

ITMD 465/565

Rich Internet Applications

Lecture 13

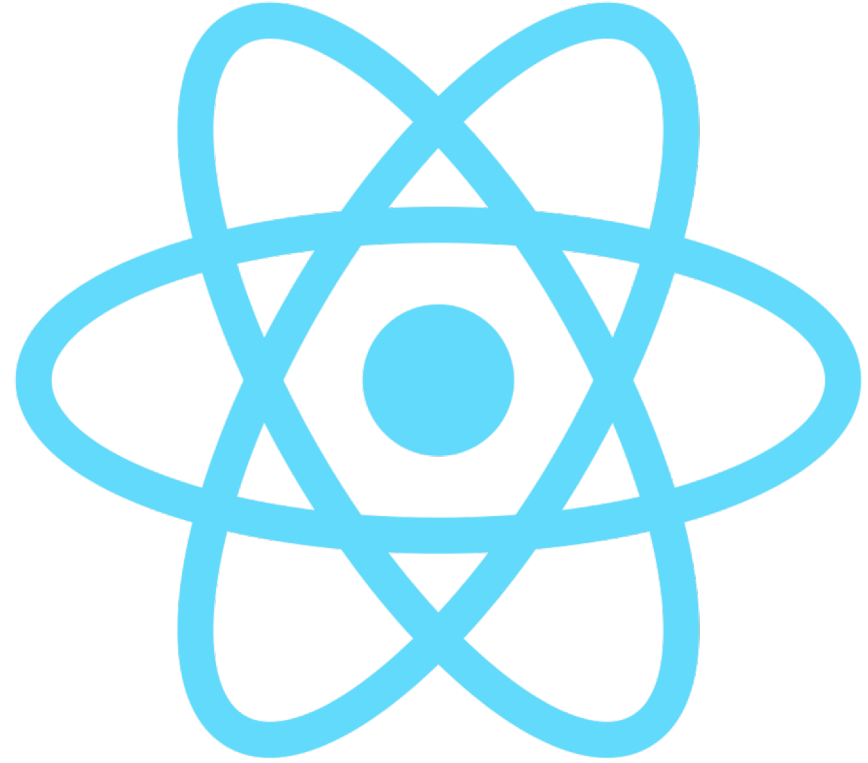
Fall 2019 – November 13, 2019

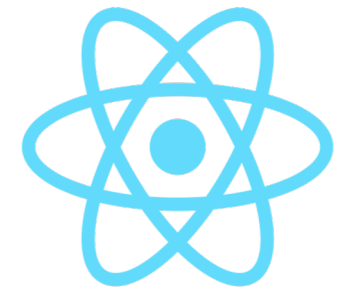
Tonight's Agenda

- Finish Class 12 Slides and Demo including
- React State Management review
- React Lifecycle events review
- Basic webpack and babel project setup and using modules
- Building React Components as modules
- Create React App tool
- React Todo App Demo

ReactJS

JavaScript library for building user interfaces



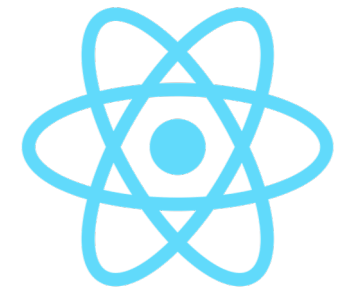


React JS – State

- State is place we can store data in a component and change the data to see a reflection in the UI.
- It is available in the `this.state` object
- Must add a constructor to your class and initialize the state there.

```
class App extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = { header: "Header from state...", content: "Content from state..." }  
  }  
  render() {  
    return ( <div> <h1>{this.state.header}</h1> <h2>{this.state.content}</h2> </div> );  
  }  
}
```

- Try to make the state as simple as possible and keep as many components stateless as possible. If you have many components that need state you should make a parent component that has the state in it and pass it down through props.



React JS – State

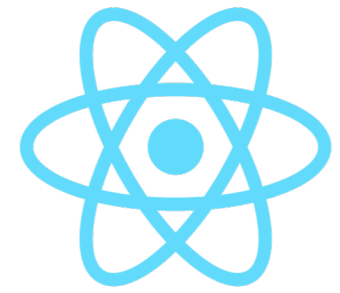
- Do not try to modify the state object directly

```
// Wrong  
this.state.comment = 'Hello';
```

```
// Correct  
this.setState( { comment: 'Hello' } );
```

Or `this.setState((currentState) => { return new state});`

- If you need to set state in one component from another you need to pass handlers down to the child component that then calls `setState()` in the parent that has the state object.
- The React library watches this state and when it detects changes it compares it to the browser DOM and updates only what is necessary
- <https://reactjs.org/docs/state-and-lifecycle.html>



