

LEARNING FACEBOOK'S



React Essential Features and Syntax

Lesson 02

Lesson Objectives

At the end of this module you will be able to:

- React App Project Directory Structure
- Overview of Webpack, Babel
- React Component Basic
- Create React Component
- Understanding JSX
- Limitations of JSX
- Working with Components and Reusing Components

LEARNING FACEBOOK'S



React.js



React App Project Directory Structure

Folder Structure :

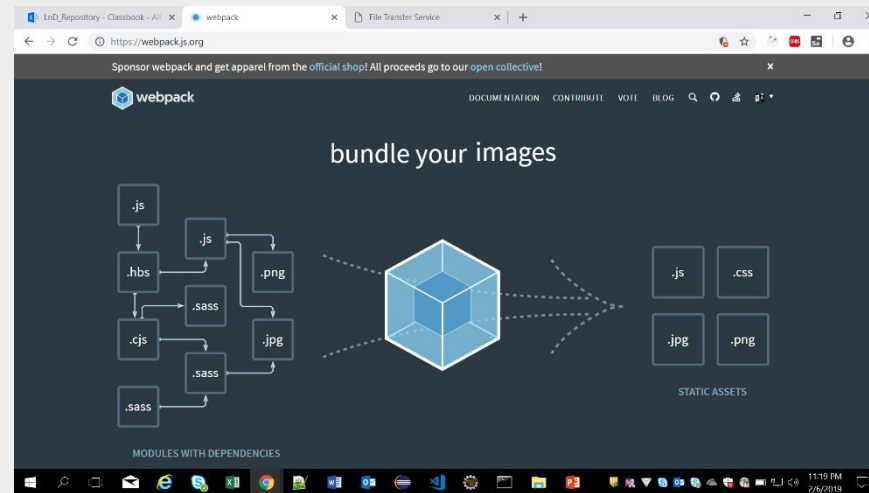
After creation, your project should look like this:

```
my-app/  
  README.md  
  node_modules/  
  package.json  
  public/  
    index.html  
    favicon.ico  
  src/  
    App.css  
    App.js  
    App.test.js  
    index.css  
    index.js  
    logo.svg
```

Overview of Webpack, Babel



- Webpack :
 - webpack is a static module bundler for modern JavaScript applications.
 - Extended to support many different assets such as images, fonts and stylesheets.



- Babel :
 - **Babel is a JavaScript compiler**
 - Babel is a toolchain that is mainly used to convert ECMAScript 2015+ code into a backwards compatible version of JavaScript in current and older browsers or environments.

React Component Basic



Components:

Components let you split the UI into independent, reusable pieces, and think about each piece in isolation.

Conceptually, components are like JavaScript functions.

Every component follows this basic structure:

```
import React, { Component } from 'react';
class componentName extends Component {
  render() {
    return (
      <div>

      </div>
    );
  }
}

export default componentName;
```

Create React Component



We can Create a new React Project using the create-react-app command.

Install create-react-app using npm by using below command

```
npm i create-react-app
```

Then you create your own component and react app by using below command

```
create-react-app my-app
```

JSX (JavaScript and XML)



- is JavaScript syntax extension which looks similar to XML
- looks like HTML but is actually a mix of JavaScript and HTML.

```
var grp = <div>
  <p>Welcome to TutsPlus</p>
</div>;
```

➤ Advantages of JSX:

- It's easier & faster to write JSX, compared to JavaScript.
- JSX code ensures readability & Maintainability.
- JSX finds most of the errors at compilation time, which makes it faster Comparitively to javascript

Babel would transform the above JSX code to the required react function as shown:

```
var grp =
React.createElement( "div", null, React.createElement(
  "p", null, "Welcome to TutsPlus" ) );
```



Limitations of JSX:

- complexity
- consequent steep learning curve

Working with Components and Reusing Components



- React lets you define components as classes or functions.
- Components defined as classes currently provide more features which are described in detail on this page.
- Component class must extend `React.Component`

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, Welcome</h1>;  
  }  
}
```



Summary

- React App Project Directory Structure
- Overview of Webpack, Babel
- React Component Basic
- Create React Component
- Understanding JSX
- Limitations of JSX
- Working with Components and Reusing Components





1. JSX is a shorthand for

- A) Javascript & XML
- B) XML and java
- C) Javascript & Java
- D) HTML & Javascript

2. To Enable a browser to read JSX, first, we need to transform JSX file into a JavaScript object using JSX transformers like?

- A) Babel
- B) Babel Transformer
- C) Gulp
- D) React Compiler

3. Search and identify why React Js is faster than other UI Frameworks