

Naimul Islam Naim

ReactJS: An Open Source JavaScript Library for Front-end Development

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Thesis

30 May 2017

Author(s) Title	Naimul Islam Naim ReactJS: An Open Source JavaScript library for front-end development.
Number of Pages Date	36 pages + 0 appendices 30 May 2017
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Markku Karhu, Principal Lecturer
<p>The main purpose of this final year thesis was to study a JavaScript based front-end library for web and mobile application development. There are many front-end application development frameworks and libraries at the moment. ReactJS library, one of the most recent web technologies, is one of them. It has been proven as the fastest rendering library. It focuses on the view part of the MVC pattern and is being widely adopted for big scale application development.</p> <p>Being developed by Facebook for their internal use, it has proved as an efficient and fast library compared to other technologies. However, it has been open-sourced later on and enriched with more functionalities everyday by plenty of contributors. When it comes to dealing with large amounts of data and users, it has been quite successful to give better user experiences. Alongside Facebook, some other big organisations and applications are also using ReactJS and React Native for their development. Instagram, Netflix, Airbnb are a few of the big names serving smoothly enormous numbers of users worldwide. Those big names prove that ReactJS is serving them quite well.</p> <p>Throughout the research, the main intention was to evaluate the library and to prove ReactJS as a compatible platform to be adopted where there are several options to choose from. The fundamentals, core architecture, features, data handling methods, popularity, and adoptability were discussed in this study. Even though, there is no constraint to use ReactJS over other frameworks it is recommended to use it as an emerging web technology to be adopted depending on the nature of the intended application to be built.</p>	
Keywords	ReactJS, Components, JSX, Virtual DOM, JavaScript, Flux, Redux, React Native, HTML, Library, Front-end, Framework

Contents

1	Introduction	1
2	Why Learn React	2
2.1	Short and Easy Learning Curve	2
2.2	React is Fast and Agile	4
2.3	React Introduced JSX	5
2.4	Big Development Community	5
3	Environment Setup	6
3.1	Installing Text Editor	6
3.2	Installing Nodejs Bundles	6
3.3	Creating Web Server	7
4	React Core Architecture	13
4.1	React Virtual DOM	13
4.2	One-Way Data Flow	15
4.3	React Components	16
4.4	Introducing JSX Syntax	17
4.4.1	JSX characteristics	17
4.4.2	Practical JSX	18
4.4.3	Children in JSX	20
5	React Native	22
5.1	Setup and Bundling in React Native	22
5.2	Styling in React Native	22
5.3	Animation and Navigation in React Native	24
5.4	Who are Using React Native?	24
6	React vs Other Frameworks	26
6.1	Comparison	26
6.2	React in Market Growth and Popularity	28
6.3	React Library vs Angular Framework	28
7	Data Flow Management in React	30
7.1	Flux as an Architecture from Facebook	30

7.2	Redux Data Flow Management	31
7.3	Which to Choose Between Flux and Redux	33
8	Discussion	35
9	Conclusion	36
	References	37

Abbreviation and Terms

API	Application Program Interface
DOM	Document Object Model
HTML	Hypertext Markup Language
JS	JavaScript
JSX	JavaScript XML
URL	Uniform Resource Locator

1 Introduction

Internet has become a busy hub for searching information and doing different tasks virtually that used to be done manually before the internet age. There are enormous numbers of mobile and web applications that have made it easier to do different tasks. A big part of our everyday task can be done on the internet at the current age. Faster internet along with fast performing devices demands faster applications.

A trend is growing to shift software or applications we used to use in desktop machines to the web. There are plenty of applications that are usable from web and mobile devices. Several JavaScript based frameworks and libraries are used to develop different applications. There are ReactJS, AngularJS, EmberJS, MeteorJS, VueJS, KnockoutJS and many more at this moment in production. ReactJS is one of them for the front-end development of applications.

React is a popular open source front-end JavaScript library developed by Facebook. React is widely popular among developer communities because of its simplicity and easy but effective developing process. React makes it easier to create interactive user interfaces. It efficiently updates through rendering the exact components to the view of each state and makes the data changes in the application.

In ReactJS, every component manages their own state and composes them to the user interfaces. This concept of components instead of templates in JavaScript, plenty of data can easily be passed to the app and thus keep the state out of the DOM. Using Node React can also be rendered on the server side. Alongside web apps, to build mobile applications we can use React Native as well.

The purpose of the thesis is to carry out an in-depth research of the ReactJS library based on JavaScript. The fundamental concepts, characteristics, features, development processes, core architecture and market research as well as compatibility will be covered in the thesis. The aim is to provide a solid understanding of the ReactJS library.

2 Why Learn React

React was introduced to the world two years ago, and since then it has seen impressive growth, both inside and outside Facebook. [1] New web projects at Facebook are commonly built using React in one form or another, and it is being broadly adopted across the industry. [1] Developers and engineers are choosing React because it allows spending more time to focus more on the product development and less time spent on fighting and learning to the framework.

A React application is a collection of discrete components, each representing a single view. The idea of every individual view component makes it easy to iterate on product development because to make changes on a single view or component, it is not necessary to consider the entire system. When an application is built with React, the code is generally predictable, it is because React wraps the DOM mutative, imperative API with a declarative one, which raises the level of abstraction and simplifies the programming model. [1] Moreover, it is easier to scale the application built with React.

The combination of React and the rapid iteration cycle of the web, has enabled to make some excellent products including many Facebook components. An amazing JavaScript framework called Relay has also been made on top of React, which helps simplifying data fetching on a large scale. [1]

2.1 Short and Easy Learning Curve

Unlike some other JavaScript libraries where it takes a lot of time to learn about the frameworks, in React it does not take much of an effort to start building an application. React is comprised of many strong features. Readability is one of the greatest strength of React. It is easily readable even to those who are unfamiliar to it. While other frameworks require learning many concepts about the framework itself, ignoring the language fundamentals, React does the absolute opposite. [3] For example, let's consider how different it is in React and Ionic (AngularJS) rendering a portion of an employer's list.

In Ionic, it requires to use the directive called `ngRepeat`. Let's assume an array of employers. Each of them contains the following fields: `first_name`, `last_name`, `is_married`.

The target is to show only employers who are married. The following Figure 1 shows a screenshot of a function written in Ionic framework.

```
1 <function EmployerCtrl($scope) {  
2   $scope.employers = [  
3     {  
4       first_name: 'Naimul',  
5       last_name: 'Naim',  
6       is_married: true,  
7     },  
8     {  
9       first_name: 'Karar',  
10      last_name: 'Habib',  
11      is_married: true,  
12     },  
13     {  
14      first_name: 'Tanmoy',  
15      last_name: 'Zadid',  
16      is_married: false,  
17     }  
18   ];  
19 }
```

As shown in Figure 1, a function is written called `EmployerCtrl` where it shows some specific information of the employers.

Figure 2 shows a screenshot of a directive called `ngRepeat` written in AngularJS framework.

```
1 <div ng-controller="EmployerCtrl">  
2   <ul>  
3     <li ng-repeat="employer in employers | filter:{is_married:true}">  
4       {{employer.last_name}}, {{employer.first_name}}  
5     </li>  
6   </ul>  
7 </div>
```

As shown in Figure 2, this is how a directive is written in AngularJS framework.

If one is not familiar with Ionic/Angular, this code snippet may raise some immediate questions of what is the `$scope` and what is the specific syntax here for the filter.

But in React, one can use the existing knowledge of language fundamentals. The above functionality can be done using filter and map functions in React. The following Figure 3 shows how to write the function in ReactJS.

```
1  const DemoComponent = React.createClass({
2    render() {
3      const employers = [
4        {
5          first_name: 'Naimul',
6          last_name: 'Naim',
7          is_married: true,
8        },
9        {
10         first_name: 'Karan',
11         last_name: 'Habib',
12         is_married: true,
13       },
14       {
15         first_name: 'Tanmoy',
16         last_name: 'Zadid',
17         is_online: false,
18       }
19     ];
20
21     return (
22       <View>
23         {employers
24           .filter(f => f.is_married)
25           .map(f => <View>{f.last_name}, {f.first_name}</View>)}
26       </View>
27     );
28   }
29 });
```

Figure 3 Screenshot of how to use filter and map function in ReactJS.

As shown in Figure 3, functions can be written in easy ways compared to other frameworks. A few questions may also rise regarding what `React.createClass` does and what `render` is but the rest of the code is regular JavaScript. This means it will be quite easy and understandable to those who do not even know much of React.

2.2 React is Fast and Agile

ReactJS is featured with one-way unidirectional data flow between the states and layers in an application. This means data flows in single direction between the application states and layers. In two-way data binding like Angular, if a model is changed, the view also changes and vice-versa. React renders the updates in the DOM much quicker than alternative frameworks and it is a much smaller library. DOM means document object model. Thus, it is easy to choose the tools to get the job done.

