**Puppeteer Script Documentation: Capture Network Requests and Save JSON Responses**

**Overview**

This Node.js script uses the **Puppeteer** library to automate a browser, capture network requests and responses, and save JSON responses from a specific website (<https://gis.paci.gov.kw/Search/>). The script listens for network requests and responses, then saves specific JSON responses related to a PACI (Personal and Civil Identification) number.

The script:

* Navigates to a page on the website based on a PACI number.
* Tracks all network requests and responses.
* Identifies specific responses containing JSON data related to the PACI search.
* Saves valid JSON responses into a file named after the PACI number.

**Pre-requisites**

Before running the script, make sure to set up the following:

**1. Node.js & npm (Node Package Manager):**

Ensure Node.js is installed on your system. You can download and install it from the official [Node.js website](https://nodejs.org/).

To check if Node.js and npm are installed, run the following commands in your terminal:

bash

Copy

node -v

npm -v

**2. Install Puppeteer:**

The script relies on the Puppeteer package to interact with the browser. To install Puppeteer, run the following command in your project directory:

bash

Copy

npm install puppeteer

**3. File System Permissions:**

Ensure that the script has permission to write to the filesystem, as it saves JSON responses to a file.

**File Structure**

bash

Copy

.

├── script.js # The Node.js script containing the code

└── responses.json # The file that will store all response URLs

**Script Breakdown**

**1. Dependencies:**

javascript

Copy

const puppeteer = require('puppeteer'); // Puppeteer for browser automation

const fs = require('fs'); // File system to write files

const path = require('path'); // Path manipulation for file saving

**2. Sanitize File Name Function:**

This function ensures that the filename used to save the JSON response is safe by replacing any non-alphanumeric characters with underscores.

javascript

Copy

function sanitizeFileName(url) {

return url.replace(/[^a-z0-9]/gi, '\_').toLowerCase();

}

**3. Capture Requests and Responses:**

* The main function captureRequestsAndResponses(paci) opens the page, listens for network events, and saves valid JSON responses.
* It logs request URLs and response URLs.
* It identifies JSON responses related to the PACI number, then saves them to a JSON file.

**4. Saving JSON Responses:**

If a response contains JSON data related to a search request (search?params=), it is saved as a .json file with the PACI number in the filename.

javascript

Copy

const fileName = paci + '.json';

fs.writeFileSync(filePath, JSON.stringify(responseBody, null, 2));

**5. Handling Navigation:**

The script navigates to the URL constructed with the PACI number.

javascript

Copy

await page.goto(parentUrl + paci);

**6. Command-Line Arguments:**

The PACI number is passed via command-line arguments:

javascript

Copy

const paciNumber = process.argv[2]; // PACI number passed as second argument

**7. Script Execution:**

After capturing requests and responses, the script waits for a specified period before saving the URLs and closing the browser.

**How to Run the Script**

**1. Clone the Repository or Save the Script:**

Save the provided script as script.js in your project directory.

**2. Run the Script:**

Open a terminal/command prompt and run the script using Node.js, passing the PACI number as a command-line argument:

bash

Copy

node script.js <PACI\_NUMBER>

Replace <PACI\_NUMBER> with the actual PACI number you wish to search for. For example:

bash

Copy

node script.js 12345678

**3. Expected Output:**

* The script will navigate to https://gis.paci.gov.kw/Search/<PACI\_NUMBER>.
* It will log all network requests and responses to the console.
* Any JSON responses related to the search will be saved in the same directory with a filename like 12345678.json.
* A file named responses.json will also be created, containing all response URLs captured during the session.

**4. File Saved:**

* **JSON Responses**: For each valid JSON response, a file will be saved with the PACI number as the filename.
* **All Response URLs**: All response URLs will be saved in responses.json.

**Error Handling and Logging**

* If no PACI number is provided as a command-line argument, the script will log an error and exit:

javascript

Copy

if (!paciNumber) {

console.log("Please provide the PACI number as a command-line argument.");

process.exit(1);

}

* Logs are displayed in the console for each network request and response, which will help in debugging and monitoring the requests made during the session.

**Customization Options**

**1. Timeout Duration:**

The current script waits for 6 seconds (await page.waitForTimeout(6000);) after loading the page to allow enough time for requests and responses to be captured. You can adjust the timeout by changing the number of milliseconds.

**2. Change Search URL:**

If you need to use a different base URL for the search (for example, a different website), you can modify the parentUrl variable:

javascript

Copy

const parentUrl = 'https://yourwebsite.com/Search/';

**3. Response Filtering:**

If you want to filter other types of responses (besides JSON), you can update the page.on('response') event listener to include other checks.

javascript

Copy

if (responseUrl.includes(targetUrlPattern)) {

// Add more filtering logic for other response types

}

**Troubleshooting**

* **Issue**: The script doesn’t save any files or doesn’t work as expected.
  + **Solution**: Ensure that you have installed Puppeteer (npm install puppeteer) and that the PACI number passed as an argument is valid. Check if your internet connection is stable for Puppeteer to load the page.
* **Issue**: The script runs in headless mode, and you want to see the browser.
  + **Solution**: Change headless: true to headless: false in the puppeteer.launch() function to open the browser visually.

This document covers how to use the provided script, its expected behavior, and customization options. Let me know if you need any further details or help with modifications!