

Backend Engineer - Python

Home Assignment

In this exercise you will be required to design and implement several back-end components to an existing application.

Pre-requisites

- 1) Docker (and docker-compose) - <https://docs.docker.com/get-docker/>
- 2) Python 3.10.0 (can be installed from [Python.org](https://python.org))
- 3) Node.js LTS version 64-bit (can be installed from [the Node.js website](https://nodejs.org))

Background

In this task you will receive a ready-made web application, which simulates a situation where you have arrived at a new workplace and you need to find your way around and integrate within an existing project.

We generated a web application called “Fred” (Based on Flask and React) and pushed its source code to GitHub. Please find the Github repository link in the instructions email.

We have two main entities in our application: “Author” and “Quote”.

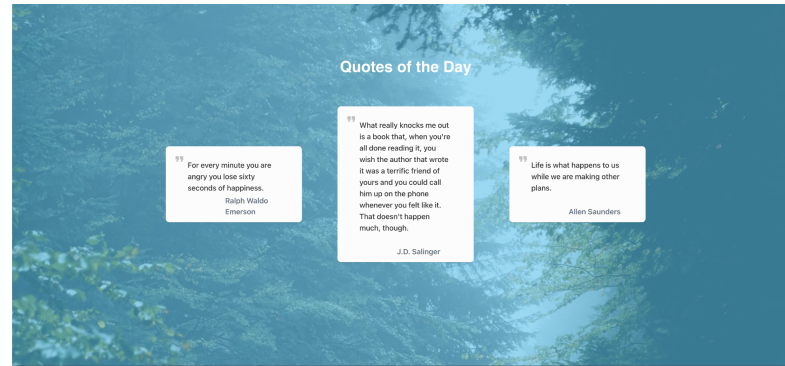
The Task

- 1) Make sure you know how to run the application and populate the DB (as specified in the README.md). Note that it might not run smoothly and you need to find a way to workaround those issues.

Why is this part of the exercise?

It's important for us to check those capabilities as well, which are not only how good your code is, but also how you deal with 'making stuff work'

- 2) In the main page you can scroll down and find the random quotes of the day (screenshot below), please change the existing relevant endpoint so it can get the amount of random quotes to generate as a parameter (between 1-10) and return the list of them (don't mind about the UI not supporting a bigger number of quotes).
Commit and push everything to GitHub.



- 3) Add a new API endpoint called “/authors” that returns a list of all authors in the DB (Name, Id, Date of birth) that were born after 1960.
Commit and push everything to GitHub.
- 4) Add a new API call that will return the author's book list.
Given an author, the API should return the list of the author's books (Note: The API provides the reviews, you'll have to extract the relevant values out of it) using a 3rd party API of <https://developer.nytimes.com/docs/books-product/1/overview> - “Books API”.
You can test your code in any preferred way - UI/Tests/Freestyle/Postman/etc...

Please design your code in a way which allows replacing the 3rd party API with other providers in an easy way (e.g tomorrow your teammate will have to add another API).

Commit and push everything to GitHub.

You can use the following API keys or create your own:
API Key - heAhyfisnHARIWMVPIHgSgzpr8UhsHXZ

Notes

We'll ask a few questions regarding the decisions you've made along the way.

- Aim to spend no more than approximately 3 hours on this task.
- Feel free to cut corners, but make a note where you do, and be prepared to explain what you would do in a production context.
- Be prepared to execute your code and discuss the rationale behind your choices,
- Please write down at the end of each part how long it took you to complete the task.

For any question/clarification, feel free to reach out to my mail (ido@dig.security) or mobile.
Good luck!