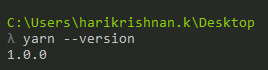
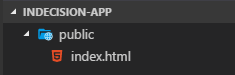
Node and yarn are pre-requisites. Check versions of npm and node.

npm install -g yarn

<https://yarnpkg.com/en/docs/install#windows-tab>

For windows, we might need a restart of pc to take effect.





<!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>Indecision app</title>

</head>

<body>

DEMO TEXT

</body>

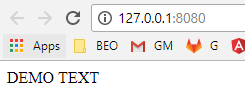
</html>

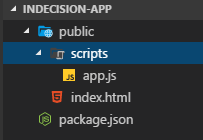


Or









<body>

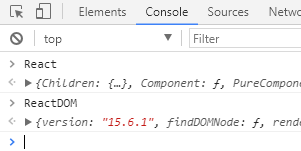
<script src="https://unpkg.com/react@15/dist/react.js"> </script>

<script src="https://unpkg.com/react-dom@15/dist/react-dom.js"> </script>

<script src="scripts/app.js"> </script>

</body>

In dev tools.



In app.js,

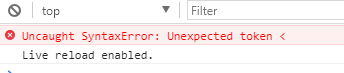
console.log('App.js loaded')

// JSX is a javascript syntax extension

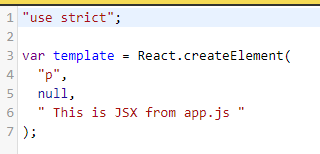
// JSX- Javascript XML

var template = <p> This is JSX from app.js </p>

ReactDOM.render(template, document.getElementById('root'))



Go to babeljs.io



If we paste it,

//var template = <p> This is JSX from app.js </p>

var template = React.createElement(

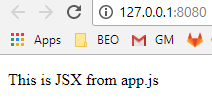
"p",

null,

" This is JSX from app.js "

);

ReactDOM.render(template, document.getElementById('root'))



**Setting up babel locally.**

Babel is a compiler but it doesn’t compile anything by default. We need to load in plugins/ presets for that. A preset is just a group of plugins.

<https://babeljs.io/docs/plugins/>

We usually don’t install plugins individually, but we install presets which wraps all plugins.

env preset gives access to es6 and es7 features.

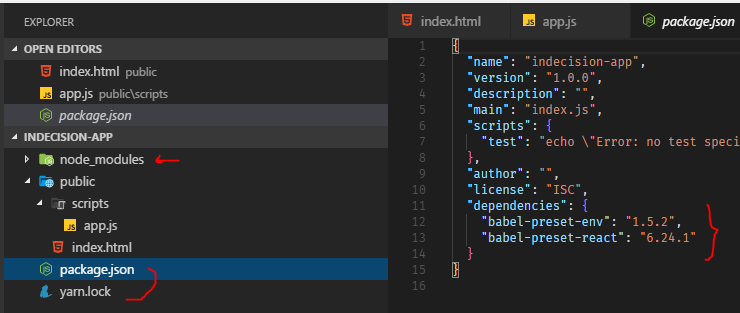




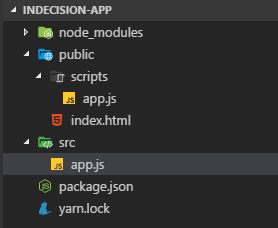
Now we need to install the presets. ( react and env )



yarn add babel-preset-react@6.24.1 [babel-preset-env@1.5.2](mailto:babel-preset-env@1.5.2)



Add a new folder and file.



console.log('App.js loaded')

var template = <p> This is JSX from app.js </p>

ReactDOM.render(template, document.getElementById('root'))

Empty the other app.js file.

babel src/app.js --out-file=public/scripts/app.js --presets=env,react

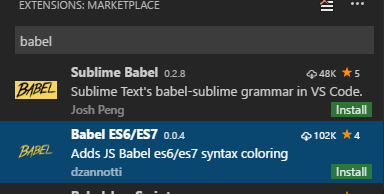
Now it autogenerates the stuff in the public/scripts/app.js folder.

babel src/app.js --out-file=public/scripts/app.js --presets=env,react –watch

Babel saws the changes and compiles them on the fly.

Instead of npm install, we can have yarn install.

Exploring JSX:



We can only have one parent element. So we cant have multiple parallel elements.

var template = <div> <p> This is JSX from app.js </p> <h1> Hello</h1> </div>

And for clarity,

var template = (

<div>

<p> This is JSXss from app.js </p>

<h1> Hello</h1>

</div>

)

In app.js

var userName = 'hari';

var template = (

<div>

<p> { userName } </p>

<h1> Hello</h1>

</div>

)

var user = {

name : 'hari',

age: 32

}

var template = (

<div>

<p> { user.name } </p>

<h1> { user.age } </h1>

</div>

)

Conditional rendering:

var user = {

name : 'hari',

age: 32

}

var template = (

<div>

<p> { user.name } </p>

<h1> { user.age } </h1>

<h2> { getLocation(user.location) } </h2>

</div>

)

function getLocation(location) {

if(location){

return location;

}else{

return "Unknown"

}

}

And we can conditionally return jsx

var user = {

name : 'hari',

age: 32,

location: 'Kochin'

}

var template = (

<div>

<p> { user.name ? user.name : 'Anonymous' } </p>

<h1> { user.age } </h1>

{

getLocation(user.location)

}

</div>

)

function getLocation(location) {

if(location){

return <p> Location { location } </p>;

}else{

return undefined;

}

}