Black Box Testing

Software testing technique

Prepared By

|  |  |  |
| --- | --- | --- |
| Ottor Mills | ID#: 180917 | Email: 180917@gist.edu.cn |
| David Thomas | ID#: 180912 | Email: 180912@gist.edu.cn |
| Nicoy Smith | ID#: 180902 | Email: 180902@gist.edu.cn |
| Kenneth Anglin | ID#: 180907 | Email: 180907@gist.edu.cn |

|  |  |
| --- | --- |
| Course Instructor: | Thomas Canhao Xu |
| Course: | SWEN3010 |
| Date: | April 30, 2019 |

# Table of Contents

# What is Puppeteer

Puppeteer is an automated browser testing framework for NodeJS. It provides a high-level API to the developer tools of Google Chrome or Chromium. Puppeteer makes an excellent tool at executing tasks such as:

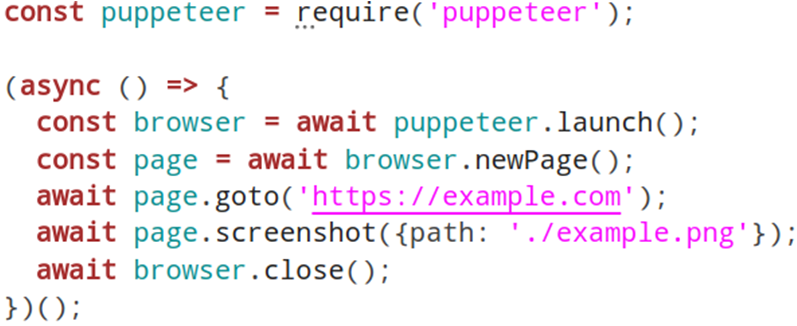
* Automated Form Submission
* Generating Screenshots and PDFs of pages
* UI Testing, Keyboard input etc.

Since Puppeteer is interfaced using JavaScript a high level of consistency exists between the code written to control Puppeteer and controlling a website using client-side JavaScript. Puppeteer offers a headless mode, making it compatible with command line operating system. Headless mode drastically conserved system resources which makes it more efficient at executing test cases and allows easy concurrency.

# Using Puppeteer

All functions provided by Puppeteer are asynchronous. As such if desired test cases are to be ran sequentially they will have to be implemented using the async/await model. Promise chaining may be used however this will result in code that becomes harder to read as mode test cases are added.

The following code snippet depicts how to take a screenshot of a website using async/await:



The following code snippet depicts how to take a screenshot of a website using promise chaining:

Specific Functionalities

|  |  |
| --- | --- |
| Task | Function |
| Starting New Browser | browser = await puppeteer.launch({headless: false}); |
| Opening New Tab | browser.newPage(); |
| Setting User Agent | page.setUserAgent(“<desired user again>”); |
| Navigate to New Page | page.goto(“<desired url>”); |
| Set Page Navigation Timeout | page.setDefaultNavigationTimeout(<desired milliseconds>); |
| Wait for Page Navigation | page.waitForNavigation(); |
| Evaluating Javascript | page.evaluate(`<javascript code here>`); |
| Keyboard Typing | page.keyboard.type("<desired text>"); |
| Taking Screenshot | page.screenshot({path: 'example.png'}); |
| Page to PDF | page.pdf({path: './test.pdf’, format: 'A4'}); |
| Close Browser | browser.close(); |

Puppeteer vs Selenium/WebDriver

Puppeteer focuses on Chromium and is developed and maintained by Google. Selenium aims to run test cases across multiple browsers. As such Puppeteer provides a higher level of reliability than as with Selenium when it comes to automated testing Test cases that are successful in Chromium may not necessarily be the same with other browsers.