4.1 Introduction to Node.js

Node.js is an open-source and cross-platform runtime environment that allows you to use JavaScript to develop server-side applications.

Node.js uses the single-threaded, non-blocking, and event-driven execution model, which is similar to the execution model of JavaScript in the web browser.

- Node.js is Single-threaded
- Node.js uses Non-blocking I/O
- Node.js is event-driven

4.2 Introduction to Node.js

Architecture could be as the following:

- Java -> Node JS -> React / Angular / another front-end framework / library
- Node JS -> React / Angular / another front-end framework / library

4.2 Express JS

- Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

- Use the application generator tool, express-generator, to quickly create an application skeleton.

- npx express-generator
- express-generator --view=ejs

4.2.1 Express JS

- Let's add real backend APIs to our Todo APP via Express JS / Node JS

4.3 Introduction to MongoDB

MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

Advantage of NoSQL Database

- Handle large volumes of data at high speed with a scale-out architecture
- Store unstructured, semi-structured, or structured data
- Enable easy updates to schemas and fields
- Be developer-friendly

4.3.1 MongoDB VS Mongoose

In terms of Node.js, mongodb is the native driver for interacting with a mongodb instance and mongoose is an Object modeling tool for MongoDB.

mongoose is built on top of the mongodb driver to provide programmers with a way to model their data.

Mongoose : object data modeling (ODM) library that provides a rigorous modeling environment for your data. Used to interact with MongoDB, it makes life easier by providing convenience in managing data.

Mongodb: native driver in Node.js to interact with MongoDB.

4.3.2 Mongoose

Let's integrate mongoDB into our Todo App

4.3.3 Summary

1. All the string, array, and object built-in method eg: forEach, filter, slice, indexOf, map,reduce…
2. Closure
3. Event Loop
4. Async programming
5. DOM manipulation(Event bubbling and Event capture)
6. This keyword

1. React component life cycle(concept)
2. React Hooks(code challenge)
3. How to communication with backend(code challenge)
4. Redux/thunk(concept)
5. Practice implementing some Apps, Todos, search bar, calendar (within 40 minutes