



Chrome DevTools vs Firefox DevTools

Devtools - это набор функций, доступных в определенных браузерах, которые позволяют выполнять ряд действий по отладке своих веб-страниц и веб-приложений без необходимости возврата к среде разработки. В наши дни некоторая степень функциональности devtool является встроенной функцией в большинстве популярных браузеров, и это, конечно, включает Chrome от Google и Firefox от Mozilla.

Сравнение Devtools

- Одним из первых и самых простых факторов является производительность браузера. Когда-то Chrome славился своей молниеносной загрузкой. К сожалению, как это часто бывает с широко распространенными программами, с тех пор он стал немного раздутым и медленным. В отличие от этого, Firefox теперь может похвастаться быстрым временем загрузки и небольшим использованием памяти.
- Необходимо учитывать и вопросы конфиденциальности, поскольку Chrome неразрывно связан с Google, в то время как Firefox не связан ни с какими крупными корпоративными организациями, и конфиденциальность пользователей занимает центральное место в его основных принципах.

- Что касается самих инструментов разработчика, оба браузера обладают такими же возможностями, как и другие, но есть некоторые различия, которые могут повлиять на ваше мнение. Например, Firefox Developers Edition поставляется со всем, что можно использовать “из коробки”. Chrome, с другой стороны, включает в себя множество стандартных функций, но полагается на дополнительные плагины, чтобы привести набор функций в соответствие с Firefox.

Также у Firefox DevTools есть функции, которые не предлагаются Chrome DevTools:

- Средство просмотра свойств DOM Проверьте свойства страницы DOM, функции и т. Д. (Вы заметите, что свойства объекта обычно не отображаются в консоли Chrome, вы никогда бы не узнали, что `body.innerHTML` существует, посмотрев на «`document.body`» в консоль.)
- Панель инструментов разработчика Интерфейс командной строки для инструментов разработчика.
- Редактор шейдеров Просматривайте и редактируйте шейдеры вершин и фрагментов, используемые WebGL.
- Редактор веб-аудио. Изучите граф аудиоузлов в аудиоконтексте и измените их параметры.
- Создание снимков экрана Сделайте снимок экрана всей страницы или отдельного элемента.

Некоторые дополнительные инструменты в панели настроек:

1. Измерьте часть страницы
 2. Блокнот
- Сетевая панель показывает трассировку стека причины каждого сетевого запроса, в Chrome вам нужно пройти консольный журнал сетевых запросов, найти тот и развернуть его. Кроме того, вы можете увидеть фактический предварительный просмотр (на вкладке предварительного просмотра) ответа `xhr`, если он возвращается в формате `html`.
 - И, пожалуй, самое полезное, инспектор показывает (ev) рядом с каждым элементом с прослушивателем событий на нем, с прямым перечислением всего, что добавляет к нему прослушиватель событий. (лучше, чем список вкладок событий Chrome справа.)

В свою очередь Chrome DevTools (начиная с Chrome 59.0) имеет больше возможностей в отношении стандартных инструментов, и они предоставляют другие инструменты ниши.

Эти функции включают в себя:

- точки останова DOM и XHR
- Панель слушателей событий.
- Панель свойств
- Инспектор по безопасности
- Ревизия
- Моделирование прикосновений и плотность пикселей в эмуляции устройства.
- Живое редактирование на панели “Источники”.
- Workspaces
- Блокировка сетевого запроса
- Инструмент расширенной памяти
- Манифест, диспетчер службы и инспектор кэша приложений
- Вызов производительности прокрутки Highlighter
- Кадры в секунду.
- Эмуляция CSS Media.
- Глобальный поиск файлов
- Анализатор покрытия кода
- Геолокация, ориентация и сенсорная эмуляция.

