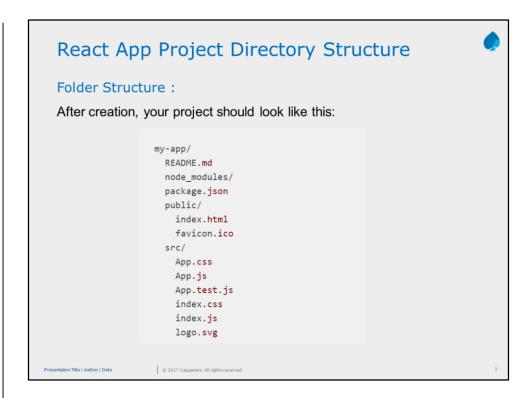
Add instructor notes here.



Add instructor notes here.



Slide explains that communication between service provider and consumer happen via SOAP messages



For the project to build, these files must exist with exact filenames:

- •public/index.html is the page template;
- •src/index.js is the JavaScript entry point.

You can delete or rename the other files.

The slide explains in brief, the components which make up the web service

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.

Presentation Title | Author | Dat

0 2017 Capgemini, All rights reserve

webpack cares about performance and load times; it's always improving or adding new features, such as async chunk loading and prefetching, to deliver the best possible experience for your project and your users.

Why webpack

To understand why you should use webpack let's recap how we used JavaScript on the web before bundlers were a thing.

There are two ways to run JavaScript in a browser. First, include a script for each functionality; this solution is hard to scale because loading too many scripts can cause a network bottleneck. The second option is to use a big .js file containing all your project code, but this leads to problems in scope, size, readability and maintainability.

Here are the main things Babel can do for you:

Transform syntax

Polyfill features that are missing in your target environment (through @babel/polyfill)

Source code transformations (codemods)

Slide explains about WSDL and UDDI registry.

Also explains regarding web service communication

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; © 2017 Capgemini. All rights reserved

every React component is a class. This class extends the Component class from the react module, and as we said before, we need to import React from react in order to get our component to work. Inside the render() method, we are going to return only one element, it's common that we wrap everything with a big <div>. Inside the component file we can define several classes, but we specify which one is going to be exported as default with the last line. I just added some information to my TestComponent, and style with some CSS:

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

You can also use an <u>ES6 class</u> to define a component: class Welcome extends React.Component { render() { return <h1>Hello, {this.props.name}</h1>; } }

Add instructor notes here.

Create React Component

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

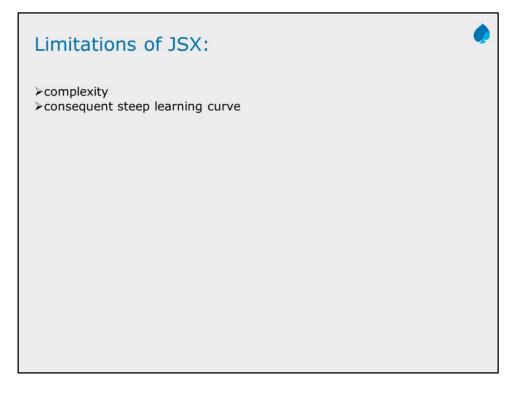
Install create-react-app using npm by using below command

npm i create-react-app

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

create-react-app my-app

Babel is nothing but Transpiler, Babel can convert JSX syntax and strip out type annotations.



Working with Components and Reusing Components



- >React lets you define components as classes or functions.
- >To define a React component class, it must extend React.Component

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

Reusable Components

➤ When designing interfaces, break down the common design elements (buttons, form fields, layout components, etc.) into reusable components with well-defined interfaces.

>That way, the next time you need to build some UI, we can write much less code.

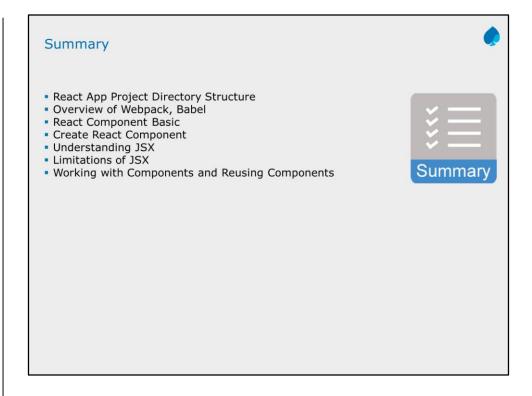
> This means faster development time, fewer bugs, and fewer bytes down the wire.

Reusable components are like in above slide Welcome is a reusable component

```
class Greeting extends React.Component {
render(){
return <h1>Hi, I'm a smart component!</h1>;
}
}
```

In above code greetings is a reusable

Add instructor notes here.



Add the notes here.

review

- 1. JSX is a shorthand for
 - A) Javasctipt & 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