React JS – Lifecycle

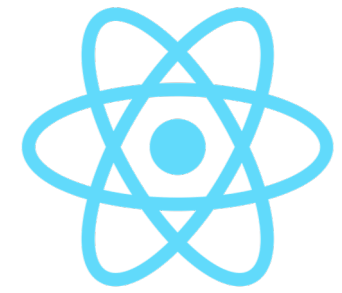
- React class components have lifecycle events you can use.
- We will look at a few of the most important ones.
- Some older ones are marked as not safe and will be removed soon.
- <https://reactjs.org/docs/react-component.html#componentdidupdate>
- `componentDidMount()`
 - Called immediately after the component is rendered to the DOM
 - This is good for initialization that requires the DOM nodes or network request.
- `componentWillUnmount()`
 - Called before the component is removed from the DOM
 - Used to clean up timers, subscriptions, and other items to keep them from running
- `componentDidUpdate()`
 - Called immediately after updating occurs. Opportunity for updated network requests but make sure they are necessary by comparing `prevProps` and `prevState` to current.

Demo

- Lifestyle & leak example

Helpful Array Functions

- Here are a few array functions you should know. The important part here is they work in a functional way. They do not modify the input but instead return a new array. Map is especially helpful when outputting JSX from arrays.
- `map()`
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map
- `filter()`
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter
- `reduce()`
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/Reduce
- `concat()`
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/concat



React JS and Babel

- Babel is used to convert JSX to plain JavaScript React calls.
- Babel can also compile our ES6 to ES5
- Babel can be directly used in the browser for development purposes

```
<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>
<!-- Your custom script here -->
<script type="text/babel">
  const getMessage = () => "Hello World";
  document.getElementById('output').innerHTML = getMessage();
</script>
```

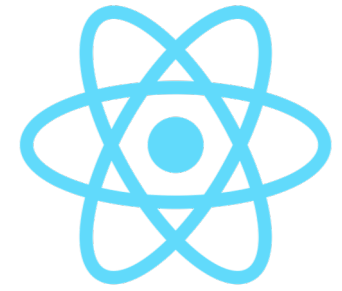
- Need to configure the babel presets or it won't do anything. Typically in a .babelrc file.
- Need to load modules for the presets you want to use and then add them to the .babelrc
- <https://babeljs.io/docs/setup/#installation>

React JS and bundling

- In an application that is all written in React it is common to use a bundler tool like webpack or browserify
- Browserify - <http://browserify.org/>
- Webpack - <https://webpack.js.org/>
- These tools take all the JavaScript files and combine them into a single file to load on your app.
- Browsers can not require or import other JavaScript files so this traces down all the imports/requires and in-lines them into the final script.
- Each has a plugin ecosystem to do many other things too, like minified output
- We will look at webpack

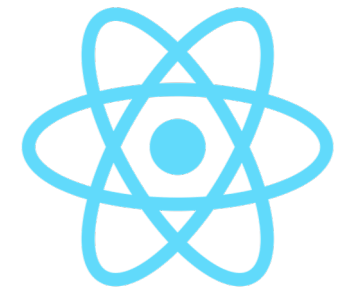
Webpack, Babel, React Setup

- We will look at setting up webpack, babel, and react ourselves before looking at the scaffolding tool.
- Here is a web tutorial we will work through.
- <https://www.valentinog.com/blog/react-webpack-babel/>
- Here is an additional Webpack 4 tutorial
- <https://www.valentinog.com/blog/webpack-tutorial/>



Create React App

- Handy CLI that will completely scaffold a React app with webpack, babel, Jest testing and scripts to build and run a dev server.
- Created by the Facebook team
- <https://github.com/facebook/create-react-app>
- Recommend usage is now with npx so you don't have to install it globally.
- `npx create-react-app my-app`
- Can use npm or yarn to run the scripts defined in the package.json, defaults to yarn



React JS, Webpack, and Babel

- <https://babeljs.io/docs/setup/#installation>
- <https://webpack.js.org/concepts/>
- <https://webpack.js.org/configuration/dev-server/#devserver>
- <https://www.valentinog.com/blog/webpack-tutorial/>
- <https://www.valentinog.com/blog/react-webpack-babel/>

Demo

- React Todo List Demo

Assignments

Reading/Assignments

- First React Lab building simple components is in blackboard and Due next week.
- You will get another final Lab posted before next class and it will be due by the last regular day of class before finals week.
- Quiz 3 is ready in blackboard and I will make it available after class tonight. Due before next class on Nov 20 at 6:30pm.