

This assignment is about using the React library to create a small web application. The goal of this assignment is to create a small Blog-like application on which the user can view, add and delete blog post items. This assignment will only deal with the front-end 'view' layer, there is no interaction with any backend API (so no data fetching, all existing data is static) in this assignment. This means that you should create some static mock data yourself to populate your application with.

At Forescout, we use TypeScript in our frontend projects instead of regular JavaScript. It would be a large plus for you to show that you are able to use TypeScript as well for this assignment. However, this is not mandatory.

Requirements

Below are the requirements for the application. This is a

- The main view of the application shows a list of pre-existing blog posts.
- Each blog post has the following properties:
 - Title
 - Date posted
 - Category (one or more categories which can be used to group posts)
 - Content
- In the item list of the main view, the title, date posted, and tags of each blog posts are displayed.
- Clicking on a blog post item in the list will show all the details about that blog post, including the actual content.
- On each item there is a delete button which when clicked will delete the blog post from the list of blog posts.
- New blog posts can be added via a form, which will ask for the Title, Categories and the Content of the blog post.
- You are free to design your own UI, but to get started quickly you could use for example the Bootstrap CSS library.
- Even though this is a small application, it is important that you show that you can think about the structure of the application and are able to break the application down into smaller reusable components.

Optional stretch goals

If you feel more confident in React, you can take inspiration from the below list to implement additional functionalities. All of these are optional, you are not required to implement any or all of this but see it as an opportunity to show your skills. Feel free to come up with additional features, but make sure to document them properly if you do.

- Client-side routing, to be able to share a direct link to a single blog post
- Searching functionality to search the content of blog posts
- Create a view to show posts grouped by categories
- Use an actual live backend instead of static data (for example, <https://jsonplaceholder.typicode.com/>, the 'posts' endpoint)
 - Make sure to consider things like error handling, loading indication etc.
 - Redux could be used to manage the state and deal with asynchronous actions
- Write unit or integration tests for them using the Jest framework (<https://jestjs.io/>)

Getting started

To get started quickly, you can use the create-react-app tool (<https://github.com/facebook/create-react-app>) to quickly setup a working project. This way you do not need to deal with the intricacies of the build process and can focus on the actual application instead.

In case you wish to use TypeScript instead of regular JavaScript, you must run 'create-react-app' differently. Refer to the documentation (<https://facebook.github.io/create-react-app/docs/adding-typescript>) on how to achieve this.

Again, TypeScript is not a must, but showing proficiency in TypeScript is definitely a huge plus.

Delivery

Upon completion of the challenge you can send your results via e-mail, or via a private or public Github repository. If you choose for a public Github repository, please make sure to **not** include any mention to Forescout, FS, or anything else that could be interpreted as mentions to Forescout. Additionally, make sure to include some minimal set of instructions to get your application running.