Application Programming Interface (API)

What's an API?

An application programming interface (API) acts as the connection between an application and some complexity which the application would like to leverage. The API can be compared to a waiter in a restaurant. Everything that happens in the kitchen has a certain level of complexity, but those dining at the restaurant do not need to be exposed to this complexity. By interacting with the waiter, people are able to dine without worrying about what happens behind the scenes in the kitchen. APIs can be used to provide access to external data, hide complexity, extend functionality, and increase security. This chapter will focus on access to external data. Furthermore, rather than discussing the creation of an API, this chapter will focus on how programmers interact with an existent API.

Interacting with Data

Applications often need to access data that is stored in databases. An API will often be used to facilitate access to this external data or internal data.

API URL

Similar to a webpage, an API becomes accessible through a URL. The URL allows you to access the API and use it's various features. The basic structure of the API URL is shown below.

```
// API URL Formatting
scheme://host/basePath

// API URL Example
https://example.api.com/api
```

Scheme

A scheme is the transfer protocol that is used by the API. While you may not be familiar with the term of a scheme, you may be familiar with scheme examples that are used in website protocols, such as http and https.

Host

The host is the domain name or IP address of the host that serves the API.

Base Path

The URL prefix is the start of the path for all the endpoints of the API.

API Endpoints

Let us return to the restaurant example from before. Instead of focusing on the comparison between an API and a server, we will consider what data would be accessible in an API representation of a restaurant. Let's say the restaurant stores information about the food, the staff, the customers, and the table availability. An API could be configured to access this information from a backend database, but we need a way to specify which resource we would like to access. This is where API endpoints come into play. An API endpoint is where requests to the API are received for a specific resource, and it can be specified through the API URL. Endpoints give us the ability to specify which resource we would like to access.

Below is an example of the URL that could be used to access and interact with data in the "dishes" endpoint from the restaurant example.

```
https://restaurant.com/api/dishes
```

API Requests

Now that we have API endpoints, we need a way of interacting with them. This is where API requests become relevant. Using a given endpoint, a client interacts with the data by sending a request to the API and receiving a response. When discussing API requests we will be referring to those involved in a REST API, which is an API that uses HTTP methods. The common HTTP methods that will be discussed in the following section are GET, POST, PUT, and DELETE.

GET

The GET method is used to receive data for the specified resource. The GET request can either be used to request a list of resources or a single resource.

```
// URL for requesting a list of the dishes available at the restaurant
GET https://restaurant.com/api/dishes

// URL for requesting a single dish by the dish ID
GET https://restaurant.com/api/dishes/{dishId}
```

POST

The POST method is used to create a new resource. The object that is being created will be specified in the request body of the HTTP request, which will be discussed in the next section.

```
// URL for a POST of a new dish
POST https://restaurant.com/api/dishes
```

PUT

The PUT method is used to update a current resource specified by the ID in the URL.

```
// URL for a PUT of an existing dish
PUT https://restaurant.com/api/dishes/{dishId}
```

DELETE

The DELETE method is used to delete a resource specified by the ID in the URL.

```
// URL for a DELETE of an existing dish
DELETE https://restaurant.com/api/dishes/{dishId}
```

API Responses

JavaScript and APIs

Activity

- Use postman to learn how to interact with an API
  - Have certain questions that need to be answered through the use of API requests
- Have everyone create a small web application using data from an API
  - Possible API: Spoonacular API.
  - Alternatively we could build our own

Sources

API basics video: https://www.youtube.com/watch?v=OVvTv9Hy91Q
Google maps API: https://developers.google.com/maps
CRUD: https://www.sumologic.com/glossary/crud/
Spoonacular recipe and food API: https://spoonacular.com/food-api
API guide: https://swagger.io/docs/specification/2-0/api-host-and-base-path/
API endpoints: https://blog.hubspot.com/website/api-endpoint
HTTP Methods (used in API requests):
https://assertible.com/blog/7-http-methods-every-web-developer-should-know-and-how-to-test-them