2.3 React Introduced JSX

JSX is a language that lets you specify the DOM elements before the components right inside of JavaScript files. This means the logic behind the components and the visuals are all in one place. This is such a great idea when other frameworks are taking queues to place them.

2.4 Big Development Community

Big companies like New York Times, Airbnb, Facebook, and Netflix are using React in production. They are continuously contributing to develop the React core and building amazing third party libraries that work great with any React applications.

3 Environment Setup

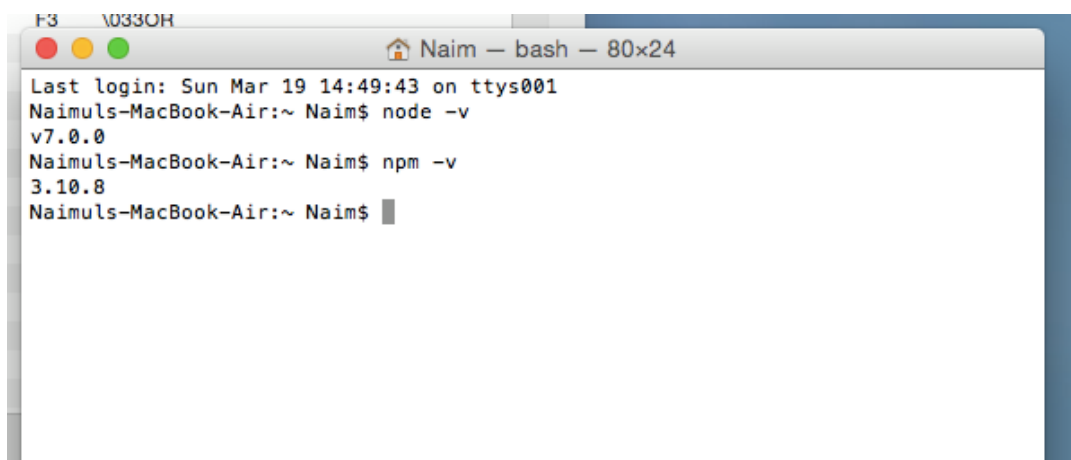
3.1 Installing Text Editor

There are plenty of text-editors to start working with. Most of them are open source and free of cost. Atom is one of them. It is a very useful text-editor. It has a great community of developers around it and they have enough useful plugin updates constantly. It is usable in every platform including Windows, Mac and Linux.

For Windows users it is needed to install Git-bash while mac and Linux users can do the job from the terminal. The reason behind using Git bash in Windows is so that you can have access to the same commands that are available in Linux environments like Ubuntu distribution or on a mac laptop.

3.2 Installing Nodejs Bundles

To install Nodejs, it is needed to go to the website called nodejs.org and there are couple of download facilities available for different operating systems. Node can be downloaded from there. This download will install a couple of things. First it will install nodejs. Nodejs allows creating a web server so that React components can be used locally and can be deployed to the web directly. It also installs node package manager called npm which will let us install various third party modules like React into our applications. Figure 4 shows a screenshot of some initial setup.

A screenshot of a terminal window on a Mac. The window title is "Naim - bash - 80x24". The terminal shows the following text:

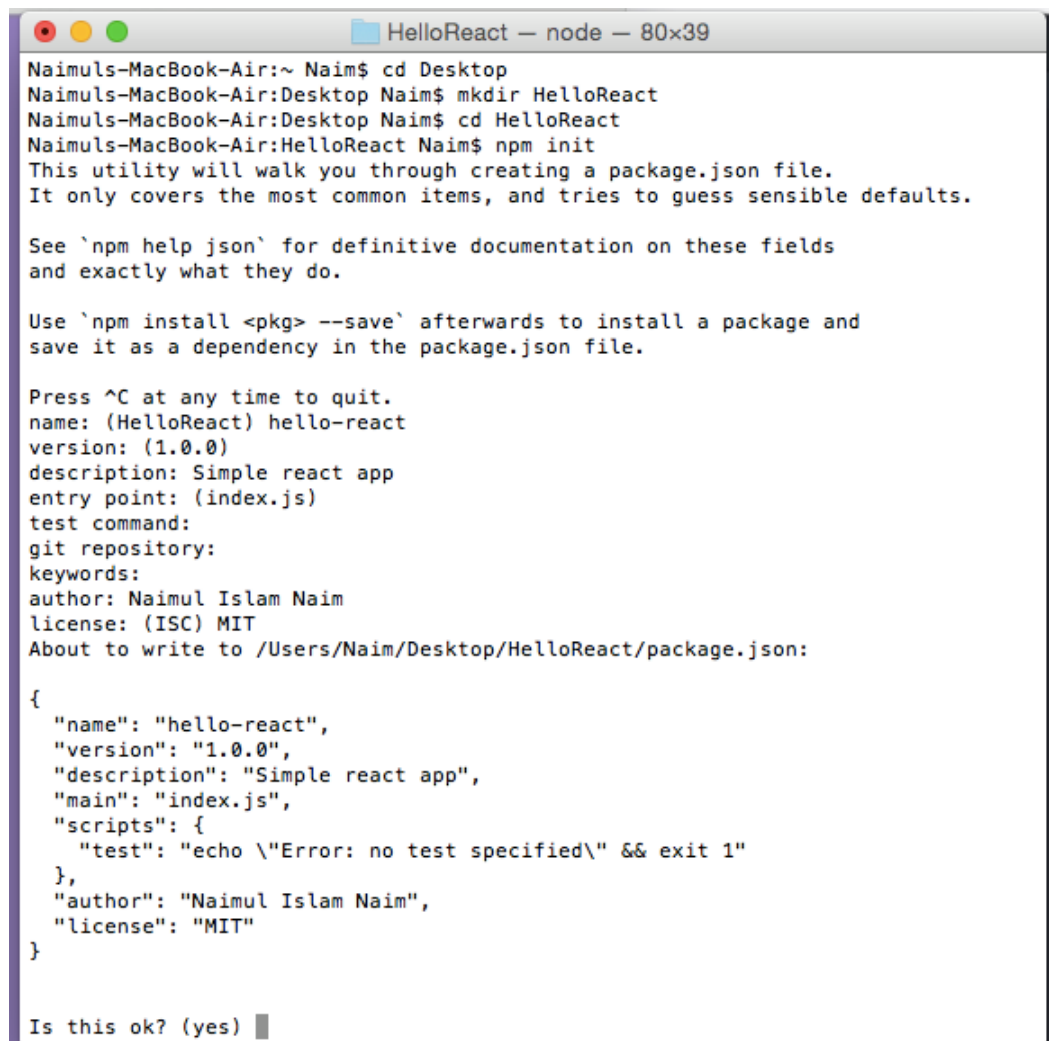
```
F3  \033OH
Last login: Sun Mar 19 14:49:43 on ttys001
Naimuls-MacBook-Air:~ Naim$ node -v
v7.0.0
Naimuls-MacBook-Air:~ Naim$ npm -v
3.10.8
Naimuls-MacBook-Air:~ Naim$ █
```

Figure 4 Screenshot of installation of Nodejs and npm package manager.

As shown in Figure 4, Nodejs and npm package manager are both primary requirements for ReactJS development setup. Node and npm both come in a bundle while downloading from the node website.

3.3 Creating Web Server

To start working with React it is necessary to create a simple webserver beforehand. Without a webserver, there is no way to see the files in the browser. After opening the terminal we can use `npm-init` to create a new node project. Here first we create a folder in the desktop to store the project named HelloReact. We run `npm-init` command from the terminal and it creates one file in the project. It gives a little introduction of what exactly it is doing and then asks a few basic questions. Figure 5 will show how to create a `package.json` file for the project.



```
Naimuls-MacBook-Air:~ Naim$ cd Desktop
Naimuls-MacBook-Air:Desktop Naim$ mkdir HelloReact
Naimuls-MacBook-Air:Desktop Naim$ cd HelloReact
Naimuls-MacBook-Air:HelloReact Naim$ npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg> --save` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
name: (HelloReact) hello-react
version: (1.0.0)
description: Simple react app
entry point: (index.js)
test command:
git repository:
keywords:
author: Naimul Islam Naim
license: (ISC) MIT
About to write to /Users/Naim/Desktop/HelloReact/package.json:

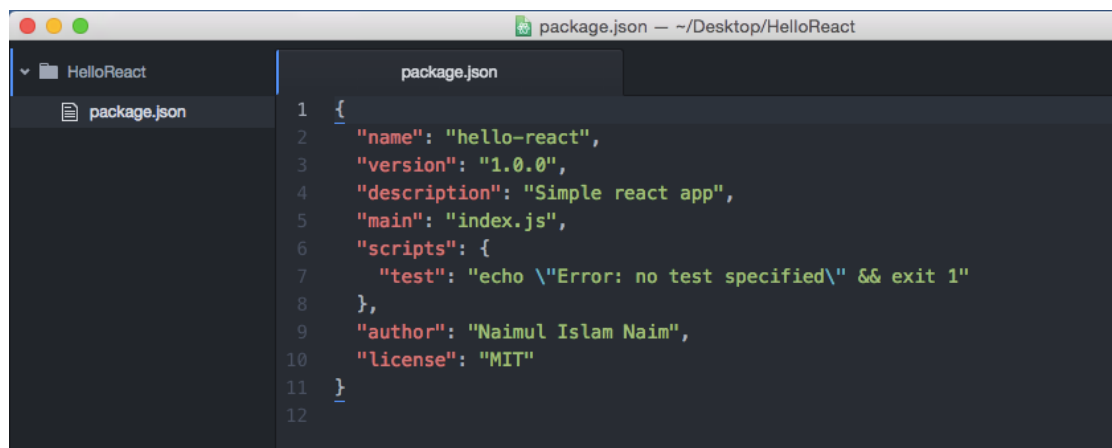
{
  "name": "hello-react",
  "version": "1.0.0",
  "description": "Simple react app",
  "main": "index.js",
  "scripts": {
    "test": "echo `Error: no test specified` && exit 1"
  },
  "author": "Naimul Islam Naim",
  "license": "MIT"
}

Is this ok? (yes) █
```

