HOW I SOLVE WEB SCRAPING PROBLEM

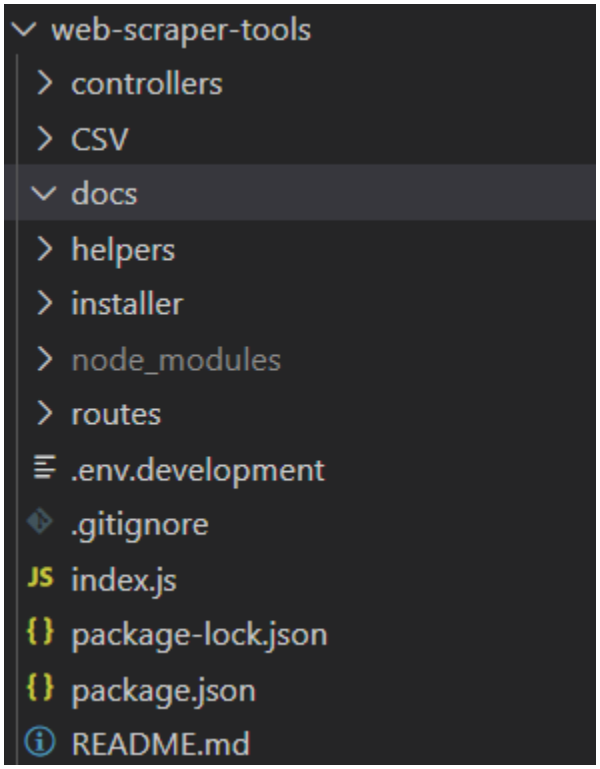
By Melvin Toni Gustave

GIT repository : https://github.com/melvin-toni/web-scraper-tools

---

Step 1 - Create project main structure

At first my attempt is to create my own Restful API to scraping data from Tokopedia, so I initialize the project using NPM and Node.js. I also create main structure for this project by separate between middleware, routes, controllers and file helpers.



Step 2 - Implement website-scraper library

After the project structure completed, I try to find the library and found **website-scraper** library from NPM. Basically how it works is by download the entire files from the URL we are given, like HTML files, image files, JS, etc. For small website it working fine and fast but for website like

Tokopedia it just not a good option, because the process to download is slow and even though download is finished, I need to find necessary data from all those files, parse it and convert it into CSV files. So basically it turned out the process was really slow. **Then I decided not to use this library and try to find another solution**.

Step 3 - Implement Axios and Cheerio Library

After that I found an awesome article from https://pusher.com/tutorials/web-scraper-node, from this web I learn about the concept and how to implement the correct library for web scraper. Then I decided to use **Axios as lib for make an HTTP request and get DOM HTML** also use **Cheerio as a lib to access the DOM easily with jQuery syntax-like.**
With Axios I can get the DOM HTML structure and then with Cheerio I pick the HTML parent of item list, then I looping through that parent div and get necessary data from each DOM.

```
axios(url).then((response) => {
    const html = response.data;
    const $ = cheerio.load(html);
    const parentDiv = $('div.css-1313178.e1nlzfl10 > div');
    const result = [];

    parentDiv.each(function () {
        result.push({
            product_name: $(this).find('a > div.css-16vw0vn > div.css-11s9vse > span').text(),
            description: $(this).find('a > div.css-16vw0vn > div.css-11s9vse span.css-1kr22w3:first-child').text(),
            image_link: $(this).find('a > div.css-16vw0vn').find('img').attr('src'),
            price: $(this).find('a > div.css-16vw0vn > div.css-11s9vse span.css-o5uqvq').text(),
            rating: $(this).find('a > div.css-16vw0vn > div.css-11s9vse > div.css-153qjw7 > div > img').length,
            store_name: $(this).find('a > div.css-16vw0vn > div.css-11s9vse span.css-1kr22w3:last-child').text()
        });
    });

    const fields = ['product_name', 'description', 'image_link', 'price', 'rating', 'store_name'];
    const opts = { fields };
    const csv = parse(result, opts);
    const filePath = './CSV/' + 'result.csv';
    fs.writeFileSync(filePath, csv);
```

