Hello, World!

Hey! We're super glad you made it to here. We wrote this problem for you to solve, so we can see how you think, how you approach solving problems, and how you write code.

The Task

The task you need to solve is made out of the following steps:

- Data extraction, preparation and loading (The ETL part)
- Creating a RESTfull API service (The API part)

The ETL Part

We have an API Service in place that serves user data. We would like to include that data into our app, but the problem is that there is a lot of user information that we don't need. Your job is to clean the user data, and save the cleaned information in our database. The current user model looks like this:


```
{
  "results": [
    {
      "gender": "female",
      "name": {
        "title": "Mrs",
        "first": "Ying",
        "last": "Akker"
      },
      "location": {
        "street": {
          "number": 2111,
          "name": "Kruissingel"
        },
        "city": "Gasselter-Nijveensche-Mond",
        "state": "Zeeland",
        "country": "Netherlands",
        "postcode": 73888,
        "coordinates": {
          "latitude": "46.9212",
          "longitude": "7.7018"
        },
        "timezone": {
          "offset": "+10:00",
          "description": "Eastern Australia, Guam, Vladivostok"
        }
      },
      "email": "ying.akker@example.com",
      "login": {
        "uuid": "f06f3f4d-ba06-4ef2-8100-3cc8632a062d",
        "username": "angryduck799",
        "password": "123123123",
```

```
        "salt": "Qay39kUg",
        "md5": "faade05e6cdbd4eee2976cea8aff0f20",
        "sha1": "3fce49045cad1390bf4d959fcf28a6713c9530ff",
        "sha256": "fade1e3736d1e1dbae2b5ba3d563e39cdc8b914869f42e38ec51ee5fea3ad522"
      },
      "dob": {
        "date": "1987-04-09T14:26:34.774Z",
        "age": 34
      },
      "registered": {
        "date": "2011-02-17T04:32:44.071Z",
        "age": 10
      },
      "phone": "(216)-041-8446",
      "cell": "(937)-168-2556",
      "id": {
        "name": "BSN",
        "value": "51089883"
      },
      "picture": {
        "large": "https://randomuser.me/api/portraits/women/39.jpg",
        "medium": "https://randomuser.me/api/portraits/med/women/39.jpg",
        "thumbnail": "https://randomuser.me/api/portraits/thumb/women/39.jpg"
      },
      "nat": "NL"
    }
  ],
  "info": {
    "seed": "f7d41cca99b70169",
    "results": 1,
    "page": 1,
    "version": "1.3"
  }
}
```


We need it to be in this format:


```
{
"id": 1,
"gender": "female",
"first_name": "Ying",
"last_name": "Akker",
"email": "ying.akker@example.com",
"username": "angryduck799",
"password": "123123123",
"birthday": "1987-04-09",
"nationality": "Netherlands"
"cell": "(937)-168-2556",
"location": {
             "street": "Kruissingel",
             "street_number": 2111,
             "city": "Gasselter-Nijveensche-Mond",
             "state": "Zeeland",
             "country": "Netherlands",
             "postcode": 73888
  }
}
```

Each time you hit the API, you will get 1 user. We need 100 users cleaned and saved. Please setup a database locally on your computer where you will store data (our recommendation is to be Elasticsearch or PostgresSQL). Pay attention to "nationality" field, should be full name of nationality (not abbreviation). Feel free to find public source with this info, and to merge it with other data. The solution should be implemented in Python, but there are no limitations to the approach you want to use.

The API Part

The second part of your task is to create a RESTfull backend service, that will expose the user data via API endpoints. The service needs to have the following functionalities:

- Get all users - Returns the list of all users in the database
- Get single user - Returns user that has matching ID you provided
- Create a user - Insert new user into database
- Update user - Update info about existing user
- Delete user - Remove user with matching ID from the database

You need to implement this using the **FastAPI** framework. FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.6+ based on standard Python type hints.

Documentation: https://fastapi.tiangolo.com

Bonus Task

We will consider it a bonus if you Dockerize your project, by creating Dockerfile and docker-compose for both ETL and API modules.

Good Luck ☺

ps. If you have any questions or get stuck, please feel free to ask.