**Documentacion**

First we install this 3 modules:

npm i puppeteer

npm i puppeteer-extra

npm i puppeteer-extra-plugin-stealth

The code:

Here we call all the dependencies and the password from a separate file

const { Keyboard } = require('puppeteer');

const puppeteer = require('puppeteer-extra');

const StealthPlugin = require('puppeteer-extra-plugin-stealth');

const { password } = require('./password')

Now you set your credentials.

const googleUsername = "Your Email";

const googlePassword = password;

We set the options for the new browser.

The options object passed to puppeteer.launch() contains the following properties:

* headless: This option is set to false, which means the browser will launch in non-headless mode, and the browser window will be visible during the automation process.
* args: This property is an array of command-line arguments to customize the behavior of the browser. In this case, the following arguments are provided:
  + --no-sandbox: Disables the sandbox for the browser, which can be useful in certain environments.
  + --disable-gpu: Disables the use of the GPU, which can be helpful in preventing some rendering issues.
  + --enable-webgl: Enables WebGL, a JavaScript API for rendering interactive 2D and 3D graphics in the browser.
  + --window-size=800,800: Sets the initial size of the browser window to 800 pixels by 800 pixels.

(async () => {

   const browser = await puppeteer.launch({

      headless: false,

      args: [

         '--no-sandbox',

         '--disable-gpu',

         '--enable-webgl',

         '--window-size=800,800'

      ]

   });

You declare the variable of the url you want to visit (in this case this code works with a google login)

const loginUrl = "Your URL”;

The User Agent is the variable that tells the new browser how to render the content

   const userAgent = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.91 Mobile Safari/537.36';

With these variables set you now open a new browser and tell what User Agent needs to use.

const page = await browser.newPage();

await page.setUserAgent(userAgent);

Now you go to the link you decide.

await page.goto(loginUrl, { waitUntil: 'networkidle2' });

And finally you tell what it should do, type in the input type email the variable we set before and then press enter. And then Wait for 6 seconds and type the password and press enter.

await page.type('input[type="email"]', googleUsername);

await page.keyboard.press('Enter');

await page.waitForTimeout(6000);

await page.type('input[type="password"]', googlePassword);

await page.keyboard.press('Enter');