

Full Stack Home Assignment

Introduction

1. Use Node for back and prefer using React for front
2. Aim to spend no more than approximately 4 hours on this task.
3. You may use any 3rd party SDK or library. In the documentation, describe what is the benefit in using it and be prepared to discuss the rationale behind your choices.
4. We are looking for a well-designed, documented code which demonstrates React and Node best practices

Guidelines

1. Note that only a working application will be considered as success.
2. Read the scope carefully and make sure you understand the implicit requirements
3. No UI design is given by purpose. The use of tools like bootstrap is recommended (but not mandatory).
4. 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
5. Try to implement as many as the time permits. We prefer a working, well design solution with less functionality or external libraries.

Front Task

You need to develop a React app that displays random people and their details.

- Use the API from <https://randomuser.me/> to get a list of 10 people.
- Use State management tool of your choice (Redux, MobX, Apollo Client etc.)

Screen 0

This screen will have only two buttons:

- “Fetch” button to get the data from random user API and will open Screen 1
- “History” button to get saved users from backend and will open Screen 2

Screen 1

This is a list-based screen, each person will have its own row

- Each row will include:
 - o Thumbnail of the profile picture
 - o Name: title+first+last
 - o Gender
 - o Country
 - o Phone number
 - o Email
- There will be an option to filter by name and country
- Clicking on a row will open full profile screen (Screen 3)

Screen 2

Identical to screen 1 but displays the saved profiles from the server

Screen 3

This is will display more information for a profile. It can be opened from Screen 1 or Screen 2

- Large image of the person
- All the data laid as info form
 - o Gender
 - o Editable Name (see below)
 - o Age + year of birth
 - o Address:
 - Street number + name
 - City

- State
- o Contact:
 - Email
 - Phone
 - Four buttons:
 - o “Save” - If the profile is NOT saved in the backend (the previous screen is 1, from random user API), “Save” button will send the profile info to the backend where it will be kept in a DB
 - o “Delete” - If the profile is saved in the DB (the previous screen is 2, from our server), “Delete” button will remove it from the DB
 - o “Update” - Name field is editable. The user will be able to modify the name of the person and click on update button
 - If this is a DB saved profile, the server will be updated
 - If the profile is not saved, it will be modified in Screen 1 (list)
 - o “Back” - navigates back to the previous screen

Back Task

- Node server implemented in TypeScript
- Any database or other persistent tool
- We are not expecting you to build a full-blown API. Do the needed minimum
- No need for authentication

Submission

- If you have a public Git account, deploy to Git and send as a link. There should be clear distinction between the Client project and the Server project (two repositories, to folders etc.)
- You can also zip the sources (without node-modules) and send it
- In both cases, add clear build, prerequisite and run instructions
- Deployment to a AWS (or other public cloud) is a big advantage but not mandatory