Figure 5 Screenshot of Creating `package.json` file in project folder.

As shown in Figure 5, running `npm-init` command in HelloReact project lets us walk through the setup for the `package.json` file. After answering all those questions including the file name, version, author and license, a little file prints out telling that it is about to write a file into our HelloReact project called `package.json`. This file is used for not only the node server but also to manage React dependencies.

Now, if the new folder called HelloReact is opened inside the Atom editor it shows the `package.json` file there looks exactly what has been printed out in the terminal. Figure 6 shows the `package.json` file in the editor inside HelloReact Folder.



```
1 {
2   "name": "hello-react",
3   "version": "1.0.0",
4   "description": "Simple react app",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \\\"Error: no test specified\\\" && exit 1"
8   },
9   "author": "Naimul Islam Naim",
10  "license": "MIT"
11 }
12
```

Figure 6 Screenshot of `package.json` file in the project folder.

As shown in Figure 6, `Package.json` consists of the answers that were made earlier in the terminal.

Now the first module is to be installed. A module is a third-party code or library we will use in our application. The following command has to be typed in the terminal to install the module.

```
npm install express@4 -save
```

Figure 7 is a screenshot of installing an `express` module in the project.

```

HelloReact — bash — 80x48
Is this ok? (yes) yes
Naimuls-MacBook-Air:HelloReact Naim$ npm install express@4 --save
hello-react@1.0.0 /Users/Naim/Desktop/HelloReact
├─┬ express@4.15.2
│   ├── accepts@1.3.3
│   │   ├── mime-types@2.1.14
│   │   └── mime-db@1.26.0
│   ├── negotiator@0.6.1
│   ├── array-flatten@1.1.1
│   ├── content-disposition@0.5.2
│   ├── content-type@1.0.2
│   ├── cookie@0.3.1
│   ├── cookie-signature@1.0.6
│   ├── debug@2.6.1
│   │   └── ms@0.7.2
│   ├── depd@1.1.0
│   ├── encodeurl@1.0.1
│   ├── escape-html@1.0.3
│   ├── etag@1.8.0
│   ├── finalhandler@1.0.0
│   │   └── unpipe@1.0.0
│   ├── fresh@0.5.0
│   ├── merge-descriptors@1.0.1
│   ├── methods@1.1.2
│   ├── on-finished@2.3.0
│   │   └── ee-first@1.1.1
│   ├── parseurl@1.3.1
│   ├── path-to-regexp@0.1.7
│   ├── proxy-addr@1.1.3
│   │   ├── forwarded@0.1.0
│   │   └── ipaddr.js@1.2.0
│   ├── qs@6.4.0
│   ├── range-parser@1.2.0
│   ├── send@0.15.1
│   │   ├── destroy@1.0.4
│   │   ├── http-errors@1.6.1
│   │   └── inherits@2.0.3
│   ├── mime@1.3.4
│   ├── serve-static@1.12.1
│   ├── setprototypeof@1.0.3
│   ├── statuses@1.3.1
│   ├── type-is@1.6.14
│   │   └── media-typer@0.3.0
│   └── utils-merge@1.0.0
└── vary@1.1.0

npm WARN hello-react@1.0.0 No repository field.
Naimuls-MacBook-Air:HelloReact Naim$

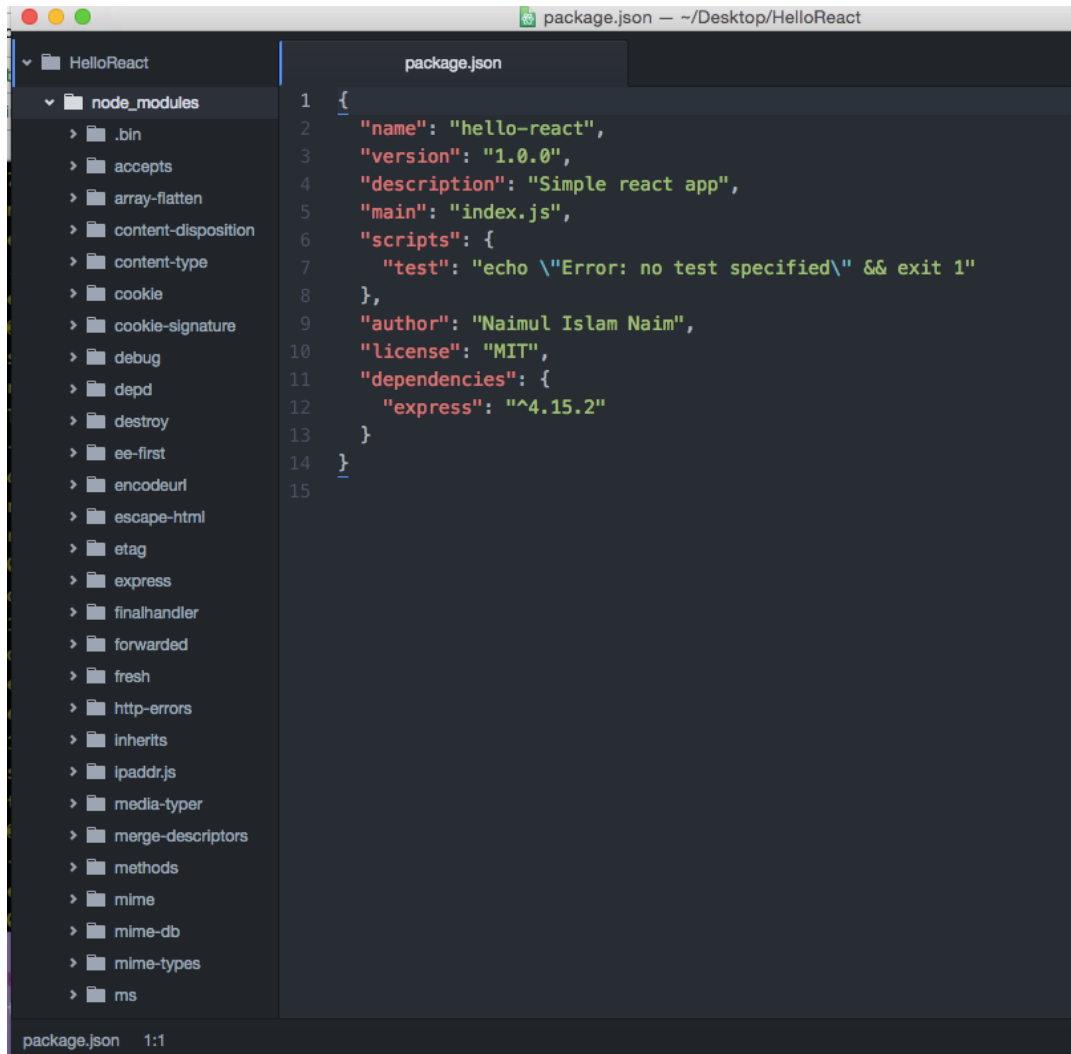
```

Figure 7 Screenshot of Installation of express module.

As shown in Figure 7, when the express module is saved in the project folder all the dependencies are also automatically downloaded.

To write the `-save` flag is important because without saving the flag it is not going to update the package.json file which stores all the dependencies. Inside Atom there is a node-module folder which has plenty of files in it. And also in the package.json file an express module is already installed.

Figure 8 is a screenshot of a node module folder with a plenty of files inside it.

