INDEPENDENT STUDY -

Full Stack Dev, React.js

[MERN stack]

Student: Yashwanth Nerella
Mentor: Dr Rohit Aggarwal

Semester: Spring 2017

Credits: 3.0

TABLE OF CONTENTS

Contents

Intro to React (Rohit's 2 sessions)

Time Spent: 2 weeks

- Learned basic concepts such as
  - React Elements
  - React Components – stateless & stateful components
  - this.state & this.props
  - constructor() & super()
  - this.setSate()
  - "render" method of both react elements & ReactDOM
  - Props
  - Data passing from parent to child (using props)
  - Data passing from child to parent (using callbacks)
- Understood the basic set up of a React application such as
  - Boilerplate
  - Why package.json is created?
  - How to structure your application
  - Usage of Babel transpiler, ES6 syntax & Webpack
  - How to structure a webpack.config file
  - Why do we need babel.rc file
- Created the following applications to better understand the concepts of react elements, props & data passage.
  - Hello World Application
  - Chat Application
  - Calculator Application
  - ReactJs All Basics Final (includes React Router)

Note: Instructions to run the applications

1. Download the zip file and unzip it
2. Do "npm install"
3. Do "npm start"

This applies to all the applications further ahead also.

Other sources of learning basics:

- **Official Documentation:** https://facebook.github.io/react/docs/hello-world.html
- **Some good tips that I followed:** https://camjackson.net/post/9-things-every-reactjs-beginner-should-know
- https://www.tutorialspoint.com/reactjs
- **YouTube video reference:** https://www.youtube.com/watch?v=JPT3bFIwJYA&list=PL55RiY5tL51oyA8euSROLjMFZbXaV7skS
- https://www.youtube.com/playlist?list=PLQDnxXqV213JJFtDaG0aE9vqvp6Wm7nBg *(Part 1 to 5 in this series)*
- https://www.youtube.com/watch?v=BYbgopx44vo
- https://www.youtube.com/watch?v=OoS5AK6qz1Q

  Note: Though I used lot of material and documentation online, my primary references were always these.

Babel, ES6, Webpack - Meetup1

Time spent: 1 week

- Learned the backbone concepts that support React applications such as
  - Babel
  - ES6
  - Webpack
  - Node.js/ Express server

- Most of this learning was in terms of reading and watching videos on various educational platforms. These are the links that I referred to learn the above concepts

  BABEL

  - **Official documentation:** https://babeljs.io/
  - **Other sources:** https://www.youtube.com/watch?v=CozSF5abcTA

  ES6

  - http://es6-features.org/
  - https://www.youtube.com/watch?v=CozSF5abcTA
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/import
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/export
  - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/let
  - https://www.youtube.com/watch?v=IEf1KAcK6A8
  - https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBhgcpA8eTYYWg7im72LgLt

  WEBPACK

  - **Official Documentation:** https://webpack.js.org/
  - https://medium.com/@rajaraodv/webpack-the-confusing-parts-58712f8fcad9

I have learned these concepts as much needed for understanding the working of React applications. I have also explained the same concepts in UReactjs meetups to a class size of 35 members

FLUX ARCHITECTURE

Vanilla flux, Redux, Alt.js*

Time spent: 3 weeks

Learned the flux architecture to pass data in a unidirectional way among components in React applications. There are many libraries that achieve the concepts presented in this architecture. 3 among those are

1. Plain Flux
2. Redux
3. Alt.js

I started learning flux from the *Vanilla/Plain flux library*. Then built on those concepts by learning *Redux & Alt.js*

Plain/ Vanilla Flux

- **Official Documentation:** https://facebook.github.io/flux/docs/in-depth-overview.html#content
- YouTube: https://www.youtube.com/watch?v=iwbkgOq1SMQ

- **Apps:**
  - First Flux
  - Chat App Plain Flux
  - User Profiles Plain Flux (includes React Router) I learned how to tackle onChange events when there are multiple inputs.

Redux

- **Official Documentation:** http://redux.js.org/
- YouTube video reference: https://www.youtube.com/playlist?list=PL6gx4Cwl9DGBbSLZjvleMwldX8jGgXV6a
- https://www.youtube.com/playlist?list=PL55RiY5tL51rrC3sh8qLiYHqUV3twEYU
- **Apps:**
  - Blog RFlux (includes React Router + React Bootstrap + Jest)

* - my major focus was on Vanilla flux & Redux for the independent study. I have learned the concepts of Alt.js very well but did not use them in Independent study applications. I used it for the capstone project.

JAVASCRIPT BASICS

Time spent: 1 week

At this stage I felt there is a pressing need to learn JavaScript thoroughly and I spent 1 week learning the basic concepts of it from *w3schools* and *mozilla.org*.

- https://www.w3schools.com/js/default.asp
- https://developer.mozilla.org/en-US/docs/Web/JavaScript I completed the following tutorials in this link
    - Complete Beginners
    - JavaScript guide
    - Intermediate

This week was just theoretical learning and less of coding. Yet I had to make two applications for Pluralsight interview.

- Coin Kata
- Package Installer

NODE.JS & EXPRESS SERVER

PUG/JADE & EJS TEMPLATING ENGINES

Time spent: 2 weeks

Learned the usage of Node.js and the need for a server. I chose to learn the API of Express server.

- https://expressjs.com/en/4x/api.html
- YouTube Video reference:
  - https://www.youtube.com/watch?v=gnsO8-xJ8rs
  - https://www.youtube.com/playlist?list=PL55RiY5tL51oGJorjEgl6NVeDbx_fO5jR

- Apps:
  - Node-Express Practice Application
  - Watson Chat App
    - http://yash-watson-chatapp.mybluemix.net (Trained to answer only Hi, Hello, Bye kind of messages) Will be unavailable after 2 weeks (i.e. 05/04)

Though I created only two applications, it took me a lot of time to understand server-side scripting and the way the middleware and routing works on this end

UTILITIES

Time spent: 2 weeks

React is just a view library and does not come with a lot of useful functionalities along with it. Therefore, we need to learn other small libraries to achieve certain functionalities. Some of them that I learned are:

React Router – ***For client side routing***

React Bootstrap – ***To write maintainable code when using bootstrap***

Super Agent – ***For connecting to external API's***

JOI – ***For validating user input and forms***

Typeahead – ***For helping users find things they need in an easy manner***

Promises ***– For avoiding callback pyramids***

Mongoose – ***ORM for MongoDB***

Socket.io – ***For real-time communication between server and client***

React Router & React Bootstrap:

- Apps have already been posted above which contain routing and bootstrap libraries

Superagent:

- Weather Application

JOI & Typeahead:

- Blog App with Joi validations and Typeahead functionality

Promises:

- Yash Watson Chat App using Promises

Mongoose:

- Node-Express Practice + Mongoose Application
  Note: start mongodb on port 27017 before running app

Socket.io:

- Real time Chat Application using Sockets

  Note: To run the application, please follow below instructions –

  1. Download and unzip the project
  2. Do "npm install"
  3. Command to start the application: "node server"
  4. To use the application:
     a. Open the browser and type "localhost:3000"
     b. Open another tab and type "localhost:3000"
     c. Follow the instructions on both tabs to simulate a real-time chat