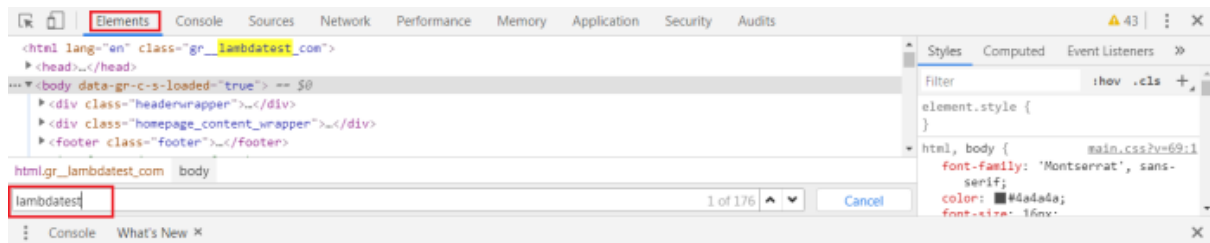
Полезные возможности DevTools для QA-специалистов

1. Поиск элемента

Эта функция помогает тестировщику (или разработчику) искать в DOM-дереве нужные элементы страницы.

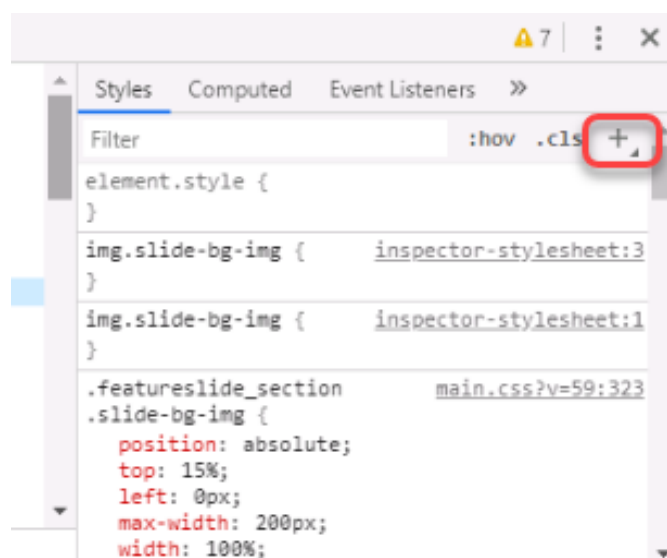
Откройте Chrome Dev Tools и зайдите на вкладку **Elements**. Собственно, ее не нужно искать: по умолчанию она открывается первой. Находясь в этом разделе, нажмите **Ctrl + F (Cmd + F)**.

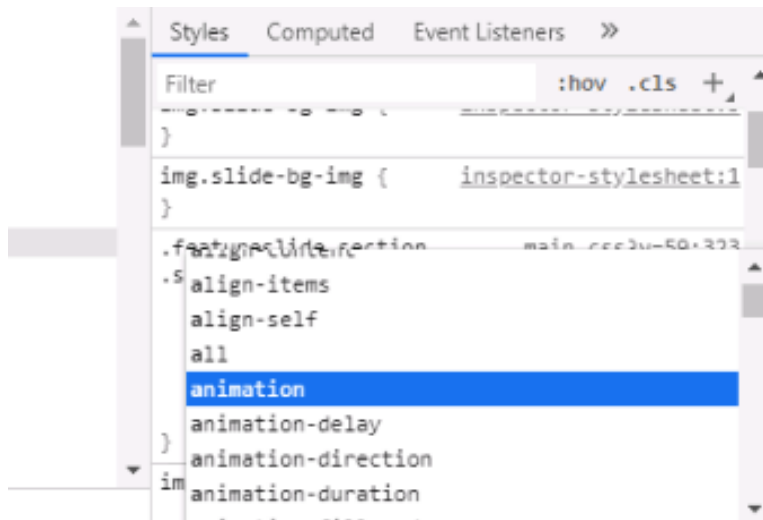
Откроется поле для ввода, где вы сможете вставить искомое слово. После ввода инструмент подсветит этот элемент в коде.



2. Добавление и редактирование элементов

Инструменты разработчика позволяют добавлять теги в код или изменить их.





