React Introduction

Sunday, June 21, 2020   12:19 PM

What is React?

React is a JavaScript library created by Facebook
React is a User Interface (UI) library
React is a tool for building UI components
React elements are immutable. They cannot be changed.
The only way to change a React element is to render a new element every time

*** Used by companies like Uber, Netflix, Twitter

What is Babel?

Babel is a JavaScript compiler that can translate markup or programming languages into JavaScript.
With Babel, you can use the newest features of JavaScript (ES6 - ECMAScript 2015).
Babel is available for different conversions. React uses Babel to convert JSX into JavaScript.
Please note that <script type="text/babel"> is needed for using Babel.

What is JSX?

JSX stands for JavaScript XML.
JSX is an XML/HTML like extension to JavaScript.

Example

```
const element = <h1>Hello World!</h1>
```

React DOM Render

The method ReactDom.render() is used to render (display) HTML elements:

Example

```
<div id="id01">Hello World!</div>
<script type="text/babel">
        ReactDOM.render(
           <h1>Hello React!</h1>,
           document.getElementById('id01')
        );
</script>
```

JSX Expressions

Expressions can be used in JSX by wrapping them in curly {} braces.

Example

```
<div id="id01">Hello World!</div>
<script type="text/babel">
        const name = 'John Doe';
        ReactDOM.render(
           <h1>Hello {name}!</h1>,
           document.getElementById('id01')
        );
</script>
```

React Components

React components are JavaScript functions.
This example creates a React component named "Welcome":

Example

```
function Welcome(props) {
    return <h1>Hello {props.name}!</h1>;
}
ReactDOM.render(<Welcome name="John Doe"/>, document.getElementById('root'));
```

React can also use ES6 classes to create components.
This example creates a React component named Welcome with a render method:
Example

```
class Welcome extends React.Component {
    render() { return(<h1>Hello {this.props.name}</h1>); }
}
ReactDOM.render(<Welcome name="John Doe"/>, document.getElementById('root'));
```

Setting up dev env

Sunday, June 21, 2020 12:56 PM

https://github.com/facebook/create-react-app

- npm install -g create-react-app // install create-react-app package globally, run from any folder, need to do once
- create-react-app react-is-fun // create an app with name react-is-fun in current folder
- cd react-is-fun // change to app directory
- code . // to open VS Code
- npm start // opens the app in browser
- npm install prop-types --save // install prop-types package
- npm run build // build the project for production deployment
- npm install serve -g // install package serve
- serve -s build // serve with a static server

React.js Building and Interface

Tuesday, June 23, 2020 9:39 PM

Exercise files:
https://github.com/planetoftheweb/reactinterface2
https://gist.github.com/planetoftheweb

Adding your own modules

- npm i -s bootstrap react-icons lodash
- npm i -s jquery popper.js moment react-moment

Include imports in index.js

```
import 'bootstrap/dist/css/bootstrap.css';
import './index.css';
import 'jquery/dist/jquery.js';
import 'popper.js/dist/umd/popper.s'
import 'bootstrap/dist/js/bootstrap.js';
```

Creating and Hosting a Full-Stack site

Monday, July 6, 2020 1:02 PM

Building navigation:

- npm install --save react-router-dom

Setting up an Express server

- Create a project directory separate from UI to be used for all backend related code i.e. react-is-fun-backend
- cd react-is-fun-backend
- npm init -y
- npm install --save express
- npm install --save-dev @babel/node @babel/preset-env
- npm install --save-dev @babel/core @babel/preset-env

Then create a **.babelrc** file in project directory (react-is-fun-backend) with content

```
{
    "presets": ["@babel/preset-env"]
}
```

- npx babel-node src/server.js --> Start the express server

Download Postman app to test the get/post calls

- npm install --save body-parser

Automatically updating with nodemon

- npm install --save-dev nodemon
- npx nodemon --exec npx babel-node src/server.js

Add an alias **"start"** in package.json so that you can start server simply by issuing **npm start** command

```
"scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "npx nodemon --exec npx babel-node src/server.js"
  },
```

Setting up MongoDB

Wednesday, July 8, 2020 6:49 PM

Nonrelational database:

- Can push data to database without worrying about format (in other words, accepts any JSON object)
- Structure of data does not have to be defined in advance
- SQL not required
- Allows for creation of modular, reusable components that can be arranged into a fully-functioning site

Installation steps for windows:
https://docs.mongodb.com/manual/tutorial/install-mongodb-on-windows/

Create database directory.

Create the data directory where MongoDB stores data. MongoDB's default data directory path is the absolute path \data\db on the drive from which you start MongoDB.

From the **Command Interpreter**, create the data directories:

```
cd C:\
md "\data\db"
```

Start your MongoDB database.

