UserZoom

Fullstack TS Test

Moderated Engineering Team

2021.02.11

UserZoom | Moderated

Summary

1. Introduction
2. Your Background
3. Github Activity Test

UserZoom

Introduction

Introduction

We want to know your approach to modern Software Engineering.

The goal of this exercise is to trigger discussion on some of the aspects related to the role.

Please note there is no right or wrong answer, just different ways of focusing the problems and build a suitable solution to them.

UserZoom

Your Background

UserZoom

Your Background

Are you passionate about Cloud Distributed Architectures and being a well-crafted Software Engineer? Please share with us some of this:

- Your online references. Sites that you come back to, time and time again, for inspiration or guidelines.
- The last 3-5 articles/decks/videos/tweets/podcasts that have managed to caught your attention, and a short explanation on why they impressed you.
- Any public resources (code, articles, open-source products, etc) you may have contributed to, if any.

UserZoom

Github Activity Test

Github Activity Tracking

Summary

We are monitoring Github repos and would like to see activity in a beautiful web application.

- The application should be a SPA so that the navigation is done always client-side, without completely refreshing the whole document.
- The application should consist of two views.
    - The first view should list the repos of a certain organization and (optional) be able to filter the repos.
    - The second view should show the details of a certain repo.

Don't worry if the website isn't complete, but it's always a bonus!

Github Activity Tracking

Guidelines

- You can use Typescript (recommended :)) or Javascript for your solution
- You must use React for the frontend.
- You must use NodeJS for the backend.
- Provide a clean design.
- Keep in mind good design considerations applicable to the problem such as extensibility, maintainability, and modularity, among others.
- You can create the application from an existing boilerplate, if you wish.
- Use of third party libraries is allowed.
- The application will be reviewed in the latest version of Google Chrome for desktop (don't worry about other browsers).
- Create a dedicated git repo for this and send us the link.
- Communicate to the customer the implemented solution. To do so, please create a README.md (Markdown) on the root directory explaining your solution to the client.
- (bonus) Dockerize your application
- Don't put more than 3-5h into this test, please. You have to live too!

Github Activity Tracking

Frontend - List view

- Provide a way to filter by Github organization, so the end user can start the interaction by setting an organization to retrieve the list of the repos via the API described at https://developer.github.com/v3/
- Sort the repos list by date of activity (most recent first).
- Display some relevant information about the repo.
- When a repo is clicked, the SPA have to navigate to the Detail view.
- Bonus
    - Add a “previous” and “next” buttons to paginate the results. Alternatively, use infinite scroll for the “next” function and omit the “previous”.
    - Add a filter by repo name, so the user can easily narrow down the results in the retrieved list.
    - No extra API call is required in order to filter the list of repos, the filter will only take into account the already downloaded list.
    - Filtering must have an immediate effect, so the items on the list will change as the user types characters on the filter.

Github Activity Tracking

Frontend - Detail view

- This view should display the details of the selected repo, according to the information available via the API described at https://developer.github.com/v3/
- Show some meta information for the repo and the owner avatar, if it exists.
- Include a commit-activity overview section, where the avatars for the most active authors are displayed with some visual indicator of their relevance in terms of commit contribution

Github Activity Tracking

Backend

Create an REStful API where the repo activity details can be submitted.

Frontend interaction:

- Add a “Track repo” button to the details view.

The backend API should have two endpoints:

- Submit a repo that we want to track. The following data should be persisted in a database:
  - meta-information of the repo (for future periodical retrieval of activity).
  - the activity collected at the frontend at this point in time (for tracking purposes).
- Retrieve the list of tracked repos. O meta-information of the repos and date of last tracked activity (not to be used by the frontend yet).

UserZoom

Thank you and good luck!

userzoom.com

UserZoom