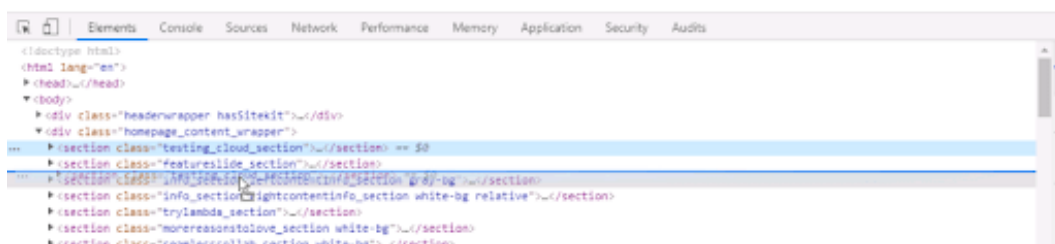
Таким образом можно быстро вносить изменения в стили сайта. Эти изменения будут применены только в вашем браузере и только в текущей сессии. Как только вы перезагрузите страницу, все восстановится в изначальном виде.

3. Изменение расположения DOM-элементов в дереве

Вы можете изменить позиции элементов во вкладке **Elements**, просто перетаскивая их.

Таким образом можно изменить структуру страницы и прикинуть, где ваш контент будет лучше смотреться.

Cross Browser Testing Cloud



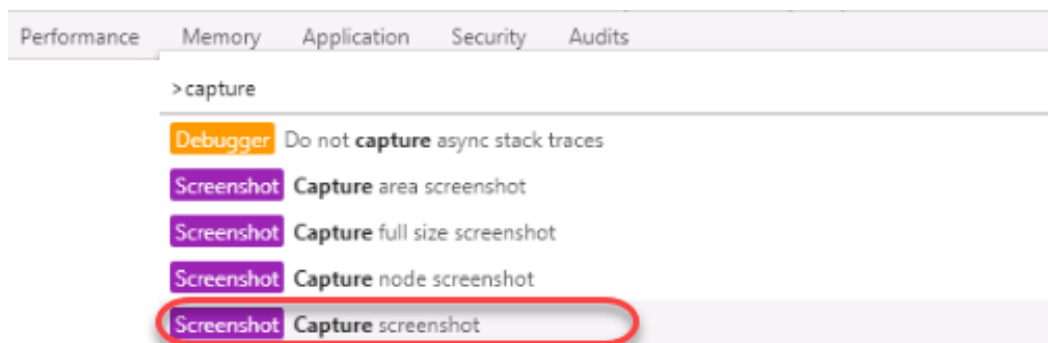
4. Поиск CSS-свойства в Sources

Тестировщику часто приходится искать в Sources свойство, определенное в Elements, для внесения каких-то изменений. Это можно быстро сделать при помощи **Ctrl + Click (Cmd + Click)**. Затем перейдите во вкладку Sources, где нужное свойство будет подсвечено.

5. Скриншот элемента

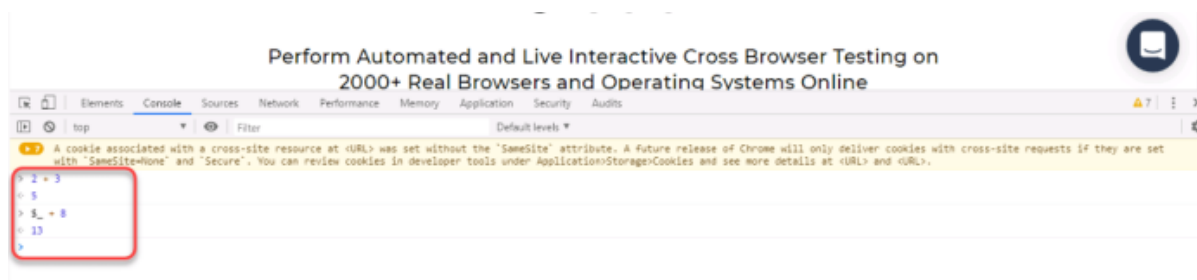
Тестировщики могут делать скриншоты элементов. Для этого выберите элемент и нажмите **Ctrl + Shift + P (Cmd + Shift + P)**.

Откроется палитра команд. Выберите в ней **Capture Node Screenshot**, чтобы сделать скриншот элемента.



6. Использование результатов предыдущей операции в консоли

Работая в консоли, вы можете заметить, что каждая новая строка выдает определенный output, зависящий от выражения, определенного в предыдущей. Инструменты разработчика позволяют использовать этот output в следующих выражениях, обращаясь к нему при помощи «\$_n».

