

Installation Guide

First you have to **install node.js** (to use npm (node package manager to install 3rd party apps)

And after you are done with the install of the latest stable version

: Go on the terminal and type

npm i -g create-react-app@latest

i= install, -g = global, and here is the package that we want to install **create-react-app**
= we want to use it to create new react app

Now the next thing you need is code editor : VS Code . Now we will add two extensions to vs code :

1) simple react snippets (By Burke Holland) { with this we can use cc , imrc and many more) and the second is

2) Prettier (By Esben Peterson): one thing with this extension enable formatting on save. To do so go into setting(Ctrl + ,) now search on the top (search box) **formatOnSave** and tick this option .(Or add "editor.formatOnSave" : true)

Now we will use the create-react-app package to create new react application. To do so go to the terminal and type (red text)

create-react-app react-app

(this will download around 150MB so it will take time)

Now this is going to install react, as well as all the third party libraries we need: So it is going to install :

1) Lightweight **development server**

2) Webpack (for bundling our files)

3) **BABEL** (for compiling our js code)

And a bunch of other tools.

Now after this command is complete it's running : go into the react-app folder

cd react-app

And then run npm-start

npm-start

So this will launch our **development server** at port: localhost::3000 and open it inside our default browser.

So if you go in the react-app folder : you will find three folders :

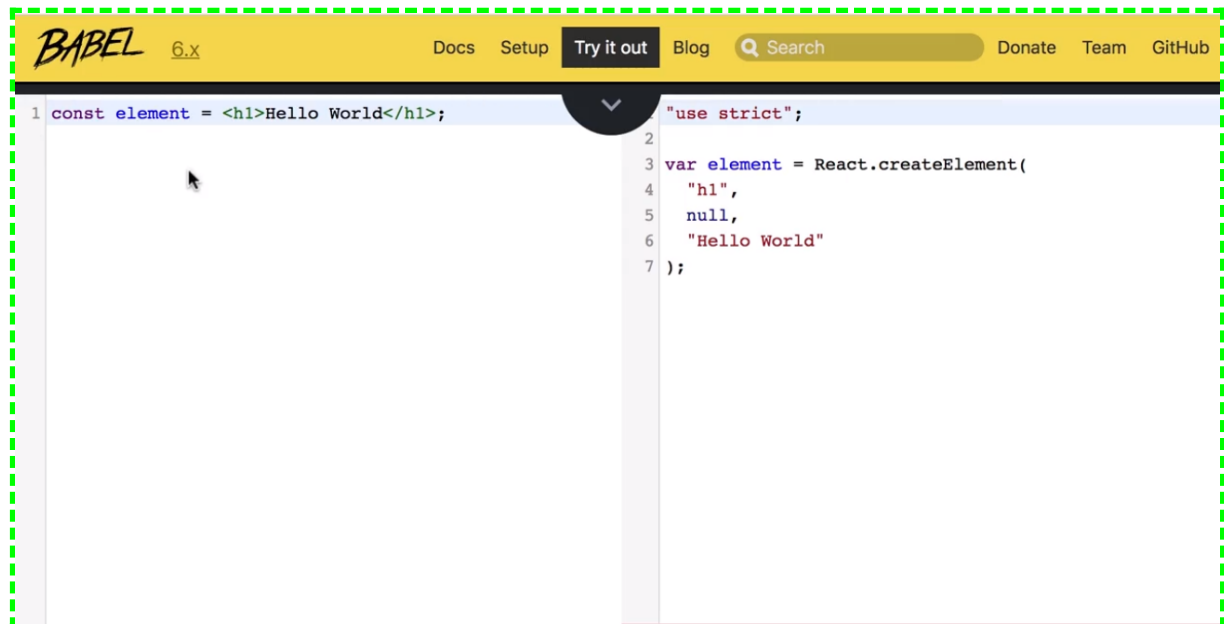
1) **node_modules**: which contains all our third party libraries as well as react it self (don't make any changes)we don't need to worry about this.

2) **public**: it contains all the public assets of our application.

3) **src**: which contain components, where will do all our project work.

Extra: And if you look in the app.js there you will render() method : and inside it you will find something which looks like HTML but it is not, and this is something that you have never

seen in JS itself because we can't have markup like this in Js. What is this ? This is jsx (Javascript XML)and we use this syntax, which look familiar to us to describe what the UI is going to look like. and to work this code we need **BABEL** which is modern JS compiler, so BABEL will take this jsx expression and convert it to javascript code which browser can understand. To have a strong concept: go to website: babeljs.io/repl



See `<h1> Hello World </h1>` is not a string :(you can guess we are not putting string literals) this is **jsx** . See on the right this code is converted to a call to **React.createElement**. The first argument is `h1` : determine the type of our element.

