Integrating Software

APIs and SDKs

TAMID GROUP

What Are They?

- It is important to realize that many of the problems you will face as a coder have already been solved. APIs and SDKs are used to implement software developed by other people, so they are valuable tools to save you time as a developer.

- **Application Programming Interface (API):** An API is a set of functions provided by an application that can be called by other programs in order to access features or data from the application. We will be focusing on RESTful APIs, which are called using an *HTTP request*.

- **Software Development Kit (SDK):** An SDK is a collection of tools that developers can use when making an application with a framework or platform. They often include modules or packages that you can import into your code. SDKs are typically necessary when your application is built around a particular framework or platform. For instance, when developing an Android app, you would use the Android SDK, which has tools to help you interface with the phone.

https://www.lexico.com/en/definition/api

Using APIs

Find Your API

When implementing a feature, search the internet for a relevant API. For instance, you might search "Weather API" if you need the weather for your app. Once you find the right API, create an account so you can get your API key.

Read the Documentation

APIs come with important documentation that explains the parameters of your API call and the return value. APIs return values in readable formats like JavaScript Object Notation and XML. If you don't know JSON, use JSON Editor to understand the results.

Call the API in Your Code

Once you are familiar with the documentation, you are ready to call functions in the API using HTTP Requests

Calling The API (JavaScript)

Asynchronous Function

Async functions enable us to write promise based code as if it were synchronous, but without blocking the execution thread. It operates asynchronously via the event-loop. Async functions will always return a value. Using async simply implies that a promise will be returned, and if a promise is not returned, JavaScript automatically wraps it in a resolved promise with its value.

The Route

All HTTP Requests require a route–the link here. The API Reference will tell you how to format the API call, which will often include your API key and other information.

```javascript
/** Uses Fetch API to make API call. */
async function putData(link, data) {
  const response = await fetch(link, {
    method: 'GET',
    body: `${data}`
  });
  return response.json();
```

Calling The API (JavaScript)

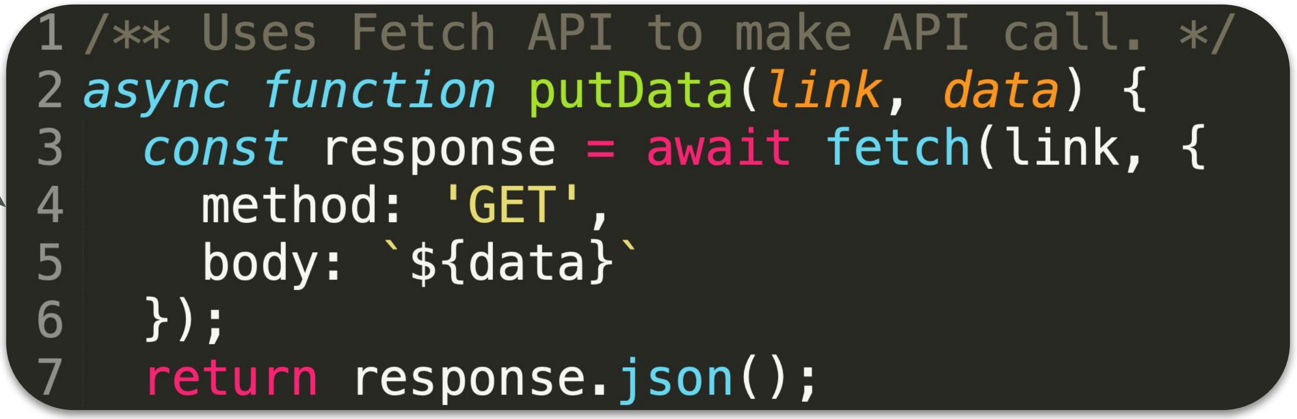
Method

When making a call, you must specify a method. They include:

- GET: used when you are only retrieving information
- POST: used to add data to an existing collection. For example, you would post a row to a database, or a file to a directory.
- PUT: used to update existing information or create a new resource if one doesn't already exist
- DELETE: deletes a resource.
- PATCH: makes a partial update on a resource.

