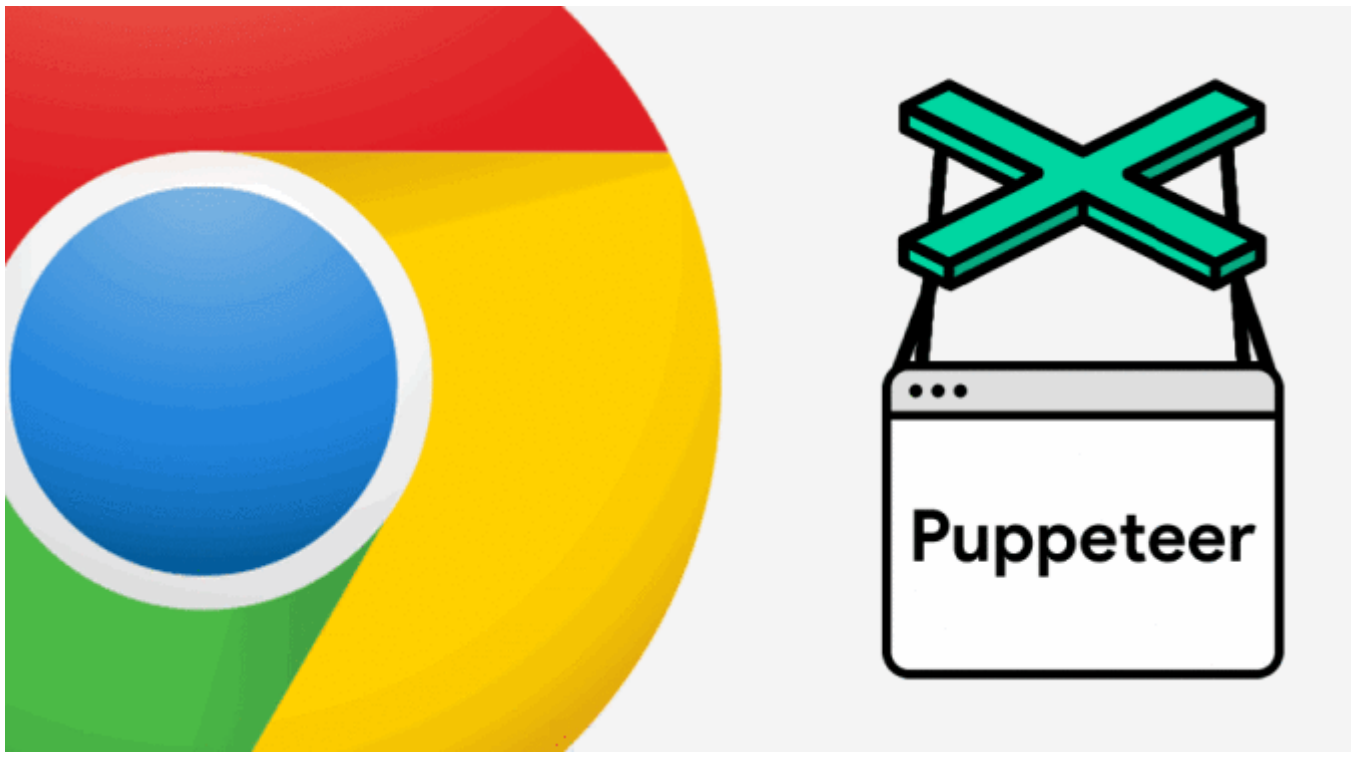


CHUYỆN CODING, **CHUYỆN LINH TINH**

Làm trò với Puppeteer – Phần 1: Cùng tìm hiểu về Puppeteer và Headless Browser

12/12/2017 | PHẠM HUY HOÀNG | 18 COMMENTS

Gần đây do đi làm phải code sml nên mình cũng hơi lười viết bài chuyên sâu về technical. Tuy vậy, mình cảm thấy lâu rồi không viết tutorial kĩ thuật nên hôm nay viết lại cho khỏi lụt nghề nhé.

Kì này, chúng ta sẽ cùng làm trò với thư viện Puppeteer của NodeJS, một thư viện cho phép chạy Chrome dưới chế độ headless browser.