Step 4 - Implement Puppeteer Library

For static web or pre-populated web page, Axios work great but unfortunately Tokopedia implement **lazy-loading** for its website, which means the list of the product will be populated after user scroll to certain point. Then for that purpose I use **Puppeteer** library.
Puppeteer basically can simulate user behavior when access page, with these libs I will be able to implement auto-scrolling which automatically activates Tokopedia lazy-loading feature.

```
async function autoScroll(page) {
    await page.evaluate(async () => {
        await new Promise((resolve, reject) => {
            var totalHeight = 0;
            var distance = 100;
            var timer = setInterval(() => {
                var scrollHeight = document.body.scrollHeight;
                window.scrollBy(0, distance);
                totalHeight += distance;

                if(totalHeight >= scrollHeight) {
                    clearInterval(timer);
                    resolve();
                }
            }, 100);
        });
    });
}
```

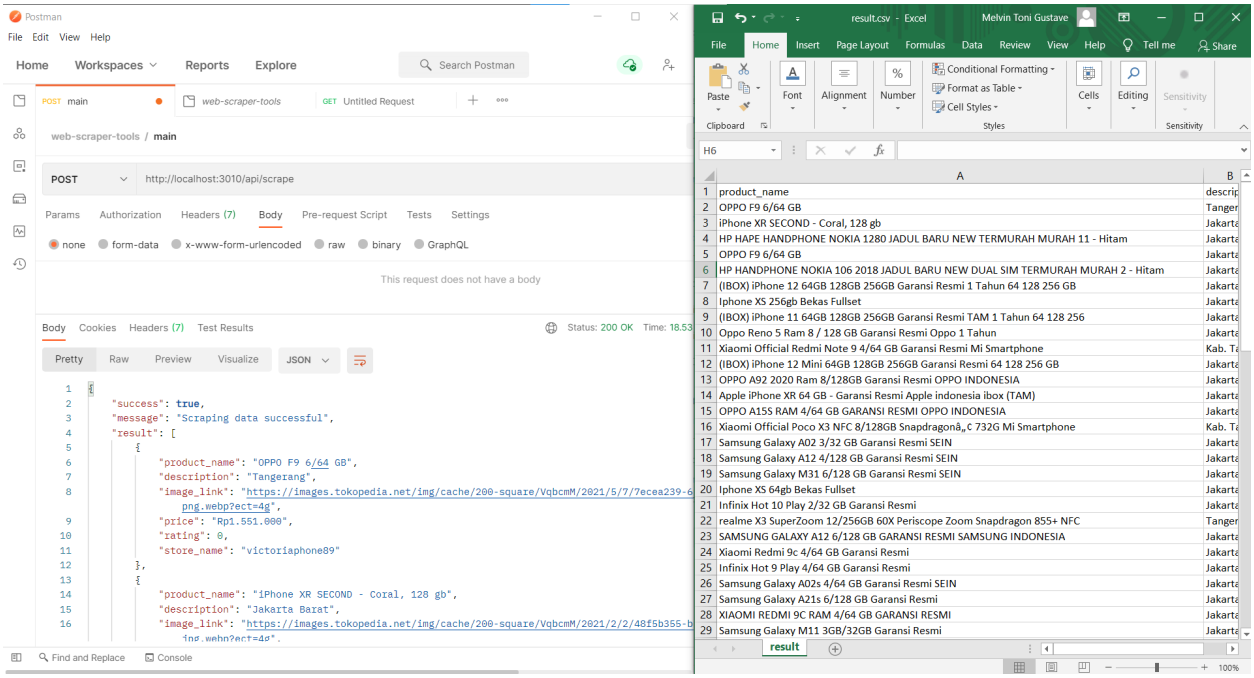
After all data populated, the rest basically same like step no 3 which I will iterate the data and insert into JSON format.

