

Submitted by Bhavik Sheth

## Architecture

Create React App: Extremely straight forward to start a project in comparison to creating a boilerplate manually. Chose not to eject from CRA because I was satisfied with the build tool and configuration choices provided

Layout: I chose this folder structure and layout because it separates all the concerns into individual components and keeps the application modular, maintainable, and scalable

Design: Due to the little content, it was difficult to fill the page as much as possible and make it look full

CSS Modules: this prevents any conflicts with class names or ids and overriding of styles

Redux: was not required in a project of this size but used it to show I am comfortable with it

Redux Sagas: not required as Redux Thunk would have been fine but I like Sagas and wanted to show I can use them

## Improvements I would consider making

Create a "common" folder which would contain reusable components like text input, button, header but haven't done so due to time constraints

Validate the date either by using moments.js and text validation or using the date format required by the api or using a date picker component depending on the business requirements

Make the currency text input a dropdown instead possibly by getting the list of supported currencies in an api request if the api supports this or storing a json file in the project if the currencies change extremely rarely

Would use prettier and eslint with airbnb styling guide to keep code consistency in code base

Use Reselect if the Redux state becomes nested and complex

Put api keys into data bags respective to the environment for example, development, testing, production, etc. maybe using the 'config' package. This improvement would be for security reasons

Implement appropriate tabbing functionality for the inputs and button

Use PropTypes to check that the type of the Prop is correct

Coverage tests to make sure I have tested all the paths and lines in the code base

## Challenges I faced and how I overcame them

Most challenges were due to time constraints as there is much to do and expectations to meet to successfully pass the test so used my experience to select and invest time in the most important tasks. This included:

- Unit testing: this is a big must on every project
- Prototypes: mandatory on every project but I chose to skip this as it did not show much value from my side
- Styling: tried to use colours and good layout to make the page look full
- API: Had a small problem with the API as 'Fixer historical with base currency' was behind a subscription plan so I found another one online which did the job
- Thinking from the perspective of the client and satisfying as many requirements as possible (and hopefully be impressed by it)