Bài viết gồm 3 phần

- **Phần 1**: Làm quen với puppeteer (https://toidicodedao.com/2017/12/12/puppeteer-headless-chrome-api-phan-1/)
- **Phần 2:** Dùng puppeteer để cào dữ liệu và vếu từ mương14. (https://toidicodedao.com/2017/12/19/puppeteer-headless-chrome-api-phan-2-cao-du-lieu-kenh14/)
- **Phụ lục**: Tổng quan về testing và automation test (https://toidicodedao.com/2017/12/26/tong-quan-testing-1-lap-trinh-vien-biet-ve-testing/)
- **Phần 3**: Viết Automation test với Puppeteer (https://toidicodedao.com/2018/01/30/lam-tro-voi-puppeteer-phan-3-bat-dau-testing-voi-puppeteer/)

Trước khi bắt đầu vào code thì chúng ta tìm hiểu sơ chút lý thuyết trước nha.

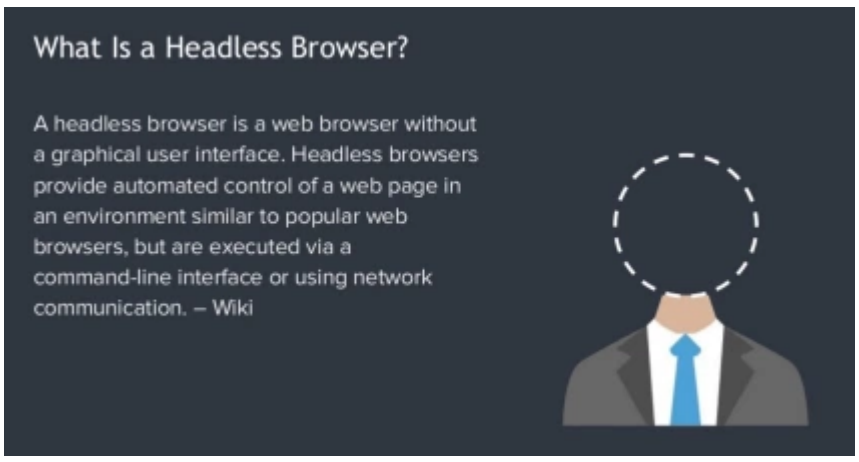
Headless browser là gì? Làm được trò gì?

Muốn biết Puppeteer làm được trò gì, ta phải hiểu về khái niệm Headless browser – dịch cho vui là **browser không đầu**, tức là browser chạy mà **không cần giao diện**.

Ủa chạy browser không cần giao diện để làm chi vậy? Đôi khi chúng ta sẽ cần mở browser lên không phải để … duyệt web, mà là cào dữ liệu, để test, chụp screenshot, đo performance.

Ta muốn làm những chuyện này **trên các server Linux, docker v…v** không có giao diện. Lúc này, headless browser là lựa chọn duy nhất.

Các bạn có thể tìm hiểu thêm ở đây: https://en.wikipedia.org/wiki/Headless_browser (https://en.wikipedia.org/wiki/Headless_browser)

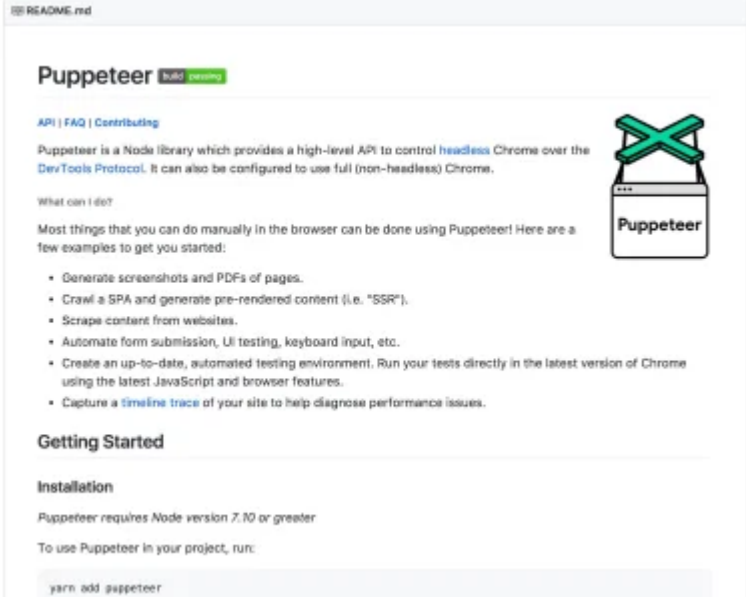


(https://toidicodedao.files.wordpress.com/2017/11/screen-shot-2017-11-26-at-11-47-30-am.jpg)

Puppeteer là cái chi chi?

