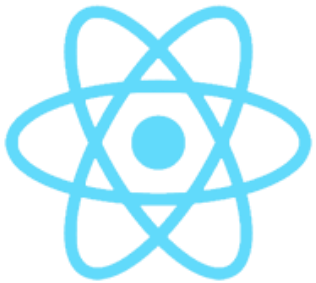
YASH Technologies™

Overview



React

Presented by : Pankaj Sharma

Topics to be covered

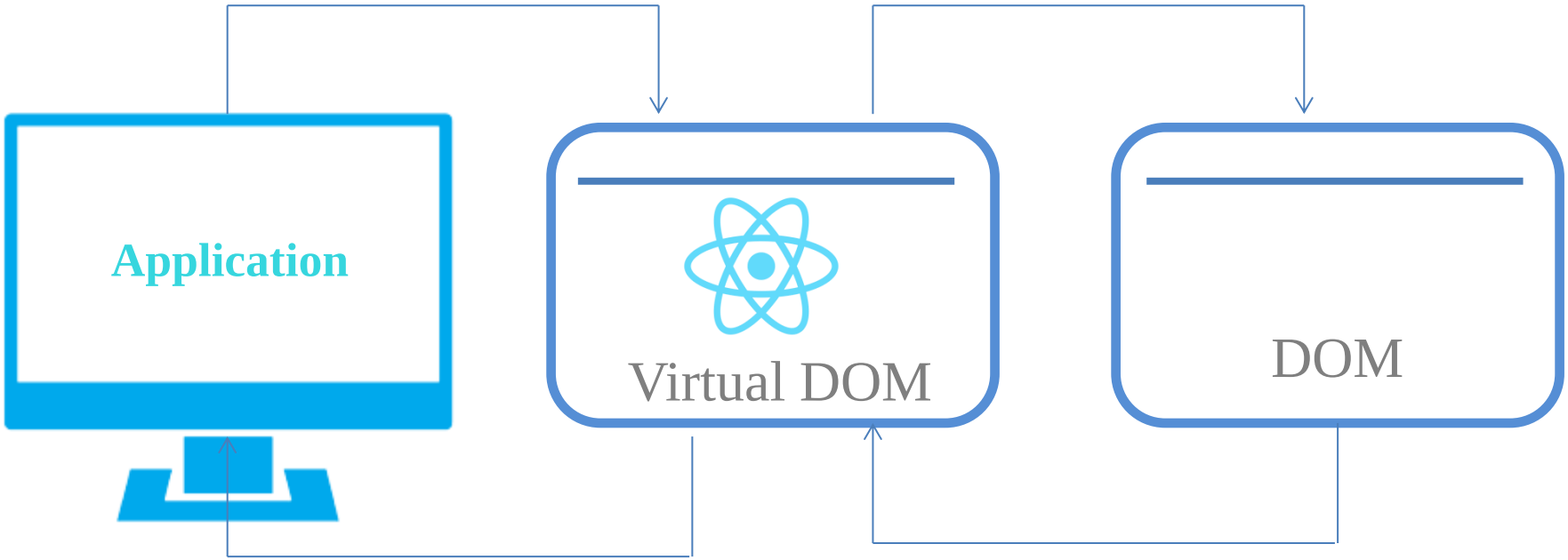
- What is ReactJS
- Virtual DOM
- Simple Hello World Application – React Simulator – Approach-1
- Simple Hello World Application – React Simulator – Approach-2
- Simple Hello World Application of ReactJS – Approach-3
- Use of Babel
- Simple Hello World Application of ReactJS – Approach-4 – use of babel
- Simple Hello World Application of ReactJS – Approach-5 – Component
- Simple Hello World Application of ReactJS – Approach-6 – Component Injection
- Bootstrap HelloWorld Application in ReactJS

What is REACTJS

Is it a framework?

