

Overview

Choice Hotels International contacted NAU to solve a problem. Their problem was that they had a large amount of websites to manage, and all were required to work correctly at all times. To achieve this reliability, Choice Hotel utilizes automated testing programs to show that their websites work as they should. These programs are called *testing suites*. However, even with these automated testing suites, testing that many web pages efficiently is difficult, as the suites require information from web pages that are constantly changing and updating. This had become quite an issue for Choice. That's where we, Team WATER, come in.

We created a Google Chrome Extension that *retrieves the information from a webpage to be used in testing suite* extremely quickly and reliably. This transformed a very time consuming process into a simple click, saving Choice Hotels tremendous amounts of time and resources.

Figure 1, below, shows an example of the interface of our project. Figure 2 shows results of our extension: output files that can be directly or indirectly used in a testing suite.

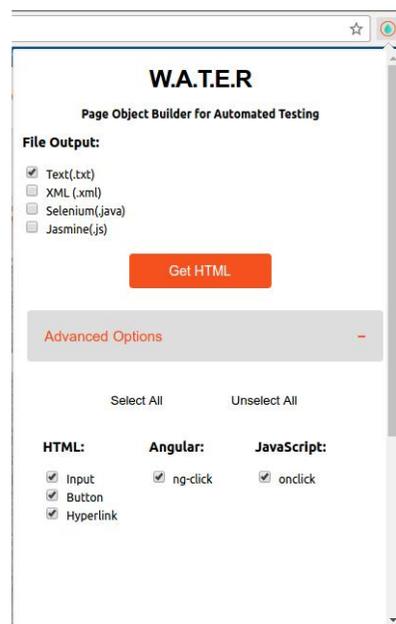


Figure 1: Graphical User Interface when the extension is opened.



Figure 2: Files generated by the extension are downloaded right into the users browser.

The Problem

Websites need to be tested in order to provide the reliability that is demanded in a web dominated world. However, websites are constantly changing, and automated testing suites need updated information from a webpage to perform testing. This requires gathering information from *all elements of a web page*. These elements make up everything you see on a web page, such as buttons and links. Because there are so many, this is not always easy to do. Figure 3 shows a diagram of the old process of accomplishing this.

Workflow without our Tool

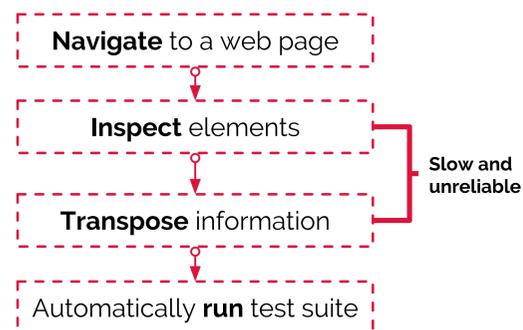


Figure 3: A workflow diagram of the old process for testing websites.

Inspecting the elements on a webpage (looking one layer below what most people see, to the source code that generates the view) and then moving that information over to a testing suite is problematic:

- It takes a long time; to do this for all of Choice hotels' webpages, for example, would take over 1,100 hours of expensive developer time.
- It is error prone as every change needs to be *manually* updated.

Moving Forward

Currently the completed tool is available on the Chrome web store for free use. An important part of the development process was a focus on developing software that can be easily understood and modified by future developers in the open source community.

Version 2.0 Wishlist:

- More file support (e.g. Dedicated Jasmine functions instead of a file containing just JavaScript objects).
- More advanced element/libraries supported (such as jQuery or even frameworks like React).
- Further user options to customize the output files and elements retrieved.

The Solution

We have developed a Google Chrome extension to solve this problem. The extension automates the process of getting elements from a webpage to a test suite. With our extension a user never needs to manually inspect or transpose the element data. Instead, the extension does this for a user automatically. Figure 4 shows the new workflow of a user getting the required website information for a testing suite when using our extension.

Workflow with our Tool

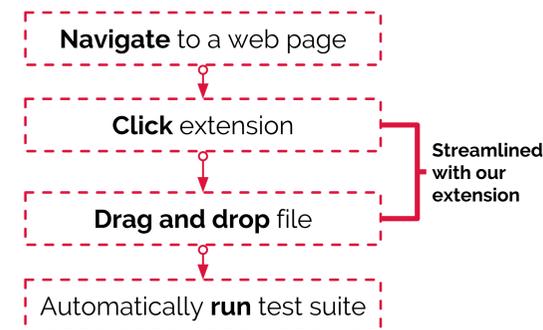


Figure 4: A workflow diagram using our tool.

By using this extension the overall time it takes to inspect and transpose elements of a webpage is *greatly* reduced, from hours to seconds.

Results

We saved Choice hotels an estimated:

- \$35,000 in initial setup costs.
- Upwards of \$15,000 saved in upkeep costs and bug fixing maintenance.
- Total savings of \$50,000 annually.

The extension is a free, open source piece of software for the public domain and for general use.

Credits

The project was developed as a Google Chrome Extension and utilizes functionality from the Chrome API.

We would like to extend our thanks to Dr. Elwakil, NAU, and especially everyone at Choice Hotels.

Architecture

We designed the architecture of the extension to use three main modules within the extension; the frontend, middleware and backend. Each module communicates with one another through data messages. Figure 5 shows how each of these modules are connected.

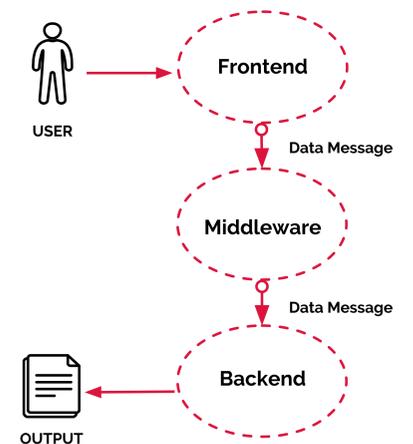


Figure 5: The architecture of the system.

- The frontend module handles the user interface of the extension as seen in Figure 1.
- The middleware processes the current webpage, pulling all elements from it and packaging it to be sent to the backend module.
- The backend receives the element information from the middleware and creates the formatted output file(s). The end result is various files (automatically downloaded into the browser) that can be used with various testing suites. Figure 2 shows examples of some downloaded files.

Learn More

Presentation: April 28, 2017, 8:30 AM,
Walnut AB Room, NAU

Feel free to ask us anything (if we're here).

Alternatively, our contact info is:
mw834@nau.edu - Max Wason
pch27@nau.edu - Peter Haschke
jl2569@nau.edu - Jason Le
jes445@nau.edu - John Sadie