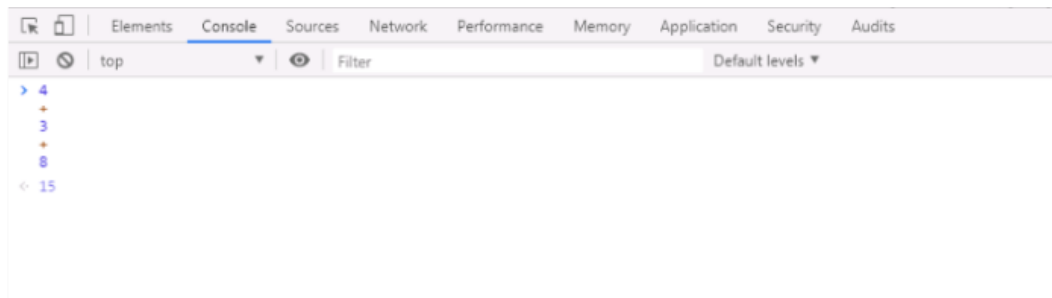


7. Многострочные скрипты

Обычно в любой программе или операционной системе консоль на каждой новой строке выдает результат предыдущего выражения.

Консоль разработчика работает аналогично.

Но в Console в Chrome Dev Tools тестировщик может применять многострочные скрипты. Для этого в конце каждой строки нажимайте не Enter, а **Shift + Enter**.



Вкладка Console быстро заполняется различными логами. Чтобы очистить консоль, нажмите **Ctrl + L (Cmd + K)** или напишите команду `clear();`

8. Локаторы для автоматизированного тестирования при помощи Selenium

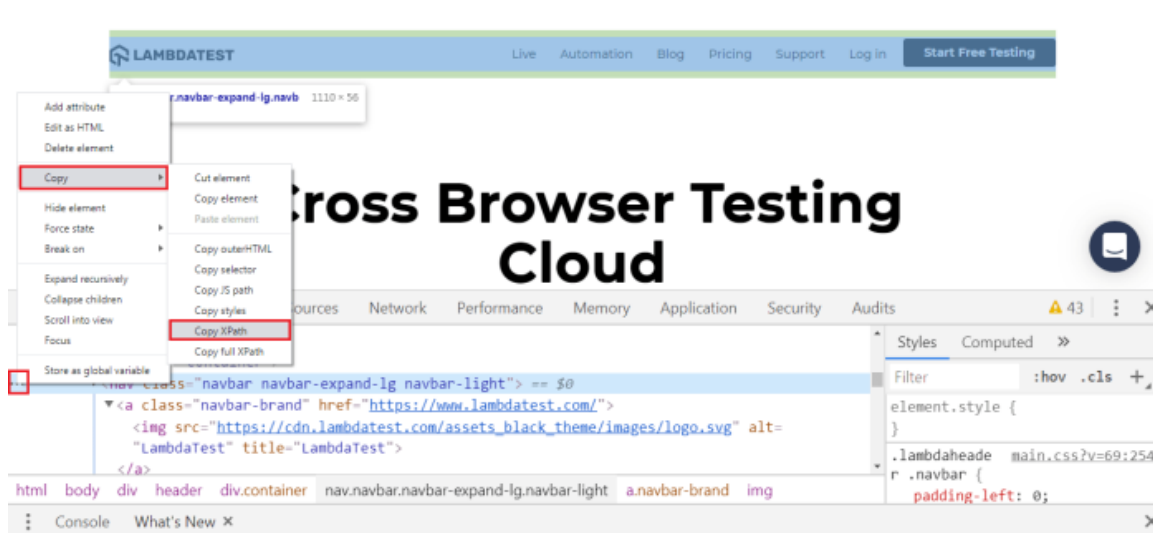
Для автоматизированного тестирования с использованием Selenium очень часто используются локаторы. Они нужны для поиска конкретных элементов на странице.

В Selenium есть следующие локаторы:

1. ID
2. Name
3. Linktext
4. Partial Linktext
5. Tag Name
6. Class Name
7. CSS Selector
8. Xpath

При помощи инструментов разработчика можно быстро находить локаторы для любых элементов страницы.

Например, если вам нужен Xpath элемента, вы можете скопировать его при помощи инструментов разработчика, и «скормить» своим скриптам в Selenium.