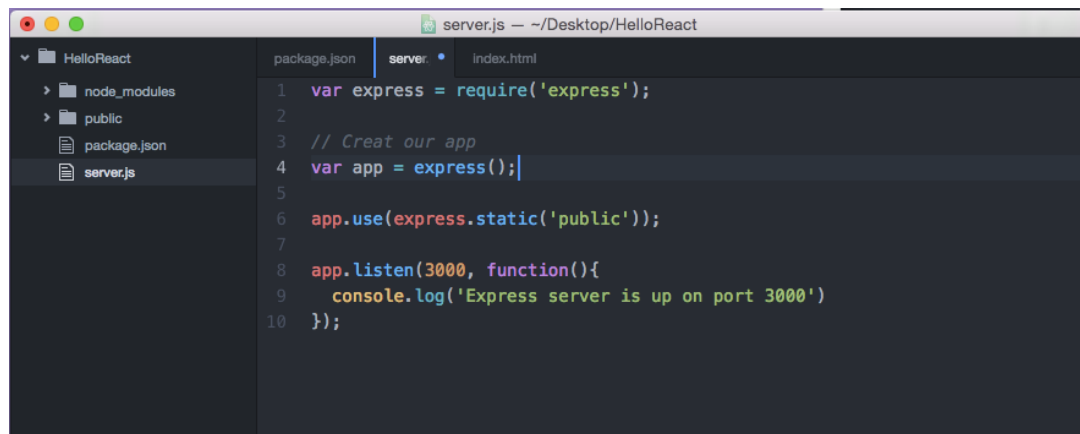


```
package.json
1  {
2    "name": "hello-react",
3    "version": "1.0.0",
4    "description": "Simple react app",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \\\"Error: no test specified\\\" && exit 1"
8    },
9    "author": "Naimul Islam Naim",
10   "license": "MIT",
11   "dependencies": {
12     "express": "^4.15.2"
13   }
14 }
15
```

Figure 8 Screenshot of a node module folder in the project.

As shown in Figure 8, node_modules in the project folder contain all the important files with dependencies. This same action can be done only with npm install command as long as the “express” module is present in the package.json file. Only command npm install can do the job as well.

Now in the root of the project a new file called server.js has been created and let us run it in localhost port 3000. Figure 9 is a screenshot of the file server.js inside the project folder as follows.



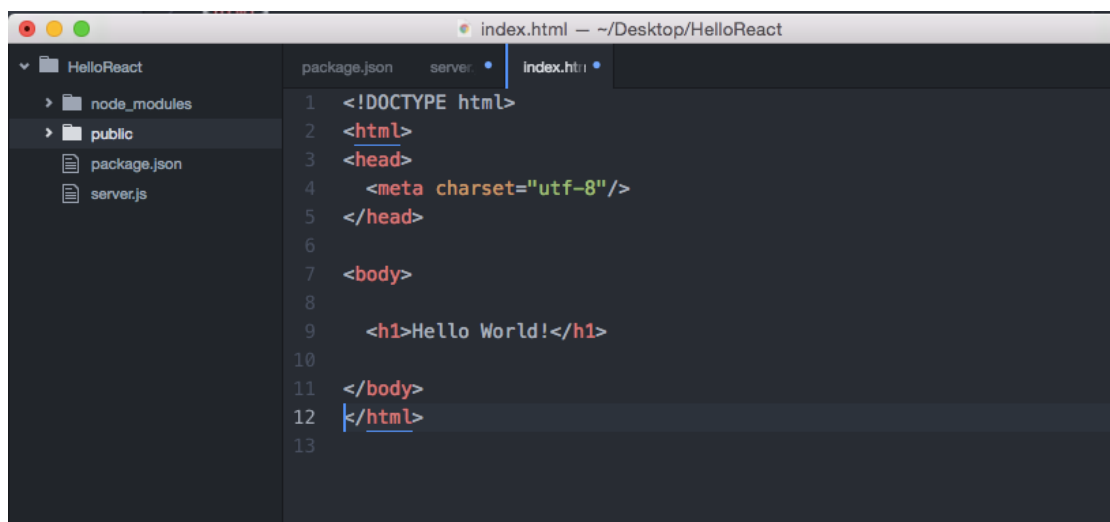
```

server.js -- ~/Desktop/HelloReact
package.json  server.js  index.html
1  var express = require('express');
2
3  // Create our app
4  var app = express();
5
6  app.use(express.static('public'));
7
8  app.listen(3000, function(){
9    console.log('Express server is up on port 3000')
10 });

```

Figure 9 Screenshot of Server.js file in the public folder.

As shown in Figure 9, the public folder has a server.js file in it. A public folder and index.html has been created in the root of the application. Index.html is the default file of the application. Figure 10 shows a public folder in the editor is a screenshot.



```

index.html -- ~/Desktop/HelloReact
package.json  server.js  index.html
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="utf-8" />
5  </head>
6
7  <body>
8
9    <h1>Hello World!</h1>
10
11 </body>
12 </html>
13

```

Figure 10 Screenshot of Index.html as the default file of the project.

As shown in Figure 10, the public folder inside the main project folder contains the package.json and server.js file.

Now if the command `ls` is run in the terminal it can be seen that all the files and folders are showing up in terminal. Figure 11 shows if every file and folder is placed inside the main project folder properly.

```
npm WARN hello-react@1.0.0 No repository field.
Naimuls-MacBook-Air:HelloReact Naim$
Naimuls-MacBook-Air:HelloReact Naim$ ls
node_modules  package.json  public          server.js
Naimuls-MacBook-Air:HelloReact Naim$ node server.js
Express server is up on port 3000
```

Figure 11 Screenshot of all files showing up in the same project folder.

As can be seen in Figure 11, ls command shows all the files and folders inside the project folder and the express server is ready to run on port 3000.

Now let us run the server in web typing node server.js command in the terminal. Figure 12 is a screenshot that shows the server running on the localhost.

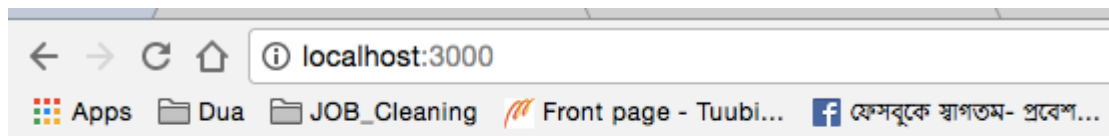


Figure 12 Screenshot of project deployment in localhost with node server.

As Figure 12 shows, the project is deployed in the localhost with node server at port 3000.

In this section, the procedure of using ReactJS in the local machine was described thoroughly. There are several other text editors available to start writing code. Atom is one of them and it is a good one. It has many plugins available and is easy to follow.

4 React Core Architecture

4.1 React Virtual DOM

DOM stands for Document Object Model. DOM manipulation is very important for modern interactive web technologies. It is often called the heart of the modern web. It is an abstraction of the structured text. But it works slower than other JavaScript operations because most JavaScript frameworks usually update the DOM even if they do not need to do it. That means those updates are not necessarily required to perform the actions but they still do by default. For example, let us assume nine items have been put in a shopping basket in an online web store. Now let us say only the first item is needed to buy and proceed to checkout. Here, most technologies would rebuild the entire list that has been put in the basket. This means the framework has to unnecessarily work ten times more. Because of only one change the system has to rebuild the list exactly how it was before.

React did not invent Virtual DOM but uses and provides it to the developer community for free. Virtual Dom is simply an abstraction of HTML DOM. React has a corresponding virtual DOM object for every DOM object like a correspondent or a lightweight copy. Virtual DOM is also characterized with similar properties to a real DOM. However, it cannot make any changes directly to the view. DOM manipulation is quite a slow process. But manipulating Virtual DOM is faster because it has nothing to do with the view part and does not make any changes to the screen. Figure 13, reprinted from stackoverflow.com, is an illustration of Virtual DOM in the memory.