- It is a view in the application
- Break down pages, or features in the small pieces of components
- Breaking down the features in the small pieces will offer the Reusability
- It is Declarative Library rather than Imperative
- It has a virtual DOM feature, which makes the DOM rendering really fast
- Component Based model

Working of ReactJS



[Small Movie on React]

Simple Hello World Application – React Simulator – Approach-1

You need to create a sample application to simulate DOM.render()?

- Challenges are:
  - Not to include any dependency of ReactJS
  - Use JavaScript to achieve the functionality

Simple Hello World Application – React Simulator – Approach-1

HTML Template:

```
Day1/withoutReact-01.html
-------------------------------------------------
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
<h1>Dom Manipulation ! Without React</h1><hr>
<div id="root">
<!--Here Dom will populate the data-->
</div>
<script type="text/javascript">
    // Add the code here
</script>
</body>
</html>
```

Simple Hello World Application – React Simulator – Approach-1

Solution:

Day1/withoutReact-01.html

```
<script type="text/javascript">
    var whatToRender="Hello World without ReactJS";
    var whereToRender="root";
    var render=function(whatToRender,whereToRender){
         document.getElementById(whereToRender).innerHTML=whatToRender;
    }
    render(whatToRender,whereToRender);
</script>
```

Simple Hello World Application – React Simulator – Approach-2

Now convert your application in dom.render()

Solution:

```
Day1/withoutReact-02.html
-------------------------------------------------
// HTML Code will be as it is
<script type="text/javascript">
    var dom={
      whatToRender:'Hello World without ReactJS',
      whereToRender:"root",
      render:function(whatToRender,whereToRender){
         document.getElementById(whereToRender).innerHTML=whatToRender;
      }
    };

    dom.render(dom.whatToRender,dom.whereToRender);
</script>
```

Simple Hello World Application – React– Approach-3

Solution:

```
Day1/withReact-01.html
-------------------------------------------------
// HTML Code will be as it is
<script type="text/javascript" src="react.js"></script>
<script type="text/javascript" src="react-dom.js"></script>
<script type="text/javascript">
    ReactDOM.render(
    React.DOM.h2(null, "Hello World (First App)- From ReactJS!"),
    document.getElementById("root")
);
</script>
```

Use of Babel

- Babel is a Java Script Compiler.
- It will compile your ES6 or ES2015 code into equivalent ES5 code
- In React You will be writing the ES6 Code, that will be automatically compiled in ES5

How to use it

```
-------------------------------------------------
// HTML Code
<script src="babel.min.js"></script>
<script type="text/babel">
   // ES6 code
</script>
```

Simple Hello World Application – React– Approach-4 - babel

Solution:

```
Day1/withReact-02.html
-------------------------------------------------
// HTML Code will be as it is
<script type="text/javascript" src="react.js"></script>
<script type="text/javascript" src="react-dom.js"></script>
<script src="babel.min.js"></script>
<script type="text/babel">
    ReactDOM.render(
       <h2>Hello World (Second App)- From ReactJS! - Use of babel</h2>,
       document.getElementById('root')
    );
</script>
```

Note: add the babel.js library in your folder

Ref : https://github.com/babel/babel-standalone

- Go to Installation section and copy & paste the babel.min.js link in browser and save it to the desired location by babel.min.js name

Simple Hello World Application – React– Approach-5 - Component

React.Component

```
Day1/withReact-03.html
-------------------------------------------------
// HTML Code will be as it is
<script type="text/javascript" src="react-dom.js"></script>
<script src="babel.min.js"></script>
<script type="text/babel">
/* Create HelloWorld Message Component*/
class HelloWorld extends React.Component{
render(){
return <h2>Hello World (Third App)- From ReactJS! - Use of Component</h2>;
}
}
/* Render the HelloWorld Component- use it as tag */
    ReactDOM.render(<HelloWorld/>,document.getElementById('root'));
</script>
```