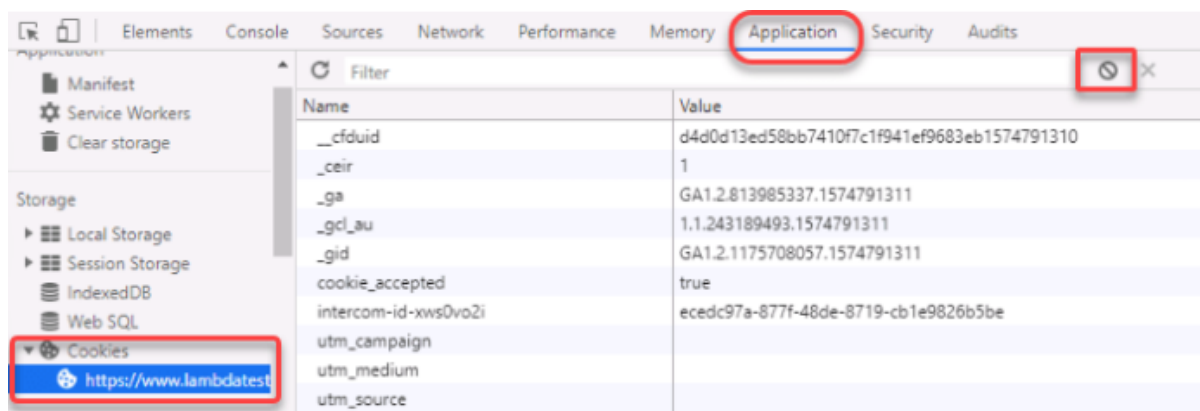


9. Очистка cookies

Тестирование любого стороннего приложения создает множество cookies при отладке. Поскольку отладка требует выполнения скриптов множество раз, cookies могут влиять на результаты. Поэтому их необходимо чистить.

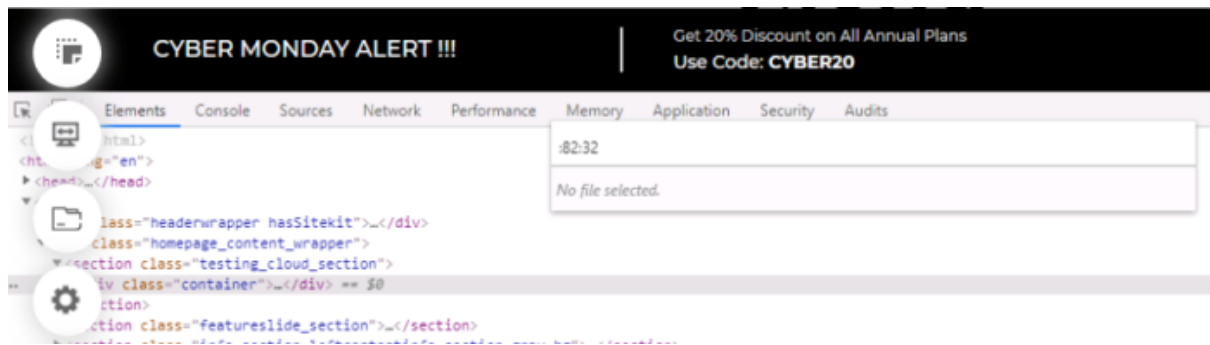
В Google Chrome для этого есть специальная опция в настройках. Но тестировщик может сэкономить время и быстро очистить все cookies на конкретном сайте или странице.

Для этого нужно перейти во вкладку **Applications** и выбрать **Cookies** в меню. Это даст вам возможность удалить cookies выборочно или все сразу.



10. Переход к определенной строке

В инструментах разработчика можно перейти к указанной строке, введя ее номер в строке поиска. Для открытия поиска нажмите **Ctrl + O (Cmd + O)**.