Puppeteer là một thư viện của NodeJS, có khả năng điều khiển Chrome headless browser thông qua code. Các bạn có thể tìm hiểu thêm tại: https://github.com/GoogleChrome/puppeteer (https://github.com/GoogleChrome/puppeteer)

Do vậy, Chrome làm được gì thì Puppeteer làm được cái đấy. Ta có thể dùng NodeJS + Puppeteer để làm nhiều trò hay ho như chụp ảnh màn hình, thu thập dữ liệu, chạy automation test.



(https://github.com/GoogleChrome/puppeteer/)

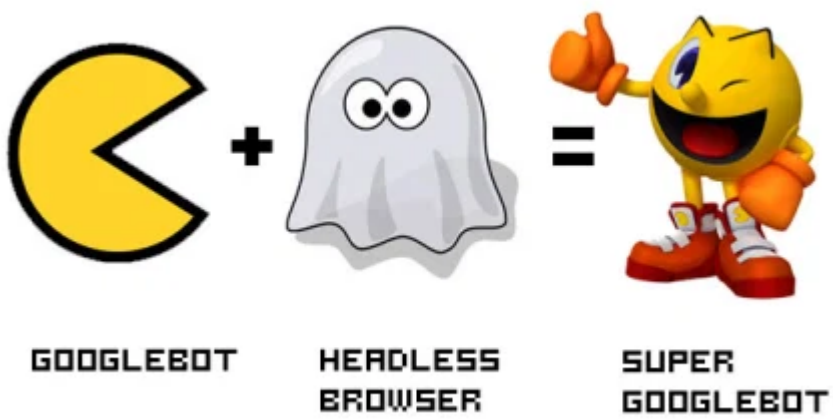
Trang Github của puppeteer

Crawl dữ liệu bằng headless browser có gì vui?

Trước đây mình đã có 1 bài về trsich xuất dữ liệu với HTML Agility Pack (https://toidicodedao.com/2015/07/28/tutorial-trich-xuat-thong-tin-tu-website-voi-html-aglitity-pack/). Tuy nhiên, cách này có một số khuyết điểm sau:

- Chỉ lấy được HTML thuần của trang web. Ngày xưa thì còn ok chứ bây giờ hầu hết các trang đều dùng JavaScript và Ajax để lấy dữ liệu và render (https://toidicodedao.com/2016/08/16/js-truyen-ki-chuong-1-luoc-su-giang-ho/). Lấy HTML thuần thì ta không chôm được gì cả.
- Bên server có một số biện pháp để chặn HTTP Request đơn thuần (dựa theo user-agent, …) nên dễ bị chặn.
- Với một số trang phải đăng nhập mới có dữ liệu, việc quản lý cookie, đăng nhập v…v với HTML Agility Pack rất rắc rối.

Dùng Headless browser, ta giải quyết được toàn bộ những vấn đề trên. Đến **cả Google còn sử dụng** headles browser để crawl các trang web dùng Ajax cơ mà.



(https://toidicodedao.files.wordpress.com/2017/11/headless-browser.jpg)

Chuẩn bị đồ nghề

Lý thuyết đủ rồi, giờ chúng ta chuẩn bị đồ nghề nào để bắt tay vào code nào

IDE: Với NodeJS, ta dùng Visual Studio Code (https://code.visualstudio.com/) là nhẹ và tiện nhất vì có thể dễ dàng debug. VS Code nhẹ lại free, chạy được trên cả Win lẫn Mac nên các bạn tải về dùng nha: https://code.visualstudio.com/ (https://code.visualstudio.com/)

Nếu khả năng trâu bò hơn thì các bạn cứ dùng Notepad ++ hoặc Sublime Text đều được

NodeJS: Để cài NodeJS, các bạn vào https://nodejs.org/en/download/ (https://nodejs.org/en/download/) nhé. Có thể dùng bản 8.9.1 hoặc 9.2 đều ok.

