



Consuming APIs
for Fun and Profit



Carly Ho



Sr. Engineer, Clique Studios (here!)



@carlymho





What's an API



- * **A**pplication **P**rogramming **I**nterface
- * A way to programmatically retrieve data for display, transformation, etc.
- * Usually accessed via an **endpoint** URL
- * Usually returns data in **JSON** format





How to work with API data?



- * In PHP, we have **file_get_contents** and the native **cURL** ("client URL") implementation
- * We can also use Javascript in any page to perform API calls on particular browser events





⚠️ *A Word of Warning* ⚠️



- * There is danger inherent to getting data from a third party
- * External servers might be compromised unexpectedly
- * Sanitize data like you would your form inputs!





Using Existing APIs



Getting API Data



- * The easiest way is to use **file_get_contents**('https://your-url.com /'), which, when returned to a variable, will fetch the contents
- * However, it'll be a string, rather than an object we can use





JSON and How to Parse It





- * JSON (**J**ava**S**cript **O**bject **N**otation) is the format for most API data
- * When we use JavaScript to fetch API data, it can usually use it right away, but in PHP, whether we use `file_get_contents` or `cURL` we need to parse it
- * **`json_decode`**(\$string, true) turns the string into a JSON object.





API Keys/Tokens?



- * An API might tell you that you need a **key** or a key and **token** to use it.
 - * API keys are unique to accounts to determine sans credentials who's using the account for security or account privilege purposes
 - * Tokens are generally application specific and paired with a key
- 
- 



Sending Headers



If you're using `file_get_contents`, headers are added in the options parameter. This is where you usually add your key and/or token.



```
● ● ●
$options = array(
    'http' => array(
        'method' => "POST",
        'header' => "Key: slkjfdajldskfjeijflasdkjf"
    )
);
$content = stream_context_create($options);
$file = file_get_contents('http://www.example.com/', false, $content);
```





Prototyping APIs



When to Prototype





- * If you have a big spreadsheet or CSV but not time to DIY an application from scratch
- * Say, if you want to build some interactive charts on a webpage from spreadsheet data





Prototyping Tools



- * Google Sheets has its own API
<https://developers.google.com/sheets/api/>
 - * Airtable a hosted database application that allows API access
<https://airtable.com>
- 
- 



Build Your Own API



Why Have an API?



- * Gets people to engage with your application
- * You may want to access some of your data asynchronously from within the application
- * Building mobile apps





What You Need



- * A database of information you want to access in whole or in part
- * A publicly-accessible PHP file that you can point requests to

(that's it!)





Sending Headers



To send a response as JSON, before outputting your object, you need to send a content-type header



```
header('Content-Type: application/json');
```







Returning JSON



If your data can be put into an array or object, you can pass it through the **json_encode** function and echo the result after the header.

If your content isn't conveniently available in that form, you can also manually format it:



https://en.wikipedia.org/wiki/JSON#Data_types,_syntax_and_example





How to Issue API Keys



- * Each key should be unique
 - * Store keys in user records and query for a match
 - * You can do this manually for small applications, or automatically on account creation
 - * An API key is like a password, so it should be possible to issue a new one
- 
- 



APIs for Other People's
Stuff



The End of Public APIs



- * Big tech companies keep deprecating APIs. Why?
- * To force traffic into their own channels
- * To prevent users from getting around ads
- * In response to security holes being discovered





Web Scraping



- * Gets the content of an HTML page and parses it as XML nodes
- * Classes such as DOMDocument can navigate these nodes and return the content you want
- * More laborious and less reliable (esp. since page layouts can change)





⚠ *More Warnings* ⚠





- * This might be against the terms of service of a website
- * This can also be kind of rude if you're using up all of a site's bandwidth by scraping the content
- * You might get rate-limited or your script may get blocked, or your account might get suspended





Cron Jobs



- * Polite way to fetch data: on specified intervals, rather than than on page load
 - * The **crontab** ("cron table") runs programs on scheduled intervals (see crontab.guru for formatting help)
 - * You can use the scheduler to import data to a database by script a few times a day so you don't overload services
- 
- 



Not an Ideal Solution



- * This is not a solution for a production application
- * If a site you use doesn't have a public API, consider getting in touch to ask if that's on the roadmap or could be
- * If it's an open-source project, consider putting in a pull request to add API support





Now: go forth and build things!



@carlymho
@carly@kitty.town
carlymho.com