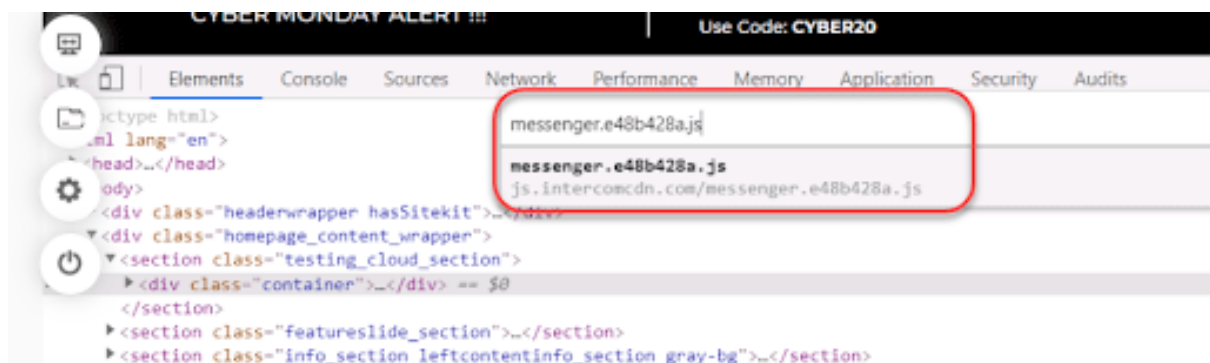


11. Переключение между файлами в рамках проекта

Веб-проекты, как правило, содержат много файлов. Тестировщику может понадобиться неоднократно переключаться между ними в ходе интеграционного тестирования или на стадии юнит-тестов.

Инструменты разработчика позволяют быстро переключаться на нужный файл (в рамках проекта), просто введя его название в строке поиска.

Поиск по файлам открывается клавишами **Ctrl + P (Cmd + P)**.

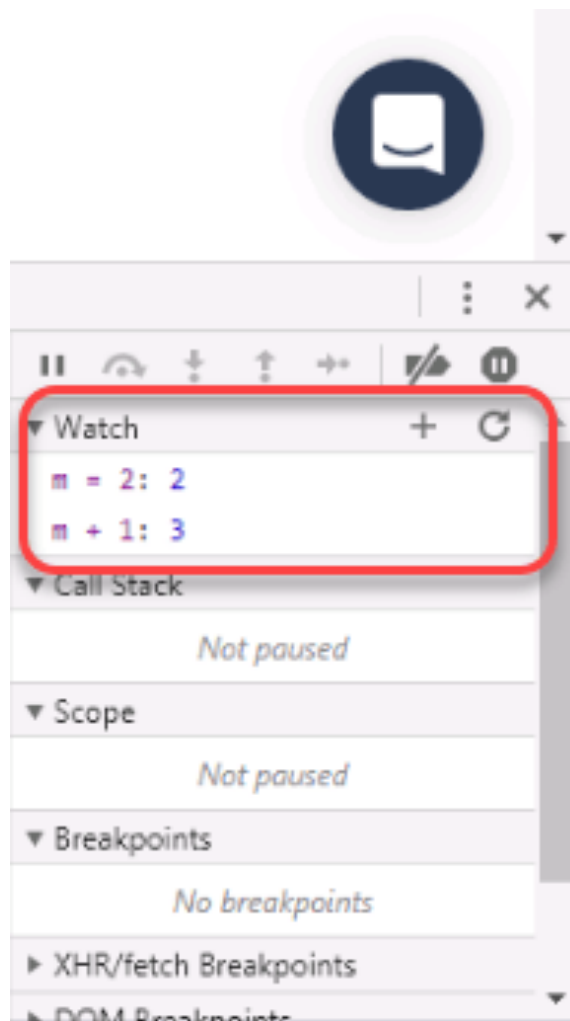


12. Watch Expression

В инструментах разработчика Chrome есть функция под названием Watch Expression. Найти ее можно во вкладке **Sources**, в колонке справа.

Тестировщику во время дебага нужно постоянно следить за переменными и выражениями. Искать их снова и снова очень затратно по времени. Watch Expression может делать это за вас.

Тестировщик просто вводит выражение или переменную в этот раздел, и они будут обновляться при каждом изменении.