Nếu máy của bạn dùng NodeJS bản cũ hơn thì nên update lên để có thể dùng async/await trong JavaScript (https://toidicodedao.com/2017/10/10/async-await-trong-javascript/), code sẽ trong sáng hơn nhìu.

Khởi tạo project

1. Tạo 1 thư mục mang tên *puppeteer-test*
2. Các bạn mở cửa sổ cmd trong thư mục này, gõ *npm init*, sau đó cứ enter ok hết để khởi tạo project nodejs
3. Tiếp tục gõ *npm install –save puppeteer* để cài puppeteer. npm sẽ tải luôn bản Chrome mới nhất khoảng 100MB nên có thể hơi lâu nhé (xem hình minh hoạ dưới).

```
1 | npm install --save puppeteer
```

(https://toidicodedao.files.wordpress.com/2017/11/screen-shot-2017-11-26-at-11-45-45-am.jpg)

(https://toidicodedao.files.wordpress.com/2017/11/screen-shot-2017-11-26-at-11-45-53-am.jpg)

4. Mở Visual Studio code hay Nodepad++ cũng được, gõ đoạn code sau vào và save lại thành file *index.js*.

```
1  const puppeteer = require('puppeteer');
2
3  (async () => {
4    const browser = await puppeteer.launch({ headless: false });
5    const page = await browser.newPage();
6    page.setViewport({ width: 1280, height:720 });
7    await page.goto('http://kenh14.vn&#39 (http://kenh14.vn&#39);, { waitUntil: 'networkidle2' });
8    await page.screenshot({path: 'kenh14.png'});
9
10   await browser.close();
11 })();
```

view raw
index.js
hosted with ❤ by **GitHub**

5. Từ cửa sổ cmd, gõ *node index.js*. Bạn sẽ thấy Chrome mở lên, sau đó đóng, chụp được file ảnh trang web kenh14.vn

(https://toidicodedao.files.wordpress.com/2017/11/screen-shot-2017-11-26-at-11-42-03-am.jpg)

Thành quả của đoạn code trên, hay ghê chưa

Tạm kết

Vậy là các bạn đã **chuẩn bị sẵn sàng đồ nghề** cho hành trình tiếp theo rồi đấy. Ở phần sau, mình sẽ giới thiệu về API của Puppeteer, sau đó chúng ta sẽ cùng **cào dữ liệu từ kenh14** về để làm kênh 15 nhé.

Các bạn có thể đọc trước về API của Puppeteer để chuẩn bị nhe: https://github.com/GoogleChrome/puppeteer/blob/master/docs/api.md (https://github.com/GoogleChrome/puppeteer/blob/master/docs/api.md)

ARTOO.JS **AUTOMATION TEST** **CHROME** **HEADLESS BROWSER** **HEADLESS CHROME** **JAVASCRIPT** **KENH14** **NODEJS** **PUPPETEER** **TỰ ĐỘNG HOÁ** **TRÍCH XUẤT DỮ LIỆU** **TUTORIAL** **WEB CRAWLER**

18 thoughts on “Làm trò với Puppeteer – Phần 1: Cùng tìm hiểu về Puppeteer và Headless Browser”

1. **An Nguyen** says:
   12/12/2017 AT 8:42 AM
   Hi anh, “dịch cho vui là browser không đầu” chỗ này em nghĩ là headless là không đầu mới đúng

2. **c90mobifone** says:
   12/12/2017 AT 10:03 AM
   cái này có thể chạy js mà không cần browser hả anh huy

   1. ⚜ **Phạm Huy Hoàng** says:
      12/12/2017 AT 10:15 AM
      Bản chất là nó mở lên luôn cái browser chạy ngầm đó e :))

      1. **c90mobifone** says:
         12/12/2017 AT 10:21 AM
         tks a , bài viết hữu ích, có tool để không bị chặn crawler = js rồi :)).