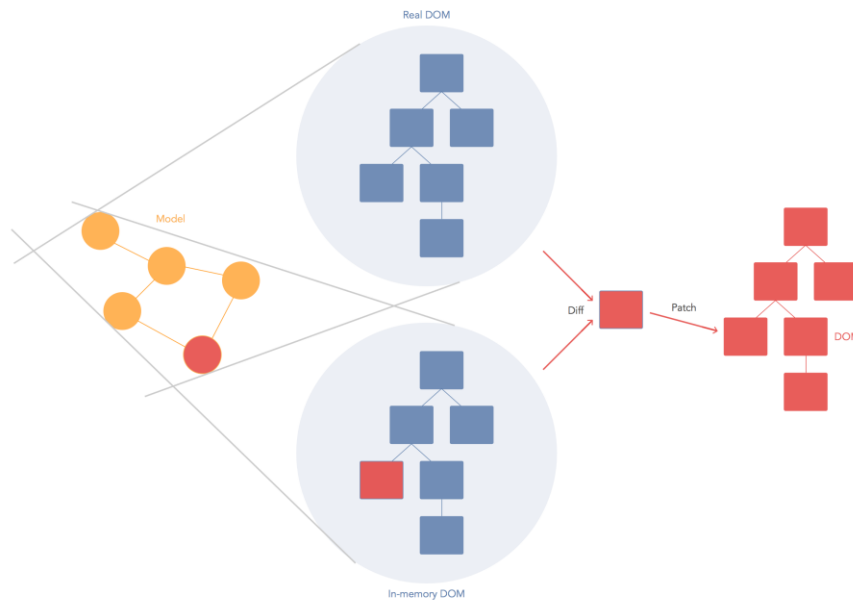


Figure 13 Screenshot of React virtual Dom in memory. Reprinted from stackoverflow.com (2017). [9]

As shown in Figure 13, a React virtual DOM in the memory is a lightweight copy of the real DOM. React uses a method called “diffing” which means rendering a JSX element gets every single Virtual DOM updated. This might sound inefficient but, in fact, it costs nothing as Virtual DOM is quite fast to get updated and does not make any impact in the process. After the DOM gets updated React compares the updated DOM with a pre-updated state of the DOM and determines which virtual DOM has been changed. Once React detects the changed DOMs, React updates only those objects to the real DOM.

Thus, React makes the updates faster through Virtual DOM. In the above-mentioned example, React would have updated only the checked item from the list and leave the rest of the items alone. This makes the difference when updating a page in an application while React can only make changes to the necessary parts of the DOM. This virtual DOM manipulation process is one of the main reasons why React is gaining much popularity among the developer communities.

React Virtual DOM Pros and Cons

Among the many advantages of the ReactJS library, a few of the key advantages are described here.

- The diffing algorithms written in React is quite fast and efficient
- Inclusion of JSX and hyperscript let us build multiple frontends for the same application.
- It is very lightweight and capable to be run in every mobile device
- Lots of tractions and mindshare
- It can also be used without React as an independent engine

A few disadvantages of React are as follows.

- It occupies quite much of the memory. Full in memory copy of the DOM.
- Static and dynamic elements don't bring much of a difference.

4.2 One-Way Data Flow

Frameworks like Angular and Ember use two-way data binding. In a two-way data binding for example in Angular, if a model is changed, the view also automatically gets changed and vice versa. An input field in the model can also mutate the model. It performs well in most of the applications but sometimes it may lead to cascading updates and changing to one model may cause updates in other models. Again, since the state is mutable by both view and controller, the data flow can be unpredictable in some cases. Flux or Redux with React can be a better solution to avoid those uncertainties since both architectures follow one-way data flow. One-way data flow does not make cascading updates and changes in view.

One way data flow ensures that data flows throughout the application in a single direction to offer more control between the states and models in an application. One way data flow also makes the architecture less complicated and understandable. Flux architecture is a functional approach. Here the view is considered as a function of the application state. Eventually, if the state gets some changes the view also gets re-rendered

automatically. Moreover, a similar view is generated from the states and gives a better understanding and predictability to the application.

To make it more predictable, in an application, data from parent to child flows in a single direction. Any data can be updated from any view, anytime in this approach. In case something goes wrong, debugging is also made less complicated in this way.

4.3 React Components

Components are very important for React. It is often considered as the heart of React, which is a collection of components. It is small reusable UI element that provides data to the view and changes over time. [6] To create the entire UI, those small components are then composed together, nested inside one another. Components let the UI (user interface) to be split into small pieces and to design and build in a comprehensive way. UI stands for user interface, i.e. what is shown on the screen. Components are like JavaScript functions. They literally perform the same task but in different environment and different approaches. Like functions, they take inputs called props and return React elements. Those elements describe what the user sees in the interface on the screen. React components can be used to build the entire interface or even a part of it.

Creating a React Component

A React component can be simply written as a JavaScript function. This function accepts props and returns a React element. They are called as functional components. ES6 class can also be used to define a component.

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}
```

A React component can also be created in several other ways. To extend or to inherit or to derive a class from the main component which it attached to object is another way to create a component. [7]

```

class Album extends React.Component { // It's the main React
  component class!
  render() {
    return(
      <div>
        <h1>Pink Floyd- The division Bell</h1>
        <ol> // Songs list
          <li>Cluster One</li>
          <li>What do you want from meli>
          <li>marooned</li>
          <li>great day for freeDOM</li>
          <li>Wearing the inside out</li>
          <li>Take it back</li>
          <li>Coming back to life</li>
          <li>Keep talking</li>
          <li>Lost for words</li>
          <li>High hopes</li>
        </ol>
      </div>
    )
  }
}

```

Functional components can also be stateless. Rendering every component builds the user interface experiencing faster and efficient.

4.4 Introducing JSX Syntax

JSX is neither a string nor HTML. It is statically typed syntax extension to JavaScript. It is similar to an object-oriented language which is designed to run on modern web browsers. JSX is recommended to be used with React to design and build the user interface. While it comes with the full power of JavaScript it might even seem as a template language too at the first glance though it is not. The React element is produced by JSX. It can be rendered to the React Virtual DOM. [9]

4.4.1 JSX characteristics

JSX has got some unique features which made JSX quite popular among the React and React Native developers. At the beginning, it may look difficult but with time adopting JSX is easy.

First of all, it is faster: While JSX source code is compiled to JavaScript, it shows a very optimized result. Compared to the equivalent code written in JavaScript, JSX generated code runs faster. JSX has proved to be 12 % faster in iOS and 29 % faster in Android. [6]

Secondly, it is safer: In contrast to JavaScript, JSX is statically-typed and mostly type-safe. The quality of applications becomes higher when being developed using JSX, since many errors will be caught during the compilation process. It also offers debugging features at the compiler level as well.

Thirdly, it is easier: JSX offers a solid class system much like Java, freeing the developers from working with the too-primitive prototype-based inheritance system provided by JavaScript. Expressions and statements, however, are mostly equal to JavaScript, so it is easy for JavaScript programmers to start using JSX. There are also plans on language-services for editors / IDEs, for example code completion to make coding easier.

4.4.2 Practical JSX

Expressions can be embedded in JSX though it is an expression too. The function written below is an expression.

```
function formatName(user) {
  return user.firstName + ' ' + user.lastName;
}
const user = {
  firstName: 'Naimul ',
  lastName: 'Islam'
};
const element = (
  <h1>
    Hello, {formatName(user)}!
  </h1>
);
ReactDOM.render(
  element,
```

```
document.getElementById('root')
);
```

JSX tags can also contain children.

```
const element = (
  <div>
    <h1>Hello!</h1>
    <h2>Is React JSX worthy or not?</h2>
  </div>
);
```

Objects in React are also represented by JSX. Here are two different code examples written below but they are identical.

```
const element = (
  <h1 className="greeting">
    Hello, Naimul!
  </h1>);
const element = React.createElement(
  'h1',
  {className: 'greeting'},
  'Hello, Naimul!'
);
```

Props can be specified in JSX in several ways. JavaScript expressions, string literals can also be passed as props. If, else if, do, while statements, for loops those are not considered as expressions in normal JavaScript, so they cannot be used directly in JSX but with a surrounding.

```
function NumberDescriber(props) {
  let description;
  if (props.number % 2 == 0) {
    description = <strong>even</strong>;
  } else {
    description = <i>odd</i>;
  }
}
```

```

    return <div>{props.number} is an {description} number</div>;
  }
}

```

4.4.3 Children in JSX

JSX elements can be provided as children to help displaying nested components. Different types of children can be mixed together so that JSX children and string literals can be used together. This is another JSX property that is similar to HTML. [7]

A JSX expression can have multiple children. As a result, it has to be wrapped in a `div` if it requires the component to render multiple things. JavaScript expressions can be passed as children within `{}` enclosing. The two following expressions are identical.

