Memorandum

TO: Dr. Kate Kipskey
FROM: Hal Nguyen
DATE: September 24th, 2023
SUBJECT: Enhancing Web Services API

After exploring Ocean Networks Canada's Web services API, I've discorvered some vulnerabilities, as well as issues with accessibilities with the front facing client. I have created this memorandum to address the mentioned issues, as well as providing some possible solutions.

Client's Profile

Ocean Networks Canada (ONC) is an not-for-profit society, hosted and owned by University of Victoria. ONC provides ocean data observed from various equipments, making ocean intelligence accessible to the public.

One of the many ways for ONC to make this possible is through their Web Services API (Ocean Data 3.0 portal). While it is a valuable tool, the developer experience could still be improved, as well as the security of the web infrastructure.

Problem Definition

To enhance the Web Services API, the problem can be defined as following:

- Need statement: To improve the web interface, and to enhance the current rate limiter [1].
- Goal statement: The targeted consumers are software developers/engineers, and data scientists. Having a documentations is crucial for all API consumers alike. Since this service is completely free, it should also have sufficient protection against malicious agents, starting with prevention of Denial-of-Service attack (DDoS) [2].
- Objectives:
  - Documentations: Improving the API consumers experience with transparent documentations.
  - Accessibility: Reasonable contrast for documentation portal.
  - Web Security: The rate limiting mechanism protects against site scrapers and bots.
- Constraints:
  - Funding: Software developers are expensive, even for small changes.
  - Backwards Compatibility: Any changes happened to the API will need to be able to maintain backwards compatibility (not introduce any breaking changes to the current consumers).

Solutions

These are 3 solutions provided for the 3 mentioned issues, respectively: documentation, accessibility, and security.

Documentations

In order to consume the API, you need to obtain an API key from OCN, and with each request, this secret key has to be attached to the request targeted URLs in form of query parameters, for example the following request will returns all locations that OCN has their equipments active:

```
https://data.oceannetworks.ca/api/locations?token=<your-secret-api-key>
```

When first landing on the documentations site, the "Before you get started" section should mention how to:

1. Obtain the secret API key,
2. How to attach this key to each request (request header, URL parameters, etc etc..),
3. The base URL for each services ("data.oceannetworks.ca/api").

Accessibility

Increasing the contrast by having a darker color on the text. The two figures below show an example of what the current site looks like (Figure 1), and what it could be (Figure 2):

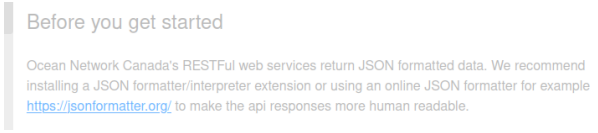

Figure 1: Current site implementation.

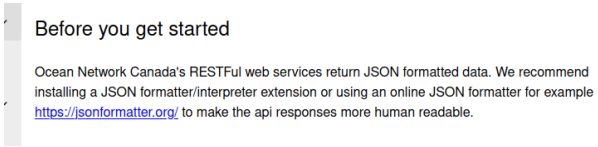

Figure 2: Suggested site implementation.

Security

The current rate limiter implementation is done with a session ID, assigned via HTTP cookie, which protects the infrastructure from attackers who are on a web browsers. However, a script can be written in a such way that on each request, while having the same API key, does not encapsulate the HTTP cookie, and thus allow the attacker to make as many request as they like, which would overload the server, and thus preventing other users from consuming this service.

A better solution for rate limiting is to **index each request by the API key** instead of their session ID, this will now eliminates the thread of DDoS on both browsers or individual scripts.

References

[1] "What is Rate Limiting?", Cloud Flare Learning, https://www.cloudflare.com/learning/bots/what-is-rate-limiting/, (accessed Sep. 24th, 2023).

[2] "What is a DDos attack?", Cloud Flare Learning, https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/, (accessed Sep. 24th, 2023).