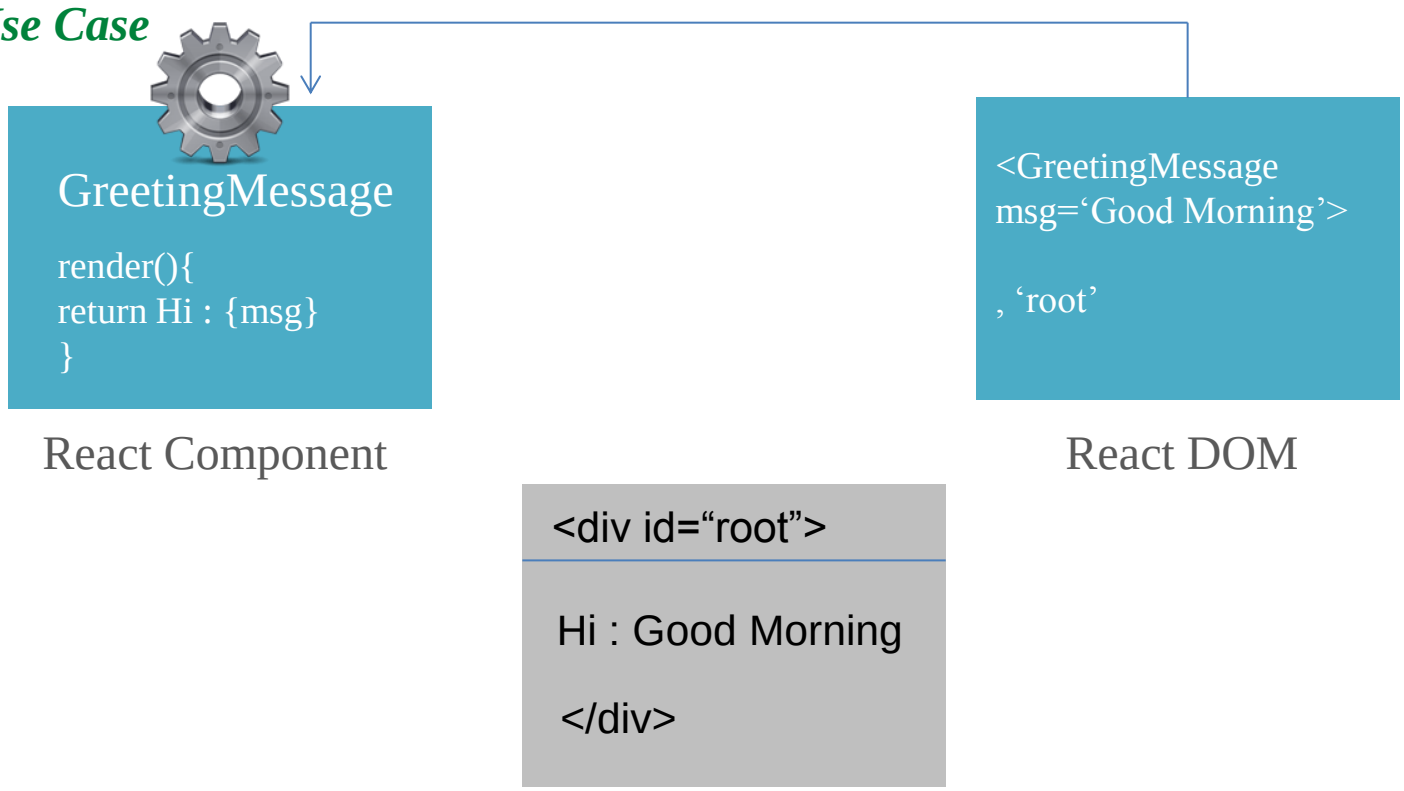
Component helps to split the UI into independent and reusable pieces, and think about each piece in isolation. ***React.Component*** is provided by the ***React***.

Simple Hello World Application – React– Approach-5 - Component

React.Component

React Components implements the render() method that takes the input data and returns what to render (display).

Use Case



Component - props


```
Day1/withReact-03.html
-----------------------------------------------------
//HTML Template will be same

<script type="text/babel">
/* Create GreetingMessage Component
that will read the property value and return that to dom for rendering
*/
class GreetingMessage extends React.Component{
render(){
return <h2>Hi : {this.props.msg}</h2>;
}
}
/* Render the HelloWorld Component- use it as tag */
    ReactDOM.render(<GreetingMessage msg='Good Evening'/>,document.getElementById('root'));
  </script>
```

Injecting one Component into Other

Day1/componentInjection.html

```
<script type="text/babel">
/* Create HelloWorld Message Component*/
class Hello extends React.Component{
render(){
return <h2>Hello - from Hello Component</h2>;
}
}
class World extends React.Component{
render(){
return <h2>World - from World Component</h2>;
}
}
class HelloWorld extends React.Component{
render(){
return <div><Hello/> <World/> </div>;
}
}
/* Render the HelloWorld Component- use it as tag */
    ReactDOM.render(<HelloWorld/>,document.getElementById('root'));
  </script>
```

Bootstrap ReactJS Hello World app

Create React App is a new officially supported way to create single-page React applications. It offers a modern build setup with no configuration.

Installation

First install the global package from the cmd or node command prompt

```
npm install -g create-react-app
```

