogg тип, а изображение - image/png)
* cache-control - общий заголовок Cache-Control используется для задания инструкций кеширования как для запросов, так и для ответов.  
  + public - указывает, что ответ может быть закэширован в любом кэше.
  + private - указывает, что ответ предназначен для одного пользователя и не должен помещаться в разделяемый кэш. Частный кэш может хранить ресурс.
  + no-cache - указывает на необходимость отправить запрос на сервер для валидации ресурса перед использованием закэшированных данных.
  + only-if-cached - указывает на необходимость использования только закешированных данных. Запрос на сервер не должен посылаться.
* Cookie - небольшой фрагмент данных, который сервер отправляет браузеру пользователя. Браузер может сохранить этот фрагмент у себя и отправлять на сервер с каждым последующим запросом. С помощью кук можно сохранить любую информацию о состоянии.
* user-agent - строка с характеристиками, по которым сервера и сетевые узлы могут определить тип приложения, операционную систему, производителя и версию пользовательского агента. Пользовательский агент - программа, представляющая пользователя и выполняющая действия от его лица. Например, веб-браузер.
* Referer - заголовок Referer позволяет понять, откуда пришли пользователи. Он содержит адрес, с которого пришел запрос.

Remote Address состоит из IP-адреса сервера и порта. Порт подключения - число, указываемое при запросе на соединение. Для установки соединения протоколу необходимы:

* Номер протокола
* IP-адреса отправителя и получателя
* Номера портов процесса-отправителя и получателя

Заголовки разделены на General, Response и Request.

Request Header используется в HTTP-запросе и не используется относится к содержимому сообщения. Заголовки запроса, такие как Accept, Accept-\* или If-\* позволяют выполнять условные запросы; другие, такие как Cookie, User-Agent или Referer уточняют контекст, чтобы сервер мог адаптировать ответ.

Response Header используются в ответе и не относятся к содержимому сообщения. Заголовки ответа, такие как Age, Location, Server используются чтобы дать более детальный контекст ответа.

Повторяются заголовки content-length, content-type. Запрос и ответ могут отличаться по типу и длине.

Тело запроса — это информация, которую передал браузер при запросе страницы. Но тело запроса присутствует только если браузер запросил страницу методом POST. Например, если отправлена форма, то телом запроса будет содержание формы.

**2. Исследование указывающих ответов сервера.**

При попытке связаться с сайтом [http://rgups.ru](http://rgups.ru/) заголовок Status Code вернул 301. Произошло перенаправление на сайт с протоколом https, для этого потребовался один запрос. Адрес изменился на <https://rgups.ru/>.

**3. Исследование получения и передачи cookie.**

Все куки, используемые на странице, можно посмотреть во вкладке Application -> Storage -> Cookies

Среди атрибутов есть:

* Name - название файла Cookie
* Value - значение файла
* Domain - уточняет, какие хосты могут получать Cookie. Если сервер не уточняет, то браузер использует тот же домен, что и у хоста.
* Path - устанавливает URL, который должен быть в запрашиваемом URL, чтобы отправить Cookie заголовок
* Expires - определяет срок жизни файла Cookie

Cookie предназначены для сохранения пользовательской информации на сайте для дальнейшего удобства пользования. В нем сохраняются некоторые введенные данные, установленные настройки, информация о посещении сайта.

**4. Исследование построения документов и сопутствующих запросов.**

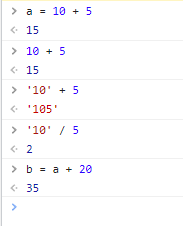
DOM - структурированное представление документа, в виде группы узлов и объектов, которые имеют свойства и методы. Теги являются узлами и формируют структуру документа. Благодаря нему можно связать веб-страницу с языками программирования.

Итоговый документ может отличаться от тела ответа, полученного от сервера, так как DOM может редактироваться языками программирования во время работы.

Браузер запрашивает различные медиа-данные для документа, например изображения, исполняет скрипты, например загрузку данных из кэша памяти.

**5. Исполнение кода в консоли. Использование консоли для отладки.**

Операции в Console:

[](https://github.com/artemtitov1/web_labs/blob/main/2_lab/img/js-console.png)

**Вывод:** в данной работе я познакомилась с инструментами разработчика браузеров на примере Chrome, научилась использовать их для диагностики некоторых неполадок веб-ресурсов.