3. **Nguyễn Minh Hiếu** says:
   12/12/2017 AT 12:45 PM
   npm install –save puppeteer lệnh thiếu một cái – trước save anh ơi =)) anh sửa cho các bạn đỡ nhầm

   1. ⚜ **Phạm Huy Hoàng** says:
      12/12/2017 AT 1:11 PM
      2 dấu chứ ko phải 1 dấu đâu em ơi, do cái font nó gắn chung đó :))

4. **Phúc** says:
   18/12/2017 AT 11:42 AM
   Cái này có dùng để cào dữ liệu được không anh

5. **Ha** says:
   26/12/2017 AT 3:09 PM
   mình chẳng thấy cái hình chụp nào cả, cũng không thấy lỗi gì xuất hiện

   1. ⚜ **Phạm Huy Hoàng** says:
      26/12/2017 AT 5:18 PM
      Bạn thử cho toàn bộ code trong đoạn async() và try/catch sau đó log error xem sao nhé 😉

1. **havuong0909** says:
   26/12/2017 AT 8:08 PM
   sau khi post comment thì mình đã tìm ra lỗi, mình thiếu (); ở khúc cuối. Chạy được rồi, cảm ơn bạn nhiều.

6. **kexaque** says:
   05/01/2018 AT 7:55 AM
   Em là dân ngoại đạo về IT nên nếu anh có thể tạo một video hướng dẫn thì tốt quá, thanks anh.

   1. **kexaque** says:
      05/01/2018 AT 8:44 AM
      Sau khi làm theo hướng dẫn , em chỉ thấy trình duyệt chrome hiện lên. Không có chụp ảnh cũng như page “kenh14”.
      http://www.upsieutoc.com/image/4ptAJB

      1. ⚜ **Phạm Huy Hoàng** says:
         05/01/2018 AT 10:05 AM
         Em gõ code bị sai nhé, browser chứ ko phải broweser.
         Bài này anh viết cho dân lập trình nên nếu ngoại đạo em đừng làm theo nha, khi gặp lỗi không tự sửa được, khó nâng cấp mở rộng thêm lắm ;).

7. **kexaque** says:
   07/01/2018 AT 11:53 PM
   Thanks anh, em sửa được rồi, đang làm theo phần 2 hướng dẫn của anh. Hì, dù ngoại đạo nhưng em vẫn thích vọc một chút về IT, có lẽ là tìm bug của web nhưng sợ quá sức mình thôi.

8. **Phan Ngọc Hoàng Anh** says:
   14/02/2018 AT 10:37 AM
   Cho em hỏi ngu cái ạ :V Liệu khi deploy lên các host như Heroku,… thì liệu nó có broser ẩn để chụp hay cào dữ liệu không ạ :V

   1. **Phan Ngọc Hoàng Anh** says:
      14/02/2018 AT 12:06 PM
      À em xem lại thấy nó có down chromium rồi, chắc là chạy được 😀

9. **Phuc** says:
   17/03/2019 AT 9:31 PM
   (node:25320) UnhandledPromiseRejectionWarning: TimeoutError: Navigation Timeout Exceeded: 30000ms exceeded
   at Promise.then (E:\puppeteer-test\node_modules\puppeteer\lib\LifecycleWatcher.js:143:21)
   — ASYNC —
   at Frame. (E:\puppeteer-test\node_modules\puppeteer\lib\helper.js:108:27)
   at Page.goto (E:\puppeteer-test\node_modules\puppeteer\lib\Page.js:656:49)
   at Page. (E:\puppeteer-test\node_modules\puppeteer\lib\helper.js:109:23)
   at E:\puppeteer-test\index.js:7:14
   at process._tickCallback (internal/process/next_tick.js:68:7)
   (node:25320) UnhandledPromiseRejectionWarning: Unhandled promise rejection. This error originated either by throwing inside of an async function without a catch block, or by rejecting a promise which was not handled with .catch(). (rejection id: 1)
   (node:25320) [DEP0018] DeprecationWarning: Unhandled promise rejections are deprecated. In the future, promise rejections that are not handled will terminate the Node.js process with a non-zero exit code.

   Nó báo lỗi vậy là sao bạn

10. **Phuc Pham** says:
    02/07/2020 AT 7:26 AM
    Link die r a Hoàng ơi :((((
    https://github.com/GoogleChrome/puppeteer/blob/master/docs/api.md