To start MongoDB, run mongod.exe.

```
"C:\Program Files\MongoDB\Server\4.2\bin\mongod.exe" --dbpath="c:\data\db"
```

The --dbpath option points to your database directory.

If the MongoDB database server is running correctly, the **Command Interpreter** displays:

```
[initandlisten] waiting for connections
```

To start MongoDB

- cd "C:\Program Files\MongoDB\Server\4.2\bin>"
- .\mongod.exe

In another terminal:

- cd "C:\Program Files\MongoDB\Server\4.2\bin>"

- .\mongo.exe

- db.use my-blog //my-blog would be the name of the database

-
  ```
   db.articles.insert([ {
     name: 'learn-react',
          upvotes: 0,
          comments: []
  }, {
     name: 'learn-node',
          upvotes: 0,
          comments: []
  }, {
     name:'my-thoughts-on-resumes',
          upvotes: 0,
          comments: []
    }
  ])
  ```
- db.articles.find({})
- db.articles.remove({}) //to delete
- db.articles.find({}).pretty()
- db.articles.find({name:'learn-react'}).pretty()
- db.articles.findOne({name:'learn-react'})

<u>Adding MongoDB to Express:</u>

- npm install --save mongodb

Connecting front-end and back-end

Thursday, July 9, 2020 9:17 AM

The Fetch API:

- npm install --save whatwg-fetch

Hosting the site

Thursday, July 9, 2020 11:40 AM

Build front-end folder and move build folder to back-end folder' src
Initialize an empty git repo using

- git init
- git status
- git add .
- git commit -m "First Commit"
- git status

Push to github

- git remote add origin https://github.com/yash-bansal/my-blog.git
- git push -u origin master

Creating and SSHing into an AWS instance

1. Login to aws.amazon.com
2. Go to AWS Management Console
3. All Services > EC2
4. Launch Instance
5. Select Amazon Linux 2 AMI (HVM), SSD Volume Type
6. Review and Launch
7. Launch
8. Create a new key pair, download key pair (.pem file)
9. Launch Instances
10. View Instances (after sometime)

- Move .pem file to .ssh folder
- Create folder using C:\> mkdir .ssh and copy file into .ssh folder
- Open .pem file properties
- Go to Security > Advanced
- Disable inheritance
- Remove all users except yourself > Apply
- Run below command as admin
  - ssh -i C:/.ssh/my-blog-key.pem ec2-user@ec2-3-16-40-8.us-east-2.compute.amazonaws.com
- After successful SSH
  - $ sudo yum install git
  - y
  - curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash [Ref: --> https://docs.aws.amazon.com/sdk-for-javascript/v2/developer-guide/setting-up-node-on-ec2-instance.html]
  - . ~/.nvm/nvm.sh
  - nvm install 10.20.1
  - npm install -g npm@latest
- Installing MongoDB on AWS (https://docs.mongodb.com/manual/tutorial/install-mongodb-on-amazon/ | Amazon Linux 2)
  - sudo nano /etc/yum.repos.d/mongodb-org-4.2.repo --> Creates a file

    Paste below in file

    ```
    [mongodb-org-4.2]
    name=MongoDB Repository
    baseurl=https://repo.mongodb.org/yum/amazon/2/mongodb-org/4.2/x86_64/
    gpgcheck=1
    enabled=1
    gpgkey=https://www.mongodb.org/static/pgp/server-4.2.asc
    ```

    And save and exit using ctrl+ O, ctrl + X
  - sudo yum install -y mongodb-org
  - sudo service mongod start
  - Mongo
  - use my-blog
  - db.articles.insert([{
      name: 'learn-react',

```
        upvotes: 0,
        comments: []
    }, {
      name: 'learn-node',
        upvotes: 0,
        comments: []
    }, {
      name:'my-thoughts-on-resumes',
        upvotes: 0,
        comments: []
      }
    ])
```

- Clone Git repo
  - git clone https://github.com/yash-bansal/my-blog.git
  - ls
  - cd my-blog
  - npm instal
  - npm install -g forever
  - forever start -c "npm start" .
  - forever list
  - sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-ports 8000
- Go to AWS console
- Check security group of instance
- Go to Security groups
- Select your security group
- Inbound rule
- Edit inbound rules
- Change to HTTP and Anywhere in 2nd rule and Save rules
- Go to instances and copy public dns url --> ec2-3-16-40-8.us-east-2.compute.amazonaws.com

AWS resource teardown

- Go to instance
- Select you instance
- From Actions select Instance State > Terminate

Important links

Monday, June 22, 2020 6:47 PM

https://codesandbox.io/

https://create-react-app.dev/docs/making-a-progressive-web-app/