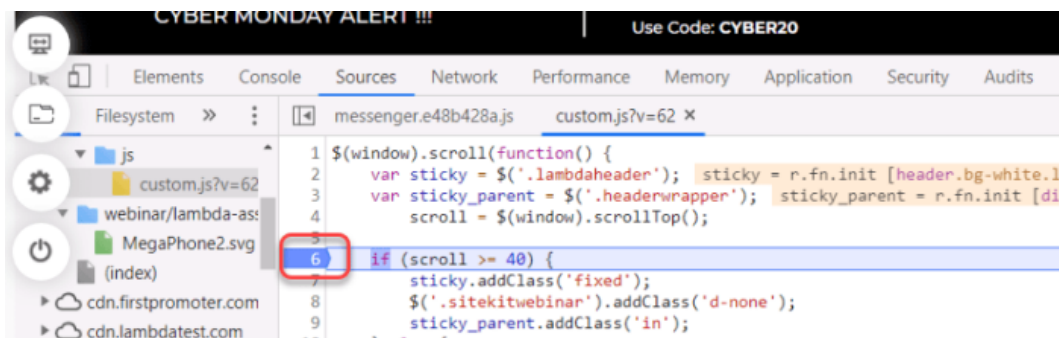
13. JavaScript Breakpoints

Современные сайты содержат много JavaScript-кода. При одном-единственном запросе может меняться множество значений, а тестировщику нужно определять, где они находятся, и наблюдать за ними.

Например, API-запрос, использующий AJAX, может запускаться 4 раза и возвращать разные коды статуса. Для эффективного тестирования его нужно проверять на каждом этапе. Для этого в инструментах разработчика есть специальная фича — JavaScript Breakpoints.

Используя JavaScript Breakpoints, вы можете определять строку, на которой хотите приостановить выполнение кода.

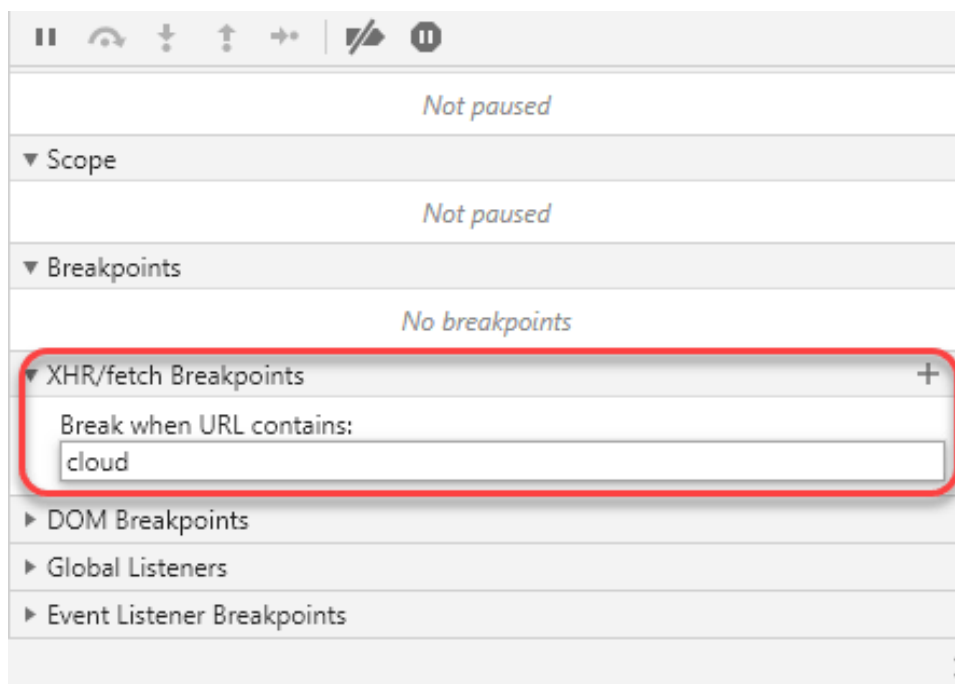
Чтобы воспользоваться этой функцией, просто кликните на номер строки, на которой выполнение JavaScript-кода должно прерваться.