So you can see using jsx to describe what user interface looks like is much more simpler than using plain React code like (in the right).So don't worry we will always use jsx.After BABEL compile the jsx expression it's output is React Element: which have some properties associated : you can see them in console.

So let's look at a demo how to create our first program in React: (you can delete all the files in the src) and start with creating a new file : **index.js**

Go to index.js

- 1) First we need to import a couple of objects from React module

import React from 'react';

So react is the module and React is the object that we are importing from that module.Even though we are not going to use this object directly in our code, but because when our code is compiler (by BABEL) then our code will have `React.createElement` . and that's why we have to import it on the top

- 2) **import ReactDOM from 'react-dom';**

So here is one thing that you need to know about the project that are created with the help of create-react-app, whenever you save the changes this application is automatically restarted. And you need not worry about going to the browser back-and-forth and then refresh the page. **This is called hot module reloading.**

Virtual DOM : which is lightweight in memory representation of UI. It's not the real browser DOM. it's virtual DOM . And the Object: `reactElement` that you see is a part of virtual DOM. So whenever the state of this object changes react will get a new react element and then react will compare it with the previous element (old one), figure out what are the changes . and then it will reach out the Real DOM and update it accordingly. So if we want to render the output of jsx expression which is element of virtual DOM on the Real DOM :
So **ReactDOM.render(element which we want to render, (where we want to render it) :**

To Setup Vedly Project

In Vedly we have to install two more libraries (Bootstrap, Font-Awesome(to render beautiful modern font) Go to terminal:

`create-react-app vedly`

`cd vedly`

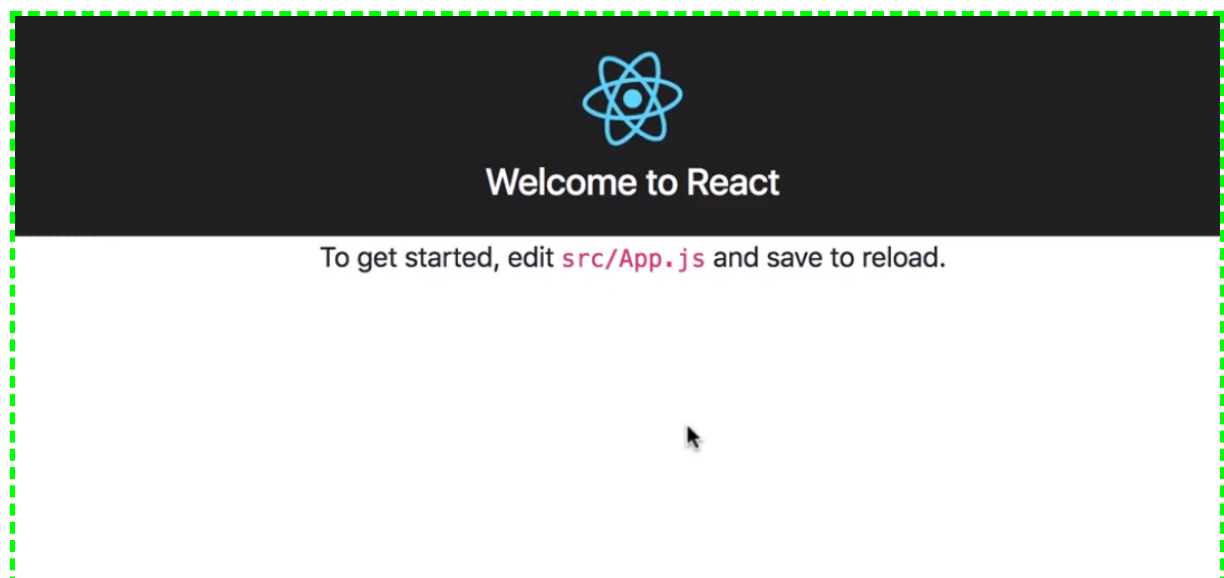
`npm i bootstrap@4.1.1 font-awesome@4.7.0`

Now after both libraries are installed, we have to import their css file. For that go to the src folder , open index.js and write these two lines.