```

<FirstComponent>Languages</FirstComponent>
<FirstComponent>{'languages'}</FirstComponent>

```

Functions in React JSX can also be passed as children. In general, expressions that are passed to JSX are evaluated to a string, an element or as a list of those things. Though, `props.children` works similar to other props and can pass data that not only React knows itself but any sort of data can be passed. For example, a `props.children` callback can be initiated in a custom component. [8]

```

function Repeat(props) {
  let items = [];
  for (let i = 0; i < props.numTimes; i++) {
    items.push(props.children(i));
  }
  return <div>{items}</div>;
}

function ListOfTenThings() {
  return (
    <Repeat numTimes={10}>

```



```
      {(index) => <div key={index}>This is item {index} in the list</div>}  
    </Repeat>  
  );  
}
```

Booleans, false, true, undefined, null are not rendered though they are valid children. They are ignored.

5 React Native

React Native is a framework developed by Facebook for native application development for both iOS and Android. It is used for making cross platform mobile applications using JavaScript as a language. It uses the same design as React.JS, takes everything that is great about React.js and allows to build rich mobile user interfaces from declarative components and gives a better application user experience. (2) A React Native app is truly indistinguishable compared to an application built with Objective-C for iOS or Java for Android. It uses the same fundamental User Interface as regular Android and iOS apps do.

5.1 Setup and Bundling in React Native

React native is a framework, where ReactJS is a JavaScript library for building user interfaces. [4] Starting a project with ReactJS requires a bundler like Webpack or Browserify which consists of all necessary modules within it. But in React Native, it comes with everything that is needed to start developing a project. The setup is quite easy to follow and fast. It takes only a few command lines in the terminal and becomes ready to go. A native React application can be built using ES6, ES7. [4]

It is required to have Xcode for iOS in Mac and Android Studio for Android application development preinstalled in the machine. It can either be run on a simulator or emulator of the specific platform or even directly to the own devices.

5.2 Styling in React Native

Components in React Native do not use HTML to render the application. It provides alternative components that render the app. Those alternative components map the actual UI components to get rendered on the application. [4] Most of those alternative components are then translated to something that is similar to HTML. At this phase, a **view** component is similar to a HTML **div** tag and a **Text** component is similar to a **p** tag. Figure 14 shows an alternative component in React Native.

```

1  import React, { Component } from 'react';
2  import { View, Text } from 'react-native';
3
4  export default class App extends Component {
5    render() {
6      return (
7        <View style={styles.container}>
8          <Text style={styles.intro}>Hello world!</Text>
9        </View>
10     );
11   }
12 }

```

Figure 14: Alternative components in React Native.

As shown in Figure 14, alternative components in React Native look similar to HTML. Since this code will not be rendered in HTML, so it is not possible to reuse any libraries that could be used in ReactJS that could render any type of HTML. But there are some alternative libraries for example `React.parts` get a solution of this.

A stylesheet in JavaScript looks similar to CSS and is required to style the React native components. [4] There is a tool called Flexbox which is designed to make the layout in React Native applications. [5] Figure 15 shows responsive styling with flexbox for React native applications.

