Core | Deadline: Sunday of Week 5 | Difficulty Level: Basic | Est. Time: 01:00-02:00

Faker API (Core)

Respond with manipulated data from the Faker API using JavaScript!

Learning Objectives

- Create RESTful routes in Express

- Respond with manipulated data from the Faker API.

Welcome to another **Core assignment**! Some students like to explore the assignments before they're finished reading through the lessons, and that's okay! It can be good for your brain to have a preview of what your future challenges might be. However, before you begin this assignment, it's important that you've first:

- Completed the preceding lesson modules

- Taken the knowledge checks to confirm your understanding

- Viewed lecture material related to the assignment topics

- Completed and submitted your practice assignments

Faker API:

In this assignment, you will use two Javascript libraries to create a simple API that generates *User* and *Company* objects depending on the requested route. The libraries you will use are **Express** for the server and **Faker** for the random data generation. The goal is to start broadening your understanding of APIs.

For this assignment, you are tasked with creating *User* and *Company* objects

- User Object
    - password
    - email
    - phoneNumber
    - lastName
    - firstName
    - _id

- Company Object
    - _id
    - name
    - address
        - street
        - city
        - state
        - zipCode
        - country

Faker

The faker library has many methods you can use to populate specific fields with randomly generated data.

To add faker to your project you need to install it using npm.

```
npm install @faker-js/faker
```

[You can find faker's API documentation for some basic examples of the library in action.](https://login.codingdojo.com/m/750/16737/124720)

Hint: You'll want to check out the location section.

Here is an example of how you can use faker

```
// we can create a function to return a random / fake "Product"
import {faker} from "@faker-js/faker";
const createProduct = () => {
    const newFake = {
        name: faker.commerce.productName(),
        price: "$" + faker.commerce.price(),
        department: faker.commerce.department()
    };
    return newFake;
};

const newFakeProduct = createProduct();
console.log(newFakeProduct);
/*
 * The output of the above console log will look like this
 * {
 *   name: 'Anime Figure',
 *   price: '$568.00
 *   department: 'Tools'
 * }
 */
```

☐ Create a project folder "Faker_API" with a "server" folder inside

☐ Create a server.js file

☐ Navigate into the server folder using the terminal

☐ Generate package.json

☐ Configure the package.json to recognize modern JS syntax and start using nodemon

☐ Install express, nodemon and faker using npm and import them into the server.js

☐ Create 2 functions: createUser, createCompany that return an object with the properties listed above

☐ Create an api route "/api/users/new" that returns a new user

☐ Create an api route "/api/companies/new" that returns a new company

☐ Create an api route "/api/user/company" that returns both a new user and a new company

☐ Run the server and test the routes with Postman