

# Overview

Hey, Thank you very much for attempting our code test. We appreciate the time and effort you are going to put into it. Hopefully you'll find it interesting. Let's go into it.

Below is a series of coding tasks which require you to write a RESTful API to perform storage, deletion and search functionality.

Use any framework you feel comfortable with to provide a small project package for your solution. Please include a readme file containing instructions for us to run your code!

It is encouraged to use version control and commit as you go.

There is no set time limit for this exercise, we understand not everyone is familiar with the YouTube API and this is not a typing speed test. Having said that we recommend to try not spend more than 2 hours on it.

## Tasks

1a. Expose an API endpoint, that when hit will store youtube video **titles** and **publishedAt** dates for the *GlobalCyclingNetwork* and *globalmtb* YouTube channels to a database[1]. We only want to store videos that match the filter criteria[2].

The above must be done using the Youtube data API v3. An API key has been provided for you to use:

**AlzaSyAsmKbjsSAXARfIZ9XO0RmvU4iLMnU3dCc**

[1]**Storage** of information must use the provided SQL database structure (**youtube.sql**).

[2]We are only interested in storing videos with titles that match the filter criteria. The criteria **must** be read from the provided file (**search\_filter**). Filtering should be case insensitive. How you implement this filter is up to you.

- 1b. Expose an API endpoint that will fetch the results from 1a.
2. Expose an API endpoint which queries the searched data from 1 by ID.
3. Expose an API endpoint which removes an ID from the data store.
4. Expose an API endpoint that, when provided with a search term, returns a list of matching results for given search term from the information stored in task 1. The item(s) returned should contain the title and id only

# Rules

- You must use the database file provided to store data.
- You **may** modify the tables constraints but **not** the fields. All the appropriate fields are included, so you should not need to add or remove any.
- You must use validation where appropriate.

# Deliverables

- Project containing the source code for the above solution.
- We're looking for a REST API that can be called from the browser or other tool (such as postman) and return the expected data in JSON form.
- Any instructions required to get the project running.

Have fun and good luck!