# DevTools браузера: общее описание, вкладки, доп. возможности.

**Как открыть DEV Tools?**

На странице, открытой в браузере нажать сочетания клавиш:

Shift+Ctrl+J (I) (Win, Linux)

Cmd + Option + K (Mac)

Для Windows работает кнопка F12.

**Для чего в тестировании может использоваться данный инструмент?**

* Просмотр информации об элементах при тестировании на соответствие макетам
* Мониторинг http-запросов
* Эмуляция экранов мобильных устройств
* Моделирование настроек сети
* Работа с cookie
* Скриншоты
* Тестирование локализации

**Рассмотрим более подробно возможности инструмента:**

1. Через dev tools доступно тестирование веб версии на разрешении мобильных девайсов. Инструмент имеет перечень эмуляторов девайсов с соответствующими разрешениями. В таком случае клик мышкой воспроизводится как тап.
2. **Console (= terminal)**

Через данную вкладку доступен мониторинг ошибок клиента и логи. Также через данную вкладку допустимо использование кода JS при необходимости, например, чтобы воспроизводить сценарии тестирования, которые недоступны с помощью devtools или нет возможности их воспроизвести через UI.

Для данной вкладки предусмотрен фильтр логов и ошибок через Console sidebar (иконка раскрытия/скрытия рядом с Clear console), есть поисковая строка, которую также можно использовать для поиска соответствующей информации.

1. **Application (TBD)**

(необходимо знать) Storage – хранилище. Service workers – работает с нотификациями, при разрешении работы с уведомлениями сайта создается Service worker для этого сайта, который слушает, есть ли в работе нотификации для отображения.

Cache storage – закэшированные данные

IndexedDB – браузерная БД? (уточнить дополнительно для чего используется и как)

Clear site data – очищает куки и кэш

(необходимо знать) Local storage – хранит пару ключ-значение (изучить дополнительно)

(необходимо знать) Cookies, (изучить по другой лекции)

1. **Elements**

Отображает элементы html страницы (язык разметки HyperText Markup Language) и применимые к ней Styles (css стили).

На данной вкладке можно искать необходимый элемент через строку поиска, поисковая строка вызывается CTRL+F.

1. **Network**

Основные используемые разделы данной вкладки:

* Headers: содержит в себе данные по запросу.
* Payload – request body, то что передается в запросе (опционально)
* Preview – аналог раздела response, данные в ней имеют более читабельный вид.
* Response – данные в формате json, есть кнопка Pretty print для улучшения отображаемой структуры файла.
* Preserve log (чек-бокс) – сохранение отображения запросов во вкладке Network при перезагрузке страницы. Если чек-бокс активен, то запросы сохраняются для отображения, при неактивном чек-боксе новая загрузка страницы идет с очищением запросов вкладки network.
* Disable cash (чек-бокс) – настройка использования данных из кэша браузера. При активном чек-боксе при перезагрузке страницы будут тянутся свежие файлы, а не из кэша браузера
* Throttling - управление сетью. Используется для воспроизведения сценариев с медленным интернетом, отсутствием интернета и т .д. Есть предзаполненные варианты настроек сети (3G, отсутствие сети и т.д.), но можно задать свои custom настройки, которые необходимы для воспроизведения определенных сценариев.

More network conditions – содержит больше настроек:

* User agent – подменяет в хэдере клиента, можно поменять браузер или отправить в хэдере информацию о мобильном клиенте.
* 3rd-party request – активирует отображение запросов в какие-то другие системы, например, запросы для аналитики.

В данной вкладке также предусмотрены фильтры и строка для фильтра.

*Для общего развития:*

*XMLHttpRequest (XMLHTTP, XHR) —*[*API*](https://ru.wikipedia.org/wiki/API)*, доступный в*[*скриптовых языках*](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA)[*браузеров*](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80)*, таких как [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript). Использует запросы*[*HTTP*](https://ru.wikipedia.org/wiki/Hypertext_Transfer_Protocol)*или*[*HTTPS*](https://ru.wikipedia.org/wiki/HTTPS)*напрямую к*[*веб-серверу*](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80)*(информация может передаваться в любом*[*текстовом формате*](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82)*, например, в*[*XML*](https://ru.wikipedia.org/wiki/XML)*,*[*HTML*](https://ru.wikipedia.org/wiki/HTML)*или*[*JSON*](https://ru.wikipedia.org/wiki/JSON)*) и загружает данные ответа сервера напрямую в вызывающий скрипт. Позволяет осуществлять HTTP-запросы к серверу без перезагрузки страницы.*

Через строку фильтра можно фильтровать отображаемые запросы как на их отображение по какому-то признаку, например, методу или домену, так и на исключение отображения таких запросов, используя символ «-» (минус) перед фразой в фильтре.

1. **Recording**

**Recording** записывает действия на сайте (попробовать самостоятельно)

**Дополнительная информация:**

Уточнения по запросам: сразу грузится страница HTML, без стилей. HTML содержит в себе ссылку на стили, которые используются для страницы.

Через мониторинг запросов можно обнаружить такие проблемы, как чрезмерный вес картинки, лишние запросы, что может сильно влиять на производительность системы.

Современные браузеры обычно поддерживают шесть одновременных соединений и параллельных загрузок на один домен.

**Копирование запросов:**

Вкладка network – выбрать запрос – правой кнопкой мыши вызвать контекстное меню – Выбрать Copy. Варианты, использование которых очень пригодилось:

* Copy as fetch
* Copy as cURL (используется для терминала)
* Copy all as HAR

**Импорт/экспорт HAR:**

Дополнительные кнопки вкладки network.

Экспорт: записывает в лог браузера все запросы и ответы, сохраняет в файл har, который скачивается на локальную машину.

Импорт: сохраненные логи браузера в файлк har, можно импортировать в браузер и получить результат на экране, соответствующий данным логам.

Решает проблему с воспроизведением дефектов у разработчика, т.к. помимо перечня шагов есть полный лог браузера для данной ситуации, и можно локализовать проблему намного быстрее.

**Скриншот всей страницы, а не только видимой зоны:**

В том случае, если нужно сделать скриншот всей страницы, не только отображаемой области страницы, в devtools есть такая возможность:

Вкладка Console – filter – в строке фильтра активируем курсор - нажимаем сочетание клавиш ctrl+shift+P – в предложенном списке функций выбрать full size screenshot (можно начать вводить и отобразится значение в предлагаемом списке)-выбрать его.

Скачивается png файл всей страницы, а не только видимой зоны.

**Подключение андроида к chrome dev tools**: TBD

Девайс подключается к компьютеру через USB, все действия на веб версии на девайсе будут доступны в dev tools компьютера.

Материалы

https://www.youtube.com/watch?v=4iIryVM77HM