Step 5 - Return data in CSV format

After get all necessary data and return it into JSON format, I be able to convert it into CSV using the **json2csv** library and write the file into **CSV** folder**.**

```
const fields = ['product_name', 'description', 'image_link', 'price', 'rating', 'store_name'];
const opts = { fields };
const csv = parse(result, opts);
const filePath = './CSV/' + 'result.csv';
fs.writeFileSync(filePath, csv);
```

And thus the result appears successfully.


Postman
File Edit View Help
Home Workspaces Reports Explore
Search Postman
POST main
web-scraper-tools
GET Untitled Request
web-scraper-tools / main
POST
http://localhost:3010/api/scrape
Params Authorization Headers (7) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL
This request does not have a body
Body Cookies Headers (7) Test Results
Status: 200 OK Time: 18.53
Pretty Raw Preview Visualize JSON
{
"success": true,
"message": "Scraping data successful",
"result": [
{
"product_name": "OPPO F9 6/64 GB",
"description": "Tangerang",
"image_link": "https://images.tokopedia.net/img/cache/200-square/VqbcmM/2021/5/7/7ecea239-6
png.webp?ect=4g",
"price": "Rp1.551.000",
"rating": 0,
"store_name": "victoriaphone89"
},
{
"product_name": "iPhone XR SECOND - Coral, 128 gb",
"description": "Jakarta Barat",
"image_link": "https://images.tokopedia.net/img/cache/200-square/VqbcmM/2021/2/2/48f5b355-b
Find and Replace
Console
result.csv - Excel
Melvin Toni Gustave
File Home Insert Page Layout Formulas Data Review View Help Tell me Share
Paste Clipboard
Font
Alignment
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Cells
Editing
Sensitivity
H6
A
B
product_name
OPPO F9 6/64 GB
iPhone XR SECOND - Coral, 128 gb
HP HAPE HANDPHONE NOKIA 1280 JADUL BARU NEW TERMURAH MURAH 11 - Hitam
OPPO F9 6/64 GB
HP HANDPHONE NOKIA 106 2018 JADUL BARU NEW DUAL SIM TERMURAH MURAH 2 - Hitam
(IBOX) iPhone 12 64GB 128GB 256GB Garansi Resmi 1 Tahun 64 128 256 GB
Iphone XS 256gb Bekas Fullset
(IBOX) iPhone 11 64GB 128GB 256GB Garansi Resmi TAM 1 Tahun 64 128 256
Oppo Reno 5 Ram 8 / 128 GB Garansi Resmi Oppo 1 Tahun
Xiaomi Official Redmi Note 9 4/64 GB Garansi Resmi Mi Smartphone
(IBOX) iPhone 12 Mini 64GB 128GB 256GB Garansi Resmi 64 128 256 GB
OPPO A92 2020 Ram 8/128GB Garansi Resmi OPPO INDONESIA
Apple iPhone XR 64 GB - Garansi Resmi Apple indonesia ibox (TAM)
OPPO A15S RAM 4/64 GB GARANSI RESMI OPPO INDONESIA
Xiaomi Official Poco X3 NFC 8/128GB Snapdragonâ„¢ 732G Mi Smartphone
Samsung Galaxy A02 3/32 GB Garansi Resmi SEIN
Samsung Galaxy A12 4/128 GB Garansi Resmi SEIN
Samsung Galaxy M31 6/128 GB Garansi Resmi SEIN
Iphone XS 64gb Bekas Fullset
Infinix Hot 10 Play 2/32 GB Garansi Resmi
realme X3 SuperZoom 12/256GB 60X Periscope Zoom Snapdragon 855+ NFC
SAMSUNG GALAXY A12 6/128 GB GARANSI RESMI SAMSUNG INDONESIA
Xiaomi Redmi 9c 4/64 GB Garansi Resmi
Infinix Hot 9 Play 4/64 GB Garansi Resmi
Samsung Galaxy A02s 4/64 GB Garansi Resmi SEIN
Samsung Galaxy A21s 6/128 GB Garansi Resmi
XIAOMI REDMI 9C RAM 4/64 GB GARANSI RESMI
Samsung Galaxy M11 3GB/32GB Garansi Resmi
Tangerang
Jakarta Barat
Kab. Tangerang
result
100%