14. XHR/Fetch Breakpoints

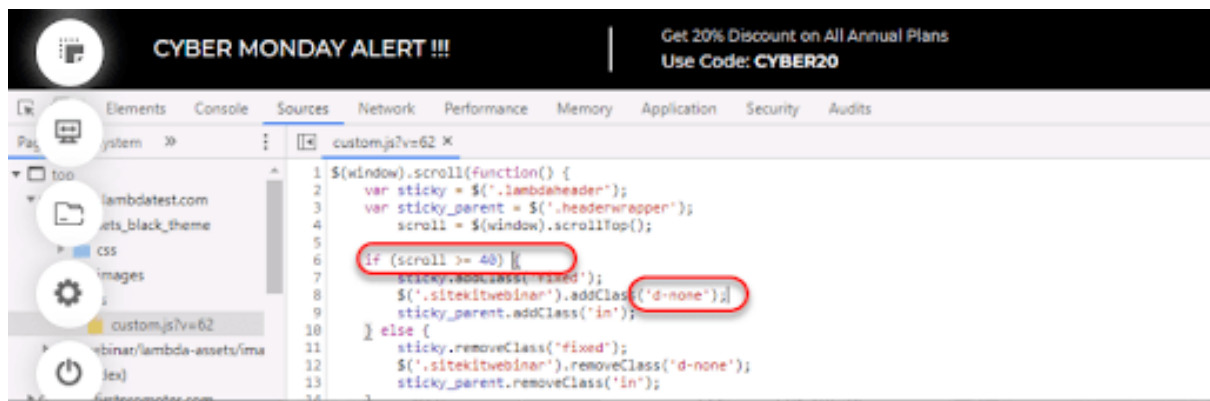
Во вкладке Sources под Watch Expression есть раздел XHR/Fetch Breakpoint. С его помощью вы можете устанавливать брейкпоинт в скрипте при каждой отправке запроса.

Тестировщик также может использовать эту функцию только для определенных запросов, с конкретными URL.



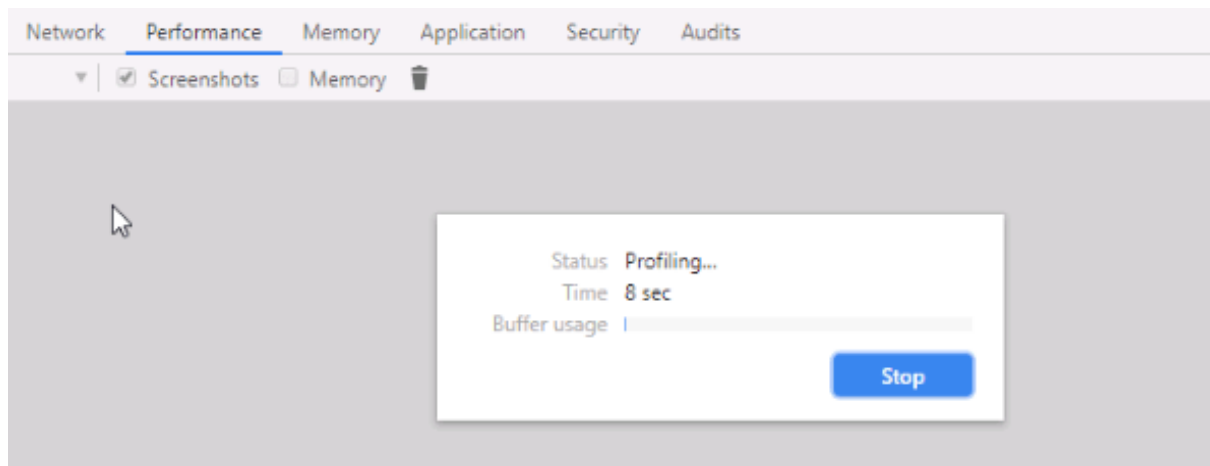
15. Включение нескольких курсоров

Разве не круто было бы иметь возможность писать на разных строках одновременно? Инструменты разработчика позволяют это делать. Для этого нужно инициализировать множественные курсоры, зажав **Ctrl (Cmd)** и кликая в нужных местах.



16. Мониторинг производительности

Dev Tools позволяют тестировщику следить за производительностью сайта прямо в панели инструментов. Для этого нужно зайти во вкладку Performance, нажать **Ctrl + E (Cmd + E)**, а затем — **Ctrl + R (Cmd + R)** для перезагрузки страницы. Чтобы прекратить замеры производительности, нажмите кнопку «Stop».



Инструменты разработчика сильно упрощают и ускоряют работу. Обязательно используйте их во время тестирования!

Untitled

Тестирование web-приложений с помощью DevTools и Firefox dev