liblab

See FAQ

Hey James Garza, we're excited for you to interview with LibLab!

This challenge is a chance to show your programming skills and understand the kind of problems you'd be working on day-to-day.

LibLab Take-Home Project (Sr.)

Do you love The Lord of the Rings? Great, how about making it easy for developers to consume information about the trilogy? In this project, you will be building an SDK for part of an existing Lord of the Rings API.

Time requirement: Please spend as much time as you feel is necessary to complete the task. We ask you to submit the project code within 72 hours of starting.

Start Assessment

Lord of the Rings SDK

Repository https://github.com/hatch...

In this assessment, you will be submitting your solution using `git`. To do so, you will need to clone the repo linked above, and push all your changes to the `dev` branch (which is also the default branch). You will be submitting this assessment by creating a pull request between the `dev` and the `main` branch.

If you need help with how to clone the repo to work on the project locally, see this **link for help**.

Overview

In this project, you will be building an SDK for an existing Lord of the Rings API.

First please explore the API and the different endpoints.

Goal: write an SDK in your favorite language that makes this API accessible to other developers. The SDK only needs to cover the **movie and quote endpoints:**

- /movie
- /movie/{id}
- /movie/{id}/quote
- /quote
- /quote/{id}

Time requirement: Please spend as much time as you feel is necessary to complete the task. We ask you to submit the project code within 72 hours of starting.

Points to Consider

- Pay attention to code quality and readability.
- Include filtering in the SDK.
- Include a testing case suite. Keep in mind we will be looking at the solutions architecture, testing, and documentation.
- Although you're not implementing all the endpoints, write the SDK as if you were. Keep maintainability and extensibility in mind.
- Modify the readme file for your SDK users specifying how to use and test the SDK.
- The SDK does not have to mirror the API. You can add abstractions and/or combine different calls.
- The SDK should be prepared for production; However, do not share your solution anywhere online or publish the SDK.
- Include a demonstration file(s) that can be used to test the SDK locally, along with instructions on how to run it.
- Modify the `design.md` with information about your SDK design.

Do not use any automation or code generation tools such as OpenAPI Generator. Please note that any submission created using such a tool will be disqualified.

Submitting Your Solution

Once you have finished writing your solution or you have reached the time limit, open a terminal window and navigate to the root directory of your project. Commit all your changes and push them to GitHub.

Finally, open a browser and open a pull request from `dev` into `main`. The repository has a pull request description template - please answer all of the provided questions when making your pull request. Do **not** merge the pull request. Follow the steps below to confirm your submission!

Submitting Your Solution

Open a pull request from `dev` into `main` once you are ready to submit. Do **not** merge the pull request. Here is a step-by-step guide on how to create a pull request.

When you have completed this task, click Ready To Submit. READY TO SUBMIT

Powered by hatchways