Body

HTTP requests often require a body, which is set of key-value pairs, much like dictionaries in most languages. The body usually contains any information you are sending to the service.

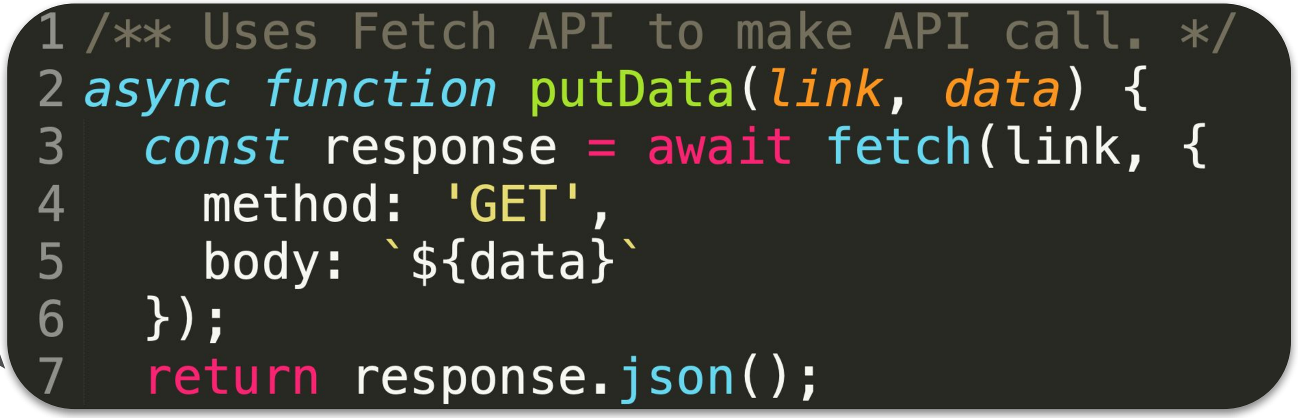


```javascript
/** Uses Fetch API to make API call. */
async function putData(link, data) {
  const response = await fetch(link, {
    method: 'GET',
    body: `${data}`
  });
  return response.json();
```

Calling The API (JavaScript)

Response

When you use an API, you are really calling a function that the provider created. Hence, all HTTPS requests will have a response. The response will be a JSON string, so you will have to parse the string version of a JavaScript object into an object of your choosing. Some frameworks (including Javascript) allow make it easy to parse the response, but otherwise you may have to download a module or parse it manually.



```
/** Uses Fetch API to make API call. */
async function putData(link, data) {
  const response = await fetch(link, {
    method: 'GET',
    body: `${data}`
  });
  return response.json();
```

Exercise

Making an HTTPS request is similar to visiting a url. Let’s see how APIs work by analyzing the results of a call to the Alpha Vantage stock market API. To visualize the results better, go to JSON Editor.

Visit the following url, which you would put in your HTTP request. Also take a look at the link itself. Can you identify portions of the API call?

https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=IBM&apikey=demo

Using SDKs

Find Your SDK

When you are given your task to build an application, you should check to see if the platform or framework you are using has an SDK. It probably will.

Read the Documentation

As with most software, you should start by familiarizing yourself with the documentation, which will share the best practices and provide information on the functions and features you can use.

Integrate the SDK

Once you are familiar with the documentation, you should install the necessary modules and get to work. Refer to the documentation often to make sure you are taking advantage of the SDK.

Useful Resources

APIs:

- Stock Market: https://www.alphavantage.co/
- Weather: https://developer.accuweather.com/
- Top 50 Most Popular APIs: https://rapidapi.com/blog/most-popular-api/

SDKs:

- Firebase: great for databases, user authentication, and other services. https://firebase.google.com/
- Android: https://developer.android.com/studio/?gclid=EAIaIQobChMI1KDut5bu6gIVURh9Ch1BPwKLEAAYAyAAEgKervD_BwE&gclsrc=aw.ds
- iOS: https://developer.apple.com/ios/