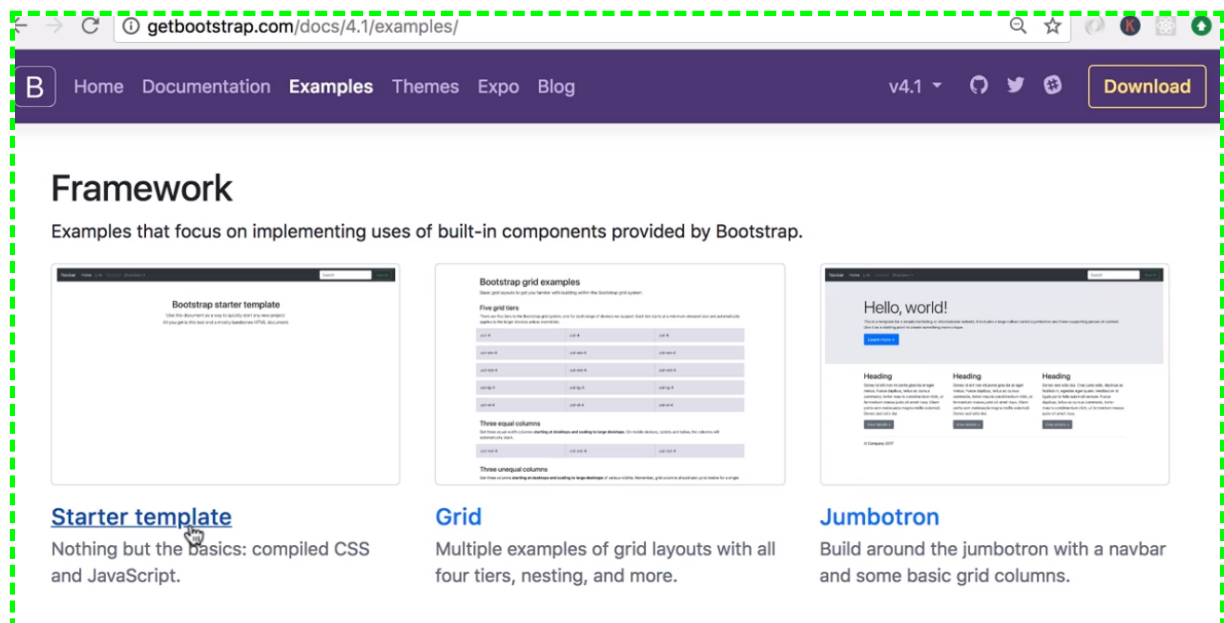
```
import 'bootstrap/dist/css/bootstrap.css';  
import 'font-awesome/css/font-awesome.css';
```

So both these libraries are installed under `node_modules` folder, when we will build this application the webpack(which bundles our js and css files) will pull these css files and put in final bundle.

Now to get rid from this black panel (Welcome to React)



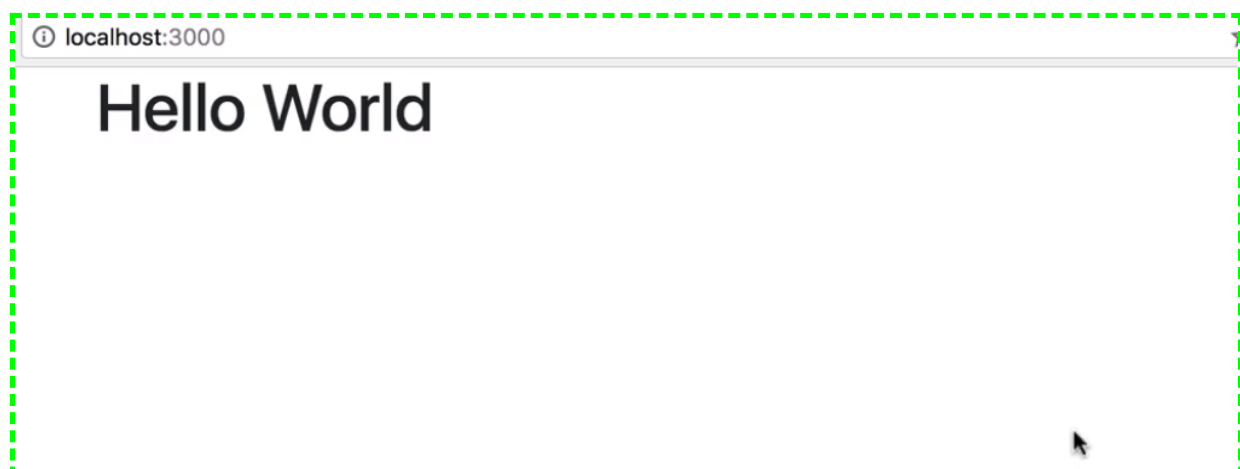
We will add a basic bootstrap template : To do so go to **getbootstrap.com** => under **example** => scroll and there => select starter template.



Now open this page : look the viewPage Source.if you see then you will find that page is using `<main className= "container"> </main>` .So we will use that in our root component which we are rendering (which is App.js) till now.

```
class App extends Component {
  render() {
    return (
      <main className="container">
        <h1>Hello World</h1>
      </main>
    );
  }
}
```

And this will be the output:



We will use some sample data while building this application (vidly):

Link: <https://drive.google.com/open?id=1LMklT5EuTBysQW9bCeRTz1V9Fd2SI4p3>

There is a services folder inside the provided link, what you have to do copy it inside the source (src) folder.

And now you can start with component folder inside src : add there are new file movies.jsx and many more will be added in future

To uninstall:

~ npm uninstall -g create-react-app