Content for API page. Have text centered, double spaced. Do that for paragraphs. Then steal someone’s settings for code setting apart. Have that be a div with class code, and have CSS do that as the set apart code. No run buttons, except for at the end.

Pages,

Introduction

Should have what the webpage is, the link to the API documentation, and a basic explanation of what the purpose is.

A tale of two APIs,

Link the article talking about both of the APIS. Show code to do the first api, the simple get url request for the first API. Talk about how the first API is deprecated and that we should use the second because it is richer.

V2 API,

Link to the good documentation. Explain a bit more in depth the three different ways that one can use the API. Go in to detail on the one he suggests and show your code.

Custom API for lists of email addresses. Check which of your friends is a cheater.

Here go in to detail on the API that I created. It should take a post request because posts don’t have length problems. The POST requests takes out the email addresses and makes get requests. Saves them in a JSON object. Return the JSON object from the POST request when you are out of email addresses. Suggest that it could be an Android app where you upload your address book and get a table of people with email addresses linked in Ashley Madison. Make sure to make a table that lists all of the email addresses provided by the JSON and indicates yes or no.

Introduction

The website haveibeenpwned.com is a website that collects the username/password/any other available data from leaks that have occurred from various companies/websites. The information is quite detailed, and is available to search particular addresses.

The point of the website is that you can see if any of the accounts that you have created have been compromised. If passwords were leaked, it will tell you that as well. If a password was leaked, you should change any other account you might use that password on.

Haveibeenpwned.com has an API that is publically available. It is on the simpler side of API’s, but can take a bit to get set up. You can get their documentation at <https://haveibeenpwned.com/API/v2>

Very Lax API Rules

Haveibeenpwned.com has very lax API usage rules. No API key is necessary, there is no limiting on the number of requests. You must use SSL, but it will just convert your request if you do not. Cross-Origin Resource Sharing is also fully allowed.

Version 1 of the API is much simpler, and does not require a API key or any specification of who is making the request. Version 2 of the API does not require an API key, but does require that there be a User-Agent specified in the request header. More on both of these APIs on the next page. This data is all available elsewhere on the internet in bulk form so the owner of the webpage is not overly concerned with someone abusing the service.

A Tale of Two APIs

The older version of the API (version 1) was extraordinarily simple. The user merely had to send a get request to [https://haveibeenpwned.com/api/breachedaccount/{account}](https://haveibeenpwned.com/api/breachedaccount/%7baccount%7d) substituting the account information of course. No key or request headers necessary. JavaScript code that accomplishes that trivial task is below.

INSERT SIMPLE CALL

The second version of the API is still pretty simple, but actually has three methods that a person can use to access it. On top of the 3 methods, there are now more possible things to request from the API. V1 did just a JSON list of accounts that have been hacked for a particular email address. In V2 you can use the API to get breaches for an account, but also for all breached sites that they have record of, or all records for a single breached site, or for others.

The three methods to access the V2 API are interestingly discussed in “Your API versioning is wrong, which is why I decided to do it 3 different wrong ways”. In that article the owner of

A tale of two APIS

The first API that was set up by haveIbeenpwned.com is the version 1 API. That is still accessible, and is very simple to use a simple url get request of: