API beginner course by FreeCodeCamp: <https://www.youtube.com/watch?v=GZvSYJDk-us>

GitHub link: <https://github.com/craigsdennis/intro-to-apis-course>

**Tip**: You can have infinite mail id’s by using name+site@gmail.com

API – Application Programming Interface

**Interface**

Interface is the thing that you interact with and control the way a device works. The process happening inside a device is abstracted by the interface. They are also used in GUIs. Like a play button in a music player. Interfaces abstract the implementation of the process. Like the UI is made for the user, the API is made for the application programmer.

Using API’s gives us many advantages. One of them is computational power. Doing some of the tasks a remote API does, like, searching for a song playing in real time, using google translate to translate something in real time using a camera. To do these things, you will need to have a database of all the songs or words on your device. But, with a remote API, you don’t need to.

URL – Universal Resource Locator, URI is a super set of URLs, I in URI is Identifier.

When clicking a link on your browser, the browser which acts as a client to the server, will send a HTTP URI GET request to the server, for this request, the server will send a response, whose body will be in HTTP(Hyper Text Transfer Protocol).

**REST API**

REST – Representational State Transfer. This type of remote API is one of the many used but, it is the most used one. This course focuses on REST API’s. For an API to be RESTful, these constraints should be followed:

Client-Server Architecture

Statelessness

Layered System

Cacheability

Uniform Design

Code on Demand

**Stateless** means it doesn’t remember anything happening in a session, so if you want to maintain state like login in a session, you must do it in every session individually using headers.

Resource in URL can be used to describe anything. In an example of a bookstore website, a book shows information like its author, which is also a resource. If you click on the author, it will show a collection of books by that author. You may also see a group of reviews from fans, which is also a resource. If you click on a fan, you will see a collection of books and authors they like.

The body of the reply you get when you send a http request is in JSON, which is JavaScript Object Notation. It provides a way to structure and nest your data. Every programming language will have ways to turn a JSON string into a native object. You can also specify a Content Type in your request.

HTTP Verbs used in REST APIs: CRUD

GET Read

POST Create

PUT Update

PATCH

DELETE Delete

This course uses Spotify and Twilio APIs. Using a Twilio API, you can send a text message from a number to your registered phone number (one of many things it can do).

This course is also using Postman, which will also be useful for the APISEC course.

Now, you can create a folder to save your API requests, you should get the request URL from the CURL, or from the API. Then for the authentication part, create variables for the username and password and use them to avoid sharing sensitive information with your team.

A RESTful API should support **caching**. If you have any file which is consuming more storage, you should be careful to conserve resources by not calling that file again and again. So, you can just store the file and check if it is modified. If it is, we should call for that file, if it isn’t, we can just use the stored file. A **layered system** is when an API calls another API.

The next part was using JavaScript and Python to write a code which can print the latest message sent or send a new message. This is done by downloading helper libraries of an API in your preferred language. These helper libraries are also called as SDK (Software Development Kit). They help in maintaining **uniform design** of a function like creating or deleting message in any language.

Stopped and 2:07:41