

Goal

Implementing a small web-server that gives information on daily COVID-19 confirmed cases.
Write it in any language you choose.

Part I - Build web-server

Build a web-server using the API described in <https://github.com/M-Media-Group/Covid-19-API>.
Use the routes: /cases & /history.

Requirements:

- Implement a web-server in any language (preferred JS/TS and node.JS / Java).
- The server will run locally (choose any port).
- The server will support the following APIs:
 - o Daily new confirmed cases in a given country
 - ♣ Input
 - ♣ Parameter country name `string`, example `Israel`.
 - ♣ Parameter date in format `<day>-<month>-<year>` | `\d{2}-\d{2}-\d{4}`, example `13-02-2021`.
 - ♣ Output
 - ♣ Response needs to be numeric value `[0 - inf]` for example `50`.
 - ♣ example:

Input: Germany, 13-05-2021

Data:

Germany, confirmed case 13-05-2021: 3578125, 12-05-2021: 3564494

Output: 13631

- o Return information about COVID deaths and confirmed cases.
The user can add and remove countries, get the COVID information based on his countries and check the current countries list.
 - ♣ To get started the user must login - in order to initialize the list.
 - ♣ The user doesn't need a password or any key just a username.
 - ♣ The client will add the username to each query (body, route or query params).
 - ♣ If an unidentified user try to call any of the APIs an appropriate error code will return.
 - ♣ A username is mandatory only for this section. the previous API (confirmed cases by country and date) doesn't need a username.
 - ♣ APIs:
 - ♣ Add user.
 - ♣ Input
 - ♣ Parameter user name `string`, example `Methuselah`.
 - ♣ Add country.
 - ♣ Input
 - ♣ Parameter country name `string`, example `Israel`.
 - ♣ Remove country.
 - ♣ Input
 - ♣ Parameter country name `string`, example `Israel`.
 - ♣ Get the list of countries.
 - ♣ Output
 - ♣ List of Countries `string[]`, example `Israel, Jamaica`.
 - ♣ Get the number of deaths by country in date range.

Input

- ♣ Parameter from, format <day>-<month>-<year> | \d{2}-\d{2}-\d{4}, example 26-02-2021.
- ♣ Parameter to, format <day>-<month>-<year> | \d{2}-\d{2}-\d{4}, example 27-02-2021.

♣ Output

- ♣ For every country the number of deaths on every day from start date to end date.

- ♣ Get the number of confirmed cases by country in date range.

♣ Input

- ♣ Parameter from, format <day>-<month>-<year> | \d{2}-\d{2}-\d{4}, example 26-02-2021.
- ♣ Parameter to, format <day>-<month>-<year> | \d{2}-\d{2}-\d{4}, example 27-02-2021.

♣ Output

- ♣ For every country the number of confirmed cases on every day from start date to end date.

- ♣ Get the country (from the list) with highest deaths cases relative to the country population.

♣ Input

- ♣ Parameter from, format <day>-<month>-<year> | \d{2}-\d{2}-\d{4}, example 26-02-2021.
- ♣ Parameter to, format <day>-<month>-<year> | \d{2}-\d{2}-\d{4}, example 27-02-2021.

♣ Output

- ♣ For every date the country with the highest ratio (|death cases| / |population|).

- ♣ Get the country (from the list) with highest confirmed cases relative to the country population.

♣ Input

- ♣ Parameter from, format <day>-<month>-<year> | \d{2}-\d{2}-\d{4}, example 26-02-2021.
- ♣ Parameter to, format <day>-<month>-<year> | \d{2}-\d{2}-\d{4}, example 27-02-2021.

♣ Output

- ♣ For every date the country with the highest ratio (|confirmed cases| / |population|).

♣ example:

- Add country (Germany).
- Add country (France).
- Get the list of countries () -> [Germany, France]
- Remove country (France)
- Get the list of countries () -> [Germany]
- Add country (Israel).
- Get the list of countries () -> [Germany, Israel]
- Get the number of deaths (26-02-2021, 27-02-2021) ->
[{"country": "Germany", "date": "26-02-2021", "deaths": 60}, {"country": "Germany", "date": "27-02-2021", "deaths": 153}, {"country": "Israel", "date": "26-02-2021", "deaths": 35}, {"country": "Israel", "date": "27-02-2021", "deaths": 20}]
- Get the country with the highest confirmed cases ratio (26-02-2021, 27-02-2021) ->
{"date": "26-02-2021", "country": "Israel"}, {"date": "27-02-2021", "country": "Israel"}

**If there is not enough information - missing date or missing country - just return the correct http error code.*

Part II - Dockerize

In this section you will dockerize your web-server.
Write a DockerFile, build it and run it!

Requirements:

- Dockerize the web-server (base on node.JS 14 and up for JS or java8 and up for java, other languages feel free to pick any base image).
- Run the dockerize web-server on port *8080*.
- Enable access to the web-server on *localhost:8080 (127.0.0.1:8080)*.

**if there is any problem with the Docker - not running, error messages...
Just contact us don't wait your time with it.*

Write a clean code and useful comments,
add documentation about the server (API - routes) and how to build and run your server.

Attach your code (ReadMe, source files) and documentation - upload it to google drive (or any similar service) and share the links with Gil.Peleg@elbitsystems.com and Shay.Gozansky@elbitsystems.com

Useful links:

- Docker windows install: <https://docs.docker.com/desktop/windows/install/>
- Docker Ubuntu install: <https://docs.docker.com/engine/install/ubuntu/>

Feel free to ask us any question during the task.