

miracl Challenge - Senior Frontend Developer

Disclaimer: while this is a real task the miracl team is going to work on soon, we are not going to use any of your work in our product. That would be illegal. This challenge only serves as a way to demonstrate your skills during the hiring process.

A lot of the innovation at miracl is happening behind the scenes - invisible to customers. One such thing is the assisted management of sensitive documents sent in by customers. However, some customers accidentally upload documents without naming them properly so they can't be processed automatically. Therefore it is up to our financing specialists to look at each of these documents and assign the correct type manually, e.g. passport, salary sheet, etc.

Our new backend developer did a great job building an API that makes this process a breeze:

1. List all documents for a given customer ID
2. Download documents by their IDs
3. List all available types for documents
4. Update document type by their IDs

Now she needs a frontend to make the functionality accessible to financing specialists. Can you help her?

Here are the requirements for the implementation you're going to work on:

- It should be a single page application, to keep it simple and clean
- Initially when the application is first opened, the document list should be empty, and the user should be able to "fetch" customer documents by providing the customer ID
- Pagination or lazy loading of documents is preferred (chunk size for one page is up to you)
- The user should be able to filter out the documents by document type
- The user should be able to filter out the documents by name
- The user should be able to update the document type
- The user should be able to "download" some specific document

Some other things to take into consideration:

- Since it's a web first application, you don't have to think about it being mobile responsive
- Since you won't be given the real API to work with, feel free to mock the Document data as you please, that also goes for the fetching and downloading parts of the API. As for fetching the document list - just create a mock Promise that returns the list of the documents from a mock JSON file by customer ID. And as for simulating the download process, just create a mock Promise that returns a boolean value of true after a random amount of time between 1 and 2 seconds, signalling that the download was successful.

As for the implementation, UI and UX, it's totally up to you how you'll approach the solution, but here are some guidelines:

- React + TypeScript is preferred as that's what we use on a daily basis
- As for styling, CSS, SCSS, styled components, or event component libraries like MUI are all allowed, feel free to choose whatever you prefer
- As for the application state management, using React Context API along with the hooks API is preferred
- No need for any kind of tests for this assignment

After you're done, please upload your code to a private repository on GitHub and share it with thomas.taschauer@miracl.at.