```

1  const styles = StyleSheet.create({
2    container: {
3      flex: 1,
4    },
5    content: {
6      backgroundColor: '#fff',
7      padding: 30,
8    },
9    button: {
10     alignSelf: 'center',
11     marginTop: 20,
12     width: 100,
13   },
14 });

```

Figure

15: Responsive styling for React Native app using flexbox.

As shown in Figure 15, responsive styling for React Native app using flexbox is nothing complicated but quite similar to CSS styling.

5.3 Animation and Navigation in React Native

React Native has brought a new dimension to application animation design and navigation. Alongside regular features, attracting animations in application swapping, sliding is an advantage for native applications. They have designed many API's for the design, animation and navigation purposes. Those animations are good to see and use.

There is no need for CSS animations in the React Native application. Different components in the application can be animated in a completely new way using JavaScript. React Native provides own built API's to animate the components. It can be compared to the existing JavaScript library called Velocity.JS which is widely used for different kind of animation associated to gestures in the web based applications. For React native, there is an API provided called **LayoutAnimation** which is very nice and easy to use but works only in iOS at this moment. [4] It also can be used in Android but it doesn't support it very well.

There is another API called **PanResponder** which is similar to JavaScript touch events to interact with user gestures. [4] To navigate between two scenes or two pages there is a component provided by React Native called **Navigator** component.

5.4 Who are Using React Native?

A large number of app developers are using React Native in production. From established Fortune 500 companies to hot new startup companies are also using React Native in developments. [7] Figure 16 shows the applications and companies that are using React Native.

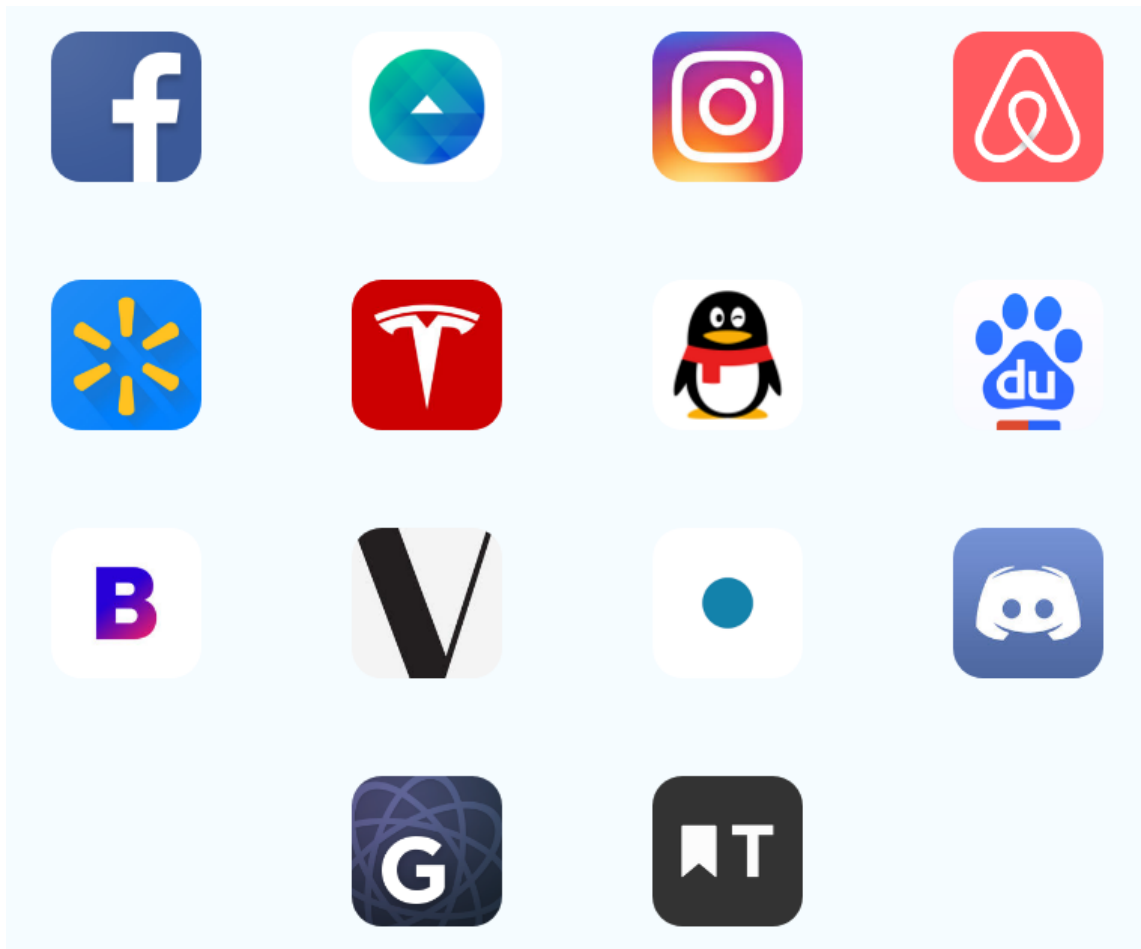


Figure 16: Apps using React Native in production. Reprinted from book.github.io/React-native/ [7]

As shown in Figure 16, apps like Facebook, Instagram, Airbnb and big organisations like Tesla, Walmart, Vogue and many other companies are using React Native in production.

6 React vs Other Frameworks

There are quite many JavaScript frameworks in the market for front-end web development. Though, ReactJS is not a framework like Angular or Meteor. Being a JavaScript library ReactJS is quite compatible when it comes to developing web applications like other popular JavaScript frameworks.

6.1 Comparison

Making comparisons between Angular and React nowadays is a popular topic in the tech community. However, AngularJS, ReactJS, VueJS , EmberJS, BackboneJS and several other web technologies are the most popular web technologies. Amongst them Angular and React are widely adopted and advanced JavaScript technologies being used to create single-page applications (SPAs).

Table 1 on the next page shows a comparison between AngularJS framework and ReactJS library. [12]

Table 1 Angular JS and React JS compared

Technology	AngularJS	ReactJS
Developed by	Google	Facebook
Technology adopted	Full-fledged MVC (model-view-controller) framework written in JavaScript	JavaScript library .View in MVC, requires Flux or Redux to implement architecture for data management Open sourced.
Concept	Converts JavaScript into HTML. Manipulates with the real DOM. Performs client-side rendering	Converts HTML into JavaScript. Works with virtual DOM. Performs server-side rendering
Data Binding	Two-way data binding	Unidirectional one-way data binding
Dependencies	Dependencies managed automatically	Third party tools are required to manage dependencies
Languages used	JavaScript and HTML	JavaScript and JSX
Suits for	Single page application (SPA) that updates single view at a time	Single page application that updates many views at a time depending on components
Latest Version Available	Angular 1.6.0 RC2	React 15

As shown in Table 1, there are some fundamentals differences between AngularJS and React in terms of data binding, dependency handling, DOM manipulation and languages they use. But the biggest difference is that Angular JS is a framework whereas ReactJS is a library. Though, both of them are used for front-end development.

6.2 React in Market Growth and Popularity

Many business, news, travel, social networking companies in the USA, the UK, Asia, France, Germany, Canada and many other countries are using both React and Angular.[7] Angular is very popular among the developers as it is a complete framework which comes with MVC (Model-View-Controller) while React is a library, not a framework. React has a View only in front lacking Model and Controller.

Figure 17, taken from Google trends, shows the ratio of searching interest over different technologies including ReactJS (blue), AngularJS (red) and VueJS (yellow) as follows.

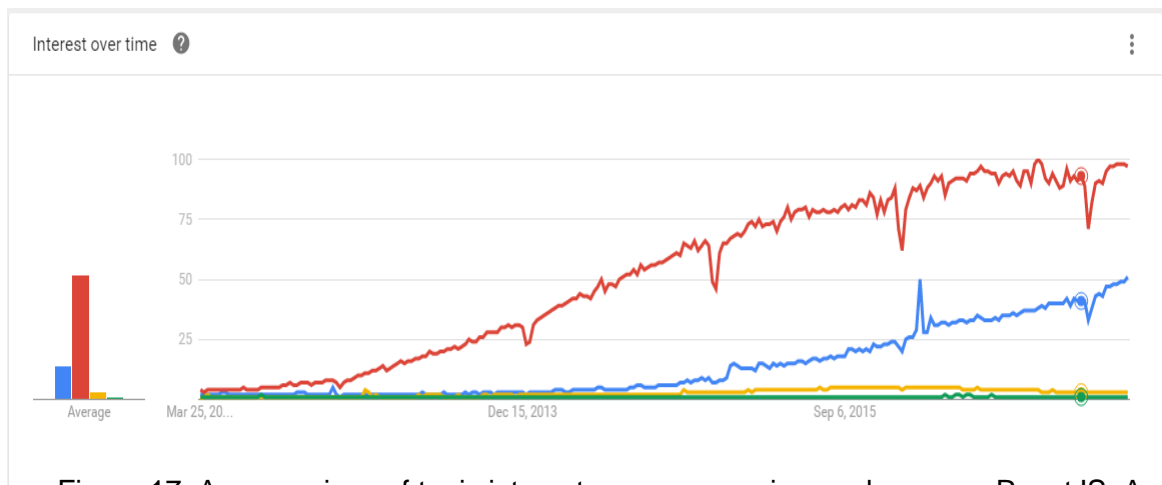


Figure 17: A comparison of topic interest among users in google among ReactJS, AngularJS and VueJS. Copied from Google Trends [10]

As shown in Figure 17, Angular JS is by far the most searched among these three, but ReactJS is quickly closing the gap.

6.3 React Library vs Angular Framework

ReactJS and Angular both are used for front-end web development. Both are very much in tech community for their scalability and compatibility compared to other existing technologies. Both are quite handy when it comes to development as both have many unique features. Both of them provide unique functionalities which are suitable to

consider depending on the nature of the intended application to be built. Hence, a comparison of pros and cons between them is shown in Table 2.

Table 2 Pros and cons between Angular and React. [6]

	React	Angular
Pros	<ul style="list-style-type: none"> • SEO (search engine optimization) friendly • Creating UI test cases is an easy task • Components are reusable • Fast when it comes to displaying the rendered result of a big amount of data in the components • Specialized Chrome extension made debugging easier 	<ul style="list-style-type: none"> • Oriented on the apps with a big amount of interactive client-side code • Good solution for single page dynamic apps • Fast development process • Demands less code in some cases • Advanced testing features • Model-view-controller balance
Cons	<ul style="list-style-type: none"> • In some cases it requires to write more code • Manual processing of data changes is required • View oriented 	<ul style="list-style-type: none"> • Relatively slow when it comes to dealing with big data amounts. • Not as SEO friendly as desired

As shown in Table 2, both of them have advantages and disadvantages. Applications that require frequent data updates involving users should consider ReactJS and big scale dynamic applications may consider AngularJS for development.

7 Data Flow Management in React

An enormous number of web users worldwide are demanding richer, faster and more interactive web applications. The web is a very busy place at this current age. Millions of websites and applications are keeping the web busy all the time with enormous amount of data. To satisfy all users with smoother and smarter services the developers have already come up with many frameworks. However, most of the modern JavaScript frameworks are quite hard to master, sometimes scary and not very easy to maintain. [8]

ReactJS is considered to be less complex, faster, easy to understand and quite handy to maintain the states. [6] It allows us to create re-usable UI components and introduced the virtual DOM instead of error prone manual DOM manipulation. It gives the freedom to design how the user interface should depend upon the state of the data. A minimal number of DOM changes came to reality to make the process faster.

Managing the states of data was always a challenging task to maintain frameworks. It is all about how the frameworks actually manage data between different states and layers in an application describes the simplicity or complexity of the frameworks. [3] If it is really complicated to manage the data between the states during application development, the framework is considered to be hard to master and it takes time to be familiar with it. React can be used with two different architectures for managing the states throughout the application layers. They are Flux and Redux.

7.1 Flux as an Architecture from Facebook

Flux is an architecture developed by Facebook and internally used when working with React. Flux is not a framework nor a library. [13] It is used for creating data layers in JavaScript applications. It is a simple pattern or architecture which allows one way data flow. It reduces the complexity between the UI components in the application. It utilizes an unidirectional one-way data flow that complements React's composable view components.

Flux architecture contains three major parts. They are dispatcher, stores and views. It might sound like the MVC pattern but is not actually similar. It works in a different way

than MVC. Still, Flux does have a controller which is controller-view that retrieves data from the stores and passes them to the children. [13] Figure 18 shows Flux architecture.

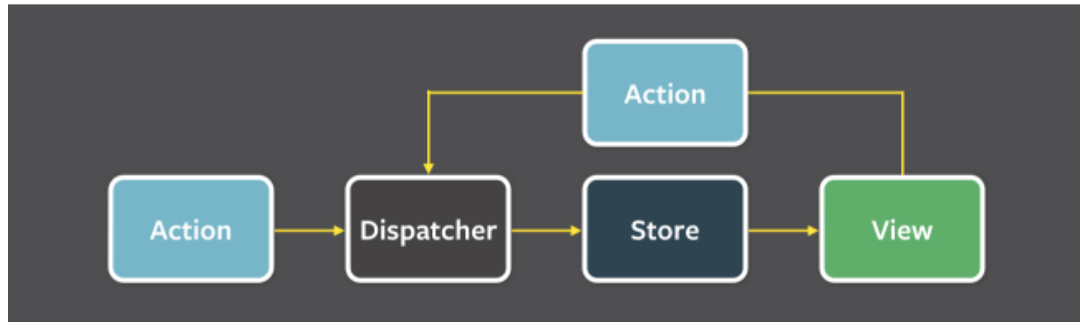


Figure 18: Flux architecture. Copied from ReactJS Cookbook (2017). [13]

As shown in Figure 18, Flux architecture has three main parts, i.e. dispatcher, store and view. Flux can be better understood by explaining the components it contains. Action is a helper method to facilitate the data passage to the dispatcher. Dispatcher receives the actions and then passes the payloads to the callbacks that are registered already. Stores contains the application states and the controller views grab the states from the stores and pass them downwards through the child components. [13]

Finally, Flux is very convenient to be used because the one way data flow makes it easily understandable. Modifying a complicated application is also made possible and much easier with Flux. While two way data binding in other frameworks is entitled to cascading updates and a change in a specific data model brings changes to another model, one-way data flow or binding in React with Flux is a good solution to overcome these inconveniences.