Creating an App

Now you can use it to create new app

```
create-react-app hello-world
```

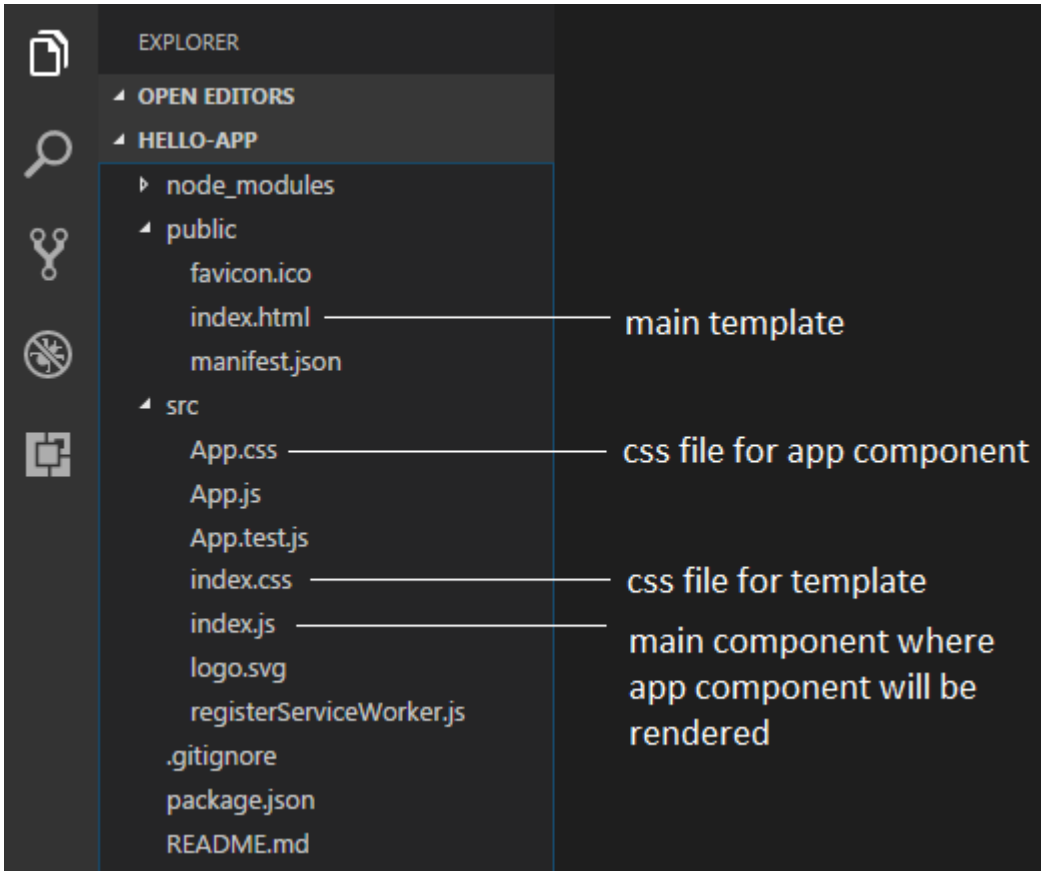
Starting the server

Run below command to launch the development server. The browser will be open automatically with the created app url.

```
npm start
```

Bootstrap ReactJS Hello World app

Some of the important files in the Quick-Start helloapp



Bootstrap ReactJS Hello World app

Let us create MessageComponent that will be rending Welcome message, and this will be used by the App Component

Create new file and name it as Message.component.js in src folder and write below code.

```
Src/MessageComponent.component.js
-------------------------------------------------
import React from 'react';

class MessageComponent extends React.Component{
render(){
return(
<h3>Welcome in React JS - Quick-start Hello App</h3>
);
}
}

export default MessageComponent;
```

Bootstrap ReactJS Hello World app

Import the MessageComponent in App Component and call the MessageComponent as a tag.

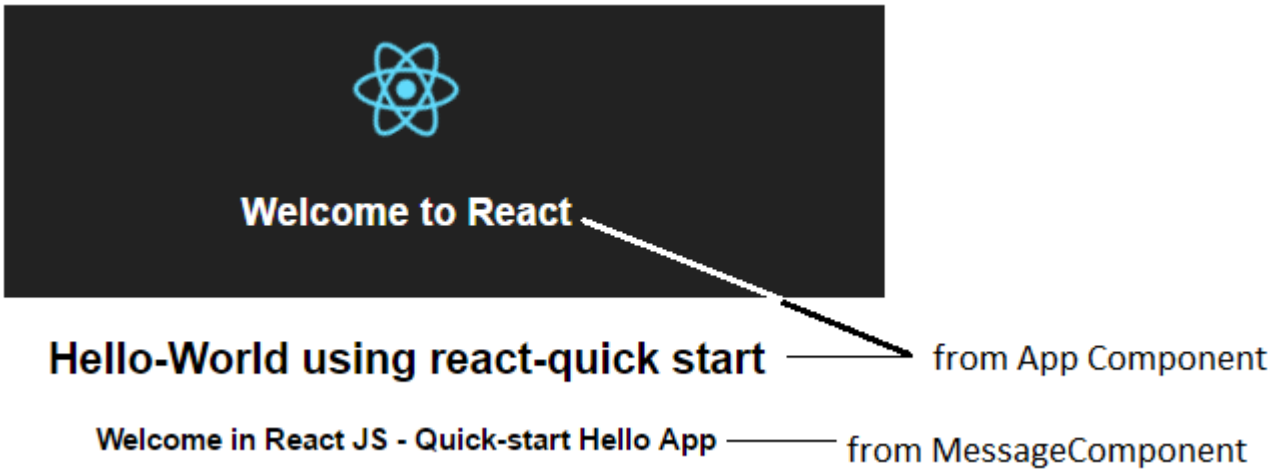
```
Src/MessageComponent.component.js
-------------------------------------------------
-----------------
import MessageComponent from './MessageComponent.component';
class App extends Component {
render() {
return (
-------------------
<p>
<MessageComponent/>
</p>
-------------------
}

export default App;
```

Now save the changes in both files and go back to browser and see the changes.

Bootstrap ReactJS Hello World app

Output



Thank You!

Email: info@yash.com
Web: www.yash.com

Presented by : Pankaj Sharma