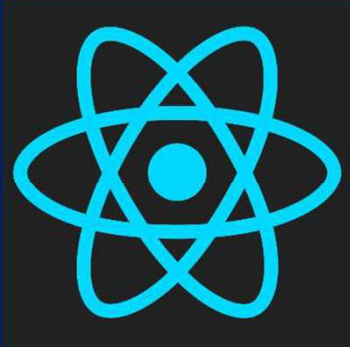
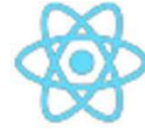


13

React JS

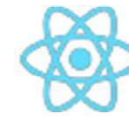


Taking To Server & Testing



Fetching Data – AJAX Requests

- React itself does not have any bias towards a particular approach of fetching data.
- React may not even know if there is a server in the picture!
- React simply renders components using data from two places: props and state.
- In order to use some data from a server that data needs to be in the component's prop or state.



A HTTP library

- To fetch data from the server a HTTP library is required.
- There are multiple libraries available:
 - Use promises? Go with axios
 - Use callbacks? Check superagent
 - Browser based window.fetch()
- *There are competing libraries and there is no “best”*
- It is **believed** that Axios and React pair nicely together.
- Install `axios` using `npm`

```
npm install axios --save
```

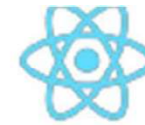
Using Axios

- Axios offers methods for all the HTTP verbs (GET, POST, PUT, DELETE).
- It performs automatic JSON data transformation.

```
import axios from 'axios';

state = {
  Post : ''
}

axios.get('http://jsonplaceholder.typicode.com/posts/2')
  .then(resp => {
    this.setState({Post:resp.data.title})
  })
```



Using Axios...

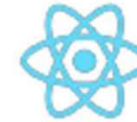
- Axios helps transform the request and response data.

- GET

```
axios.get('/employee?eid=123')  
  .then(function (response)  
    { // handle success})
```

- POST

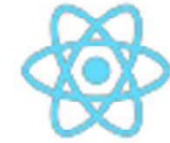
```
axios.post('/employee', { firstName: 'Jack', lastName: 'Straw'  
  })  
  .then(function (response) {  
    // todo})
```



Services in React

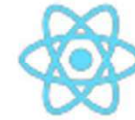
- React as a library does not explicitly advocate the use of services.
- A “service” in react can essentially be regarded as a collection of helper functions.

```
const DataService = {  
  firstMethod: function(value)  
    { //to do},  
  secondMethod: function(value)  
    { //to do }  
};  
export default DataService;
```

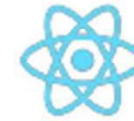
Summary

- AJAX allows fetching of data and update the UI without refreshing the page.
- Client side frameworks incorporate services to make HTTP calls.
- React does not have any built-in capability for making AJAX calls.
- Libraries like React rely on other libraries to facilitate the same.



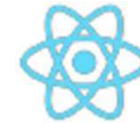
Testing

- **Unit Testing**
 - Unit testing refers to testing individual pieces of code.
 - In React, Unit tests typically do not require a browser.
- **Functional Testing**
 - This involves testing the behavior of a component.
 - Functional tests usually run in isolation.
- **Integration Testing**
 - This tests the entire application and attempts to replicate an experience the end-user would have when using the application.



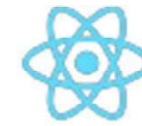
Testing - Tools

- Jest
 - Introduced by Facebook for component tests in React.
 - Jest is not only a testing library but also a test runner.
- Enzyme
 - Enzyme was introduced by Airbnb for component tests in React
 - Enzyme is used to render components, access elements, props etc and simulate events.
- Storybook
 - This can be used to render components in their different states.
 - Can be used to ensure components behave correctly in their different states



Testing

- `create-react-app` application comes with Jest as a test runner and assertion library.
 - It is the *official* test runner
- Jest includes all the goodness of Jasmine with its own improvements.
- Jest supports TypeScript and contains most of the functionality in one package and is the best option for React application testing.
- Enzyme is popular for its React testing utility functions that simplify writing assertions.



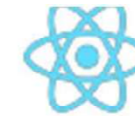
Jest Basics

- *App.test.js*

```
describe('Addition', () => {  
  it('knows that 2 and 2 make 4', () =>  
    { expect(2 + 2).toBe(4);  
    });  
});
```

- The test can be run using

```
npm test
```



Enzyme

- *Enzyme* is built to support different versions of React.
- *Enzyme* introduced adapters to gel well with React.
 - An adapter appropriate to the version of React being used is to be installed.

```
npm install --save-dev enzyme
```

```
npm install --save-dev enzyme-adapter-react-16
```

- `setupTests.js` file tells *Jest* and *Enzyme* the adapters that will be used

```
import { configure } from 'enzyme';
```

```
import Adapter from 'enzyme-adapter-react-16';
```

```
configure({ adapter: new Adapter() });
```

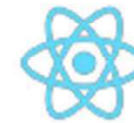


Enzyme

- App.test.js

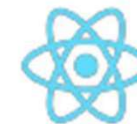
```
import React from 'react';
import { shallow } from 'enzyme';
import App from './App';

describe('App component', () => {
  it('has text Hello World', () => {
    const wrapper = shallow(<App />);
    const text = wrapper.find('p').text();
    expect(text).toEqual('Hello World');
  });
});
```

Summary

- Writing tests is an essential part of software development.
- React components are very *testable*
- With *Jest* and *Enzyme* developers can run declarations by gaining access to properties, state and child props of React components.
- *Jest* tooling is developer-friendly and intuitive.



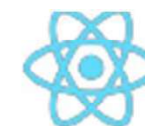
Build

`npm run build`

- Creates a `build` directory with a production build of the application.
- Copy the files in the `build` directory to the target server
- The HTTP server can be setup such that clients are served `index.htm`
- Delete the following files to disable debugging support.

`build/static/js/some_file.js.map`

`build/static/css/some_file.css.map`

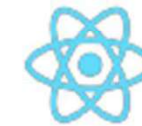


Building for relative paths

- Create React App produces a build that assumes that the application is hosted at the server root.
- To override this the “homepage” property can be specified in `package.json`

“homepage” : `http://mysite.com/relativepath`

- This will allow Create React App to infer the root path correctly for use in the HTML file.



Session Summary

- React is a JavaScript **library** for building user interfaces.
- Declarative views make code predictable.
- React helps build encapsulated components that manage their own state.
- React uses one-way data binding which makes code more predictable and simplifies debugging.
- React makes it faster and easier to build client facing web interfaces leveraging newer JavaScript features.



Resources

- Awesome React - <https://github.com/enaqx/awesome-react>
 - A collection of awesome things regarding React ecosystem
- React components - <https://github.com/brillout/awesome-react-components>
 - A collection of react components
- Books
 - Learn ReactJS fast
 - React in action
 - Learning Web Development with React and Bootstrap