7.2 Redux Data Flow Management

Redux is a conceptual architecture or pattern to manage data and communication between data layers within an application in the browser. Redux is basically inspired by Flux but not a pure Flux implementation. The key difference between Redux and Flux is that Flux uses a single store to wrap all state objects that contain all the states of an application. It plays with data in the client side. It actually works like a backend data-

base in the front-end client-side browser where all the required data are kept in order to generate the view. In a React application those data from the Redux database can be used to generate a proper view depending on the actions made from the user's browser.

Managing those states in an entire application is always a hard and challenging task for the developers. The bigger the scale of the application the bigger is the challenge to maintain states. Here Redux provides the state container to the application so that it behaves consistently. Redux is very useful to build scalable, interactive React applications. It is easier to learn than Flux. It is also very efficient, straight forward and powerful compared to any other solution to manage data flow because it is built with several unique features and is very helpful to make real life applications. It is also a standard tool to work with for the one way data flow featured applications and programs. Redux is very powerful and a must use for the server side React applications. It is a unique and effective technology to be used along with React on the applications or websites where the customer experience is top-notch. For example, Facebook, Instagram, Airbnb and many popular news services around the world use Redux server rendering implementations for quick customer services and responses.

Installation ----- `npm install -save redux`

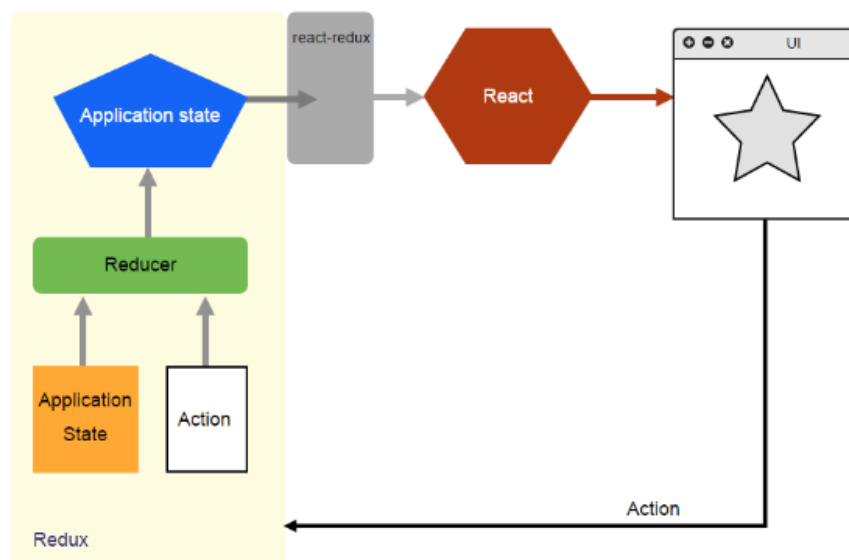


Figure 19: Redux architecture. Printed from startuprocket.com (2017). [14]

As shown in Figure 19, data flows in one way in Redux architecture. Mostly Redux is used as a collection of CommonJs. When Redux is imported in a Webpack, Browserify or in a Node environment these modules are automatically what also downloads. The Redux code is written in ES2015 and works fine in every browser. [14] Using Babel or a module bundler is not mandatory to get started with Redux. The whole state of your app is stored in an object tree inside a single store. The only way to change the state tree is to emit an *action*, an object describing what happened. To specify how the actions transform the state tree, you write pure reducers.

7.3 Which to Choose Between Flux and Redux

Redux is basically an implementation of Flux. Flux is used in Facebook for developments. Flux is neither a library nor a framework. It is a recommended architecture for building web applications. Redux is also not a pure library or framework. Redux is quite like Flux but only reduces some complexity by using functional compositions where Flux architecture is based on callback registers. They are fundamentally almost similar but some abstractions are made less complicated and possible to implement that would be hard or in some cases impossible to be implemented in Flux. Among many great features of Redux is that it is built with tools and there is only one way to do that. Thus, Redux reduces complexity and confusion while developing.

Why Redux should be preferred over Flux is described with some examples.

- Both Redux and Flux both have actions which are comparable to events. Flux considers actions as simple JavaScript object, Redux does so. But if we use middleware with Redux, actions can be functions and promises too.
- Flux has a convention to have multiple stores per application. Redux allows having a single store per application.
- As Flux is a singleton object, it has a single dispatcher and all the actions should pass through the dispatcher. Multiple dispatchers are not allowed in

Flux because it can have multiple stores and a single dispatcher acts as a manager which is required for the dependencies between those stores.

- Redux is free of any dispatcher entity. Instead, stores in Redux have the dispatching capability.
- Stores in Flux decide how to manage the data and show it based on the actions received makes it most powerful in Flux architecture.
- In Redux, the reducer function in the store decides what to do with the data based on the actions received. Reducer functions are the smarter player in Redux.

8 Discussion

In this chapter of this study, a brief evaluation of the topic will be discussed. The intention of this research work over ReactJS library was to explore its features, to find out clearly what it offers, the core concepts behind the architecture, how it is different from other library and frameworks, its data handling processes and other functionalities. The reason behind choosing ReactJS as a topic was to know it better by researching and thus enriching personal learning over it.

Throughout the research, it was found that ReactJS is a less complicated library compared to other frameworks. It is a very fast rendering library and agile in development. Learning and mastering React is less time consuming compared to others since it is only a library, not a framework. Frameworks usually take a longer time to know about them and learn the terminologies.

In addition, the main concept behind React is components. Everything shown in the front-end is nothing but components. The user-interface is a collection of components. Making changes in one component does not affect other components. Thus updating the application over changes in data has been made easier and less complicated. To perform this functionality React introduces a modified concept of DOM to virtual DOM.

One-way data flow for the data management is another good thing about React. Data can be changed from anywhere in the application layer. As the data flows in a single direction, it offers solid control between the models and states in the application. The inclusion of a new syntax called JSX has been proved quite fast while rendering.

As a faster rendering library, React has been proved successful since applications like Facebook, Instagram, Airbnb, Netflix, Uber have adopted it for their developments. When it comes to dealing with enormous amounts of users and data, React is undoubtedly adoptable. All those applications mentioned above have billions of users and countless data to be dealt with. Thus, when an application intended to serve big numbers of users, play with tons of data, ReactJS is recommended to be used for faster and smooth user experiences.

9 Conclusion

The goal of the thesis work was to study and review the JavaScript based open source front-end library called ReactJS. Facebook developed ReactJS for their own purpose and later open sourced it. Since the beginning, within a very short time ReactJS has gained immense popularity among the developers and tech industry.

To conclude, in this document, a clear instruction has been illustrated of how to start with React, React features and functionalities with examples, when to consider React over other alternatives and what data architecture management system to consider as well as its prospects in detail.

Since, React is a demanding and important technology to be learnt, learning through further research would be beneficial to enrich personal skills over this technology. Keeping this in mind, the topic was chosen. Developing an application in React would be great in terms of enriching practical skills, but due to some limitations, an in depth review and evaluation was produced. So far, a solid concept has been built already while studying over React for the last few months.

Finally, it can be said that ReactJS is quite a technology to be learnt and certainly worth considering for application in production. It has brought a new dimension in the web application development. The fast rendering library accelerates the application efficiency and it can be seen that React has a bright future ahead and learning React is worth the effort.

References

1. A JavaScript Library for Building User Interfaces [Online]
URL: <https://facebook.github.io/React/>. Accessed February 5 2017
2. React Native Bringing Modern Web Techniques to mobile.
URL: <https://code.facebook.com/posts/1014532261909640/React-native-bringing-modern-web-techniques-to-mobile/>. Accessed February 9 2017
3. Github React Native [A framework for building native apps using React native] [Online] URL: <https://facebook.github.io/React-native/>. Accessed April 22 2017
4. React Native Mobile Apps
<https://www.smashingmagazine.com/2016/04/consider-React-native-mobile-app/>
Accessed April 25 2017
5. React native Animation and navigation. [Online]
URL: <https://facebook.github.io/React-native/docs/animated.html>.
6. Stefanov Stoyan, editor. React: Up and Running: Building web Applications. First Edition; 2016. Accessed April 4 2017
7. Horton Adam. Vice Ryan, author. Mastering React; February 23; 2016.
Accessed April 8 2017
8. Stein Johannes, author. ReactJS Cookbook. December 6; 2017.
Accessed March 22 2017
9. Masiello Eric, author. Mastering React Native. January 11; 2017.
Accessed April 30 2017
10. Google Trends. [Online][Cited: 26 March 2017]

URL: <https://trends.google.com/trends/explore?q=%2Fm%2F012l1vxv,%2Fm%2F0j45p7w,VueJS>

Accessed April 1 2017

11. A comparison of React and Angular [Online]

URL: <https://erminesoft.com/what-are-the-pros-and-cons-of-angularjs-vs-Reactjs/>

Accessed April 12 2017

12. React Virtual DOM [online]

URL: <http://stackoverflow.com/questions/21109361/why-is-React-s-concept-of-virtual-dom-said-to-be-more-performant-than-dirty-mode>

Accessed April 2 2017

13. Flux: Application Architecture for Building User Interfaces. [Online]

URL: <https://facebook.github.io/flux/docs/in-depth-overview.html>

Accessed April 4 2017

14. Evolution Towards One- Way Data Flow: A Quick Introduction to Redux.

URL: <https://www.startuprocket.com/articles/evolution-toward-one-way-data-flow-a-quick-introduction-to-redux>

Accessed April 11 2017

15. What the Flux! Lets Redux. . [Online]

URL: <https://blog.andyet.com/2015/08/06/what-the-flux-lets-redux/>

Accessed April 16 2017

