NIBBLE Vision Statement
Tong Wu, Mark Oparowski, Jimmy Le, Andrew Duong

Project Description

Documentation is one of the most prominent issues for a developer creating or using an application program interface (API). The software developer of the API must create and maintain the document, a process considered by most programmers to be the most tedious part of the software development process. Meanwhile, developers wanting to use the API must hunt down the documentation and hope that it exists, is complete, and is up-to-date. TADAPI (The Automated Discovery of Application Program Interfaces) from Novacoast x Nibble is an automated API discovery web application that produces deliverable API documentation in the most intuitive way: through the simple navigation of websites or applications that use the API calls.

Outcome

Documentation is currently written in two methods, either after the software is developed, or while the software is being developed. The first method requires that the developer create a finished product, then backtrack and tediously write out the API functionality by going through their code and memory. This often yields incomplete and sometimes inaccurate information. The second method requires that the developer take intermittent breaks throughout the coding process to document their code, distracting the coders from their main purpose and thought process. TADAPI simplifies the documentation process and addresses the downside of the two current methods by creating archives for API functionality through an intuitive automated tool. TADAPI runs in the background by scanning HTTP requests, then analyzing the

responses in order to create and log API functionality as the user navigates through the web application that uses the API calls. Even non-technical application users can help generate documentation simply by running TADAPI as they navigate through these web applications. TADAPI also makes the maintenance of documentation after API updates extremely simple because it updates its own API database immediately after TADAPI is used on the updated web application. TADAPI is an easy and foolproof way to comprehensively document API functionality without having to backtrack through code or breaking the development process.

Similarly, writing a web application that uses external API calls requires developers to search up documentation for the API's they wish to use. If they are lucky, they will find up-to-date and accurate listings of the API's functionalities, otherwise, the developer would have to run repetitive tests on the API to determine its functionalities. TADAPI creates an up-to-date archive for API's as they are discovered, but if a certain web application does not have its API functions documented, the developer can run TADAPI in the background and discover its functions by navigating the application instead of tediously running numerous sets of repetitive tests.

TADAPI makes API discovery and documentation easy enough for developers to delegate these same tasks to non-developers.

Implementation

TADAPI is primarily interfaced through a website with the backend utilizing the Ruby programming language. Traffic monitoring is implemented via the PCAP libraries and CouchDB will serve as the database for TADAPI. The front end will utilize Ruby on

Rails along with other traditional web technologies such as javascript, CSS, and HTML.