**NODEJS + SQL Exercises - Part III**



We are going to create a small RESTful API using Node + Express + a PostgreSQL database that will serve the client with some data about users and their orders.

Work with Git:

**1 First Commit +** **1 Commit per addition of a request endpoint**

1. Create a Github Repo for this exercise
2. Create a database instance on [ElephantSQL](https://www.elephantsql.com/)
3. Write the code to create the database. You need two tables: one for the users, and one for the orders, with the following properties:
   1. users
      1. id
      2. first\_name
      3. last\_name
      4. age
      5. active
   2. orders
      1. id
      2. price
      3. date
      4. user\_id

You can use the ElephantSQL integrated SQL terminal to run your table creation code.

1. Create an Express server with separate routes for:
   1. The users:

* GET / : To get all the users
* GET /:id : To get one user (through its id)
* POST / -> To create a new user
* PUT /:id : To edit one user (with the id)
* DELETE /:id : To delete one user (with the id)
  1. The orders:
* GET / : To get all the orders
* GET /:id : To get one order (through its id)
* POST / -> To create a new order
* PUT /:id : To edit one order (with the id)
* DELETE /:id : To delete one order (with the id)

EXTRAS:

- Validate all the data coming from the users/orders for the Post/Put routes ([help](https://express-validator.github.io/docs/index.html))

- Create a separate module for your pool object ([help](https://node-postgres.com/guides/async-express))

- Separate your routes into 2 router files. One for Users, and one for Orders ([help](https://expressjs.com/en/guide/routing.html))

EXTRAS 2: Create the following 2 routes:

- Create a user route that will return all the orders of a specific user

**GET /:id/orders** : To get all orders linked to a specific user

- Create another user route that will set a user as inactive if they have never ordered anything:

**PUT /:id/check-inactive** : If a user has never ordered, he should be set as inactive.

Don’t forget to check the documentation here: [https://node-postgres.com](https://node-postgres.com/features/connecting#Environment%20variables)