

УДК 004.4'234

КАК УСТРОЕН HEADLESS CHROME

М.А. БАЛАБАШ

(Представлено: канд. физ.-мат. наук, доц. А.Ф. ОСЬКИН)

В статье дается определение headless веб-браузерам и способы их применения. Рассматриваются основные компоненты Headless Chrome веб-браузера и протокол взаимодействия.

Введение. Headless браузер это незаменимый инструмент front-end разработчика, с помощью которого можно тестировать код, проверять соответствие верстке и множество других возможных применений.

Headless браузеры обеспечивают автоматизированное управление веб-страницей в среде, аналогичной популярным веб-браузерам. Они предоставляют специальные программные интерфейсы для взаимодействия со средой выполнения и позволяют управлять веб-браузером, исполнять код в его среде и получать любую доступную информацию из контекста веб-браузера.

Chromium — это открытая реализация браузера на основе WebKit. Google Chrome — это брендированная версия Chromium, у которой есть проприетарные кодеки, интеграции с сервисами Google и др. Headless Chrome - это headless версия браузера Google Chrome. Именно её мы и будем исследовать.

В качестве интерфейса для управления (API) используется Chrome DevTools protocol. Chrome DevTools - это набор инструментов для веб-разработчиков, встроенный непосредственно в браузер Google Chrome.

Компоненты Chromium. Самый низкий уровень Platform layer.

- Ozone — абстрактный менеджер окон в Chrome — то, с чем взаимодействует оконный менеджер операционной системы. На Linux это либо X-server, либо Wayland. На Windows это менеджер окон Windows;

- Scheduler — тот самый планировщик, который занимается синхронизацией потоков и процессов, т.к. Chrome — это многопроцессное приложение;

- Net — компонент для работы с сетью, который парсинг HTTP, создает заголовки и т.д.

Уровень Content layer — самый большой компонент, который есть в Chrome.

- Blink — веб-движок на основе WebCore из WebKit для работы с HTML и CSS; V8 (JavaScript движок); API для всех расширений, которые мы используем в Chrome. Туда же входит DevTools protocol;

- Content API — это интерфейс, при помощи которого можно очень легко использовать все возможности веб-движка. Поскольку внутри Blink много всего (несколько сотен тысяч интерфейсов), то, чтобы не потеряться во всех этих методах и функциях, нужен Content API.

Уровень безголового браузера (Headless layer).

- Headless library;

- Embedder API интерфейс для встраивания Headless library в приложение;

- Client API — интерфейс, который использует Puppeteer.

Уровень приложения (Application layer).

- Ваше приложение (Embedding app);

- Мини-приложения, например, Headless shell.

Chrome DevTools protocol

Все front-end разработчики (и не только) сталкивались с Chrome DevTools protocol, поскольку пользовались панелью разработчиков в Chrome или удаленным отладчиком — теми же самыми средствами разработки. Если вы запускаете средства разработчика в удаленном режиме, общение с браузером происходит при помощи именно DevTools protocol. Когда вы ставите debugger, смотрите code coverage, используете геолокацию или еще что-нибудь — все это управляется при помощи DevTools.

В протоколе есть 2 компонента:

- DevTools target — вкладка, которую вы инспектируете;

- DevTools client — например, это панель разработчика, которая запущена в удаленном режиме.

Они общаются при помощи простого JSON:

- Есть идентификатор команды, название метода, который нужно выполнить, и некоторые параметры;

- Ответ тоже выглядит очень просто: идентификатор, который нужен потому, что все команды, которые выполняются при помощи протокола, асинхронные. Для того, чтобы мы всегда могли сопоставить, какой ответ на какую команду нам пришел, нам и требуется идентификатор;

- Результат.

Puppeteer. Это библиотека (доступная для нескольких языков программирования), которая предоставляет API высокого уровня для управления Chrome или Chromium по протоколу DevTools. На текущий момент это лучшее решение для работы с Headless Chrome браузером. Она активно развивается и поддерживается командой из Google.

Большинство вещей, которые вы можете сделать вручную в браузере, можно сделать с помощью Puppeteer.

Несколько примеров использования:

- Генерация скриншотов и PDF-файлов страниц;
- Сканирование SPA (одностраничное приложение) и генерация предварительно визуализированного контента (рендеринг на стороне сервера);
- Автоматизация отправки форм, тестирование пользовательского интерфейса, ввод с клавиатуры и т.д.;
- Замерять и диагностировать показатели производительности;
- Перехват сетевых запросов / ответов;
- Тестирование расширений Chrome.

При первой установке Puppeteer скачивает версию Chromium и устанавливает ее именно под вашу архитектуру и ОС. Вам не нужно ничего скачивать дополнительно, он делает это автоматически. Вы также можете использовать и вашу любимую версию Chrome, которая установлена у вас в системе.

Заключение. Headless Chrome — инструмент, позволяющий тестировать наши веб-приложения и автоматизировать взаимодействия с ними. DevTools protocol имеет богатый набор возможностей, который позволяет разрабатывать полноценные и самостоятельные программные решения для широкого спектра задач (мониторинг веб-приложений, среда разработки VS Code). Возможность получать информацию, исполнять код и автоматизировать действия браузера — открывают много возможностей.

ЛИТЕРАТУРА

1. GitHub [Электронный ресурс] Headless Chrome Node.js API. Режим доступа: <https://github.com/GoogleChrome/puppeteer>. Дата доступа: 27.09.19.
2. ChromeDevtools [Электронный ресурс] Chrome DevTools Protocol Viewer. Режим доступа: <https://chromedevtools.github.io/devtools-protocol/>. Дата доступа: 27.09.19.
3. Chromium [Электронный ресурс] The Chromium Projects. Режим доступа: <https://www.chromium.org/>. Дата доступа: 27.09.19.