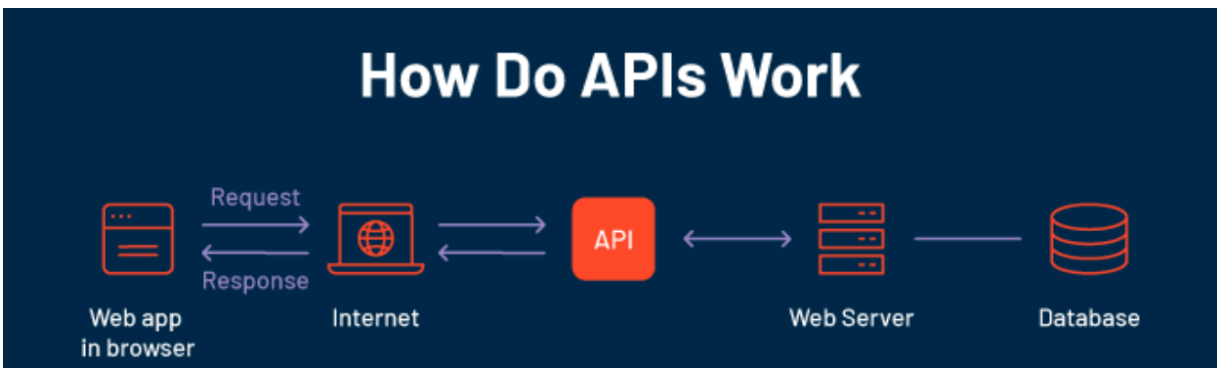


## PHP and API's - Written by Anthony Adams

At the end of the day, the internet is just a bunch of data we are all trying to access. How we access that data is key. In some cases we can grab data for free because its public and allows for it. In which case we can scrape it. However if the target data we want is not public, we need access to it granted by the owner of the data. This could be the end user or the owner of the service (like Facebook or something). This is where API's come in.



An API (Application Programming Interface) is a mechanism what allows 2 different components of an app, webapp or DaaS to communicate with each other using a protocol or a set of protocols. API communication is usually done the same way you already understand

networking communication where there is a client and a server. The application that is sending the request is the client and the service that is answering the request is the server.

Many popular web apps have an API that can allow developers to get data from a particular service and use it somewhere else. For example Twitter has an API that allows developers to get access to data from the platform.

## Twitter API

---

The Twitter API enables programmatic access to Twitter in unique and advanced ways. Tap into core elements of Twitter like: Tweets, Direct Messages, Spaces, Lists, users, and more.

[Subscribe to Pro access](#)

[API access levels and versions](#)

[Try a live request](#)

---

When using an API there will be certain things we can and cannot do and that will be defined by the service in question. Also some APIs may require a paid subscription or may be free to a certain point and then you may have to pay

depending on usage.

## Using PHP with APIs

There are a few ways we can call an API with PHP. Let's look at a few of them one by one.

### `file_get_contents`

This function is used to read the content of a file and return it as a string, It allows you to retrieve content in a file and use it in a php script.

To use it you simply provide the function with the file path or URL you want to read. PHP will attempt to open the file or URL and read its contents.

If the file is accessible and readable the `get_file_contents` function will read the entire content and return it as a string. You can assign this string to a variable and return the variable.

This is a simple way to read the content of a file

or URL but not very suitable for more complex operations. For example this is not the best way to extract specific data from an entire file. There are ways you can do that using this method by using other means such as arguing a strpos argument but this method might not be the best approach.

```
<?php
//scrape using get_file_contents
$html = file_get_contents("https://
humber.ca/");

echo $html;

?>
```

**See files:**

**[week8/get-file-contents-scrape.php](#)**

**[week8/get-file-contents-api.php](#)**

**[week8/getfilecontent/](#)**

## **cURL**

cURL (Client URL) is a library in PHP. It allows

you to send and receive data from a URL. It provides a way to make HTTP requests and interact with web services.

To use cURL in PHP you 1st have to initialize the cURL session:

```
// create a new cURL resource  
$ch = curl_init();
```

Then set options:

```
// set URL and other appropriate  
options  
curl_setopt($ch, CURLOPT_URL,  
"https://www.google.ca/");  
curl_setopt($ch, CURLOPT_HEADER, 0);
```

Execute the cURL request:

```
// grab URL and pass it to the browser  
curl_exec($ch);
```

Once done with the request close it:

```
// close cURL resource, and free up  
system resources
```

```
curl_close($ch);
```

cURL is pretty flexible and a powerful way to interact with remote servers from PHP. It supports various protocols like HTTP, HTTPS, FTP and more. You can also tailor make requests by setting headers, cookies, handle authentication and so on. You can use cURL to handle API requests, scrape web content, upload files and more.

See File:

[week8/cURL.php](#)

## SDKs

An SDK (Software Development Kit) is like a set of tools that can be used to build applications more easily. It provides pre built components, libraries and tools that streamline the development process. Think of it conceptually as giving you a set of building blocks to create some app without having to

start from scratch.

The SDK typically has documentation, sample code and API's that show you how to use, implement and configure. Remember that every API is different and may require different methods to use and implement. Please refer to the documentation of the platform you wish to use an API from and acquire the appropriate keys to access any API you need to access.

**See file: twitter-sdk.php**

## The Google Books API

To use the Google Books API, you need to follow these steps:

### **Create a Project and Enable the Google Books API:**

- Go to the Google Developers Console (<https://console.developers.google.com/>).
- Create a new project or select an existing project.

- Enable the Google Books API for your project.
- Obtain an API key for accessing the API.

### **Set Up Your Development Environment:**

- Install a PHP HTTP client library. You can use Guzzle, as we discussed earlier, or any other HTTP client library of your choice.

### **Make API Requests:**

- Use the HTTP client library to send HTTP requests to the Google Books API endpoints.
- Include your API key in the request as an authentication mechanism.

```
<?php
// Set the API key
$apiKey = 'YOUR_API_KEY';

// Set the base URL for the Google
Books API
$baseUrl = 'https://
www.googleapis.com/books/v1/';

// Define the search query
```

```
$query = 'some book';

// Build the request URL
$requestUrl = $baseUrl . 'volumes?
q=' . urlencode($query) . '&key=' .
$apiKey;

// Send the HTTP request
$response =
file_get_contents($requestUrl);

// Check if the request was successful
if ($response !== false) {
    // Decode the JSON response
    $data = json_decode($response,
true);

    // Process the response
    if (isset($data['items'])) {
        foreach ($data['items'] as
$item) {
            // Access the book
information
            $title =
$item['volumeInfo']['title'];
            $authors =
$item['volumeInfo']['authors'];
```

```
        // Display the book
information
        echo "Title: $title<br>";
        echo "Authors: " .
implode(', ', $authors) . "<br><br>";
    }
} else {
    echo "No books found.";
}
} else {
    echo "Error occurred while making
the request.";
}
?>
```

**See file: google-books.php**

## Outputting CMS content as JSON

You can also convert data from a web service or CMS system to json. Sometimes it might be advantageous to do so. Because:

Data Interoperability: JSON is supported by

many programming languages so it can serve as a good go between from system to system.

It's lightweight: It's quick, lightweight and efficient.

Easy to parse: JSON has a simple syntax making it easy to work with.

Easy to read: JSON is designed to be easy to read by humans.

Compatibility with web technologies: It is easy to use with web development. It is natively supported by JS and many web APIs.

Again you will need to consult with the API you want to work with and read its documentation to see what steps you need to take to convert get and convert its data to JSON. You will also need an API key.

**See file: [twitter-json.php](#)**

