

Tuto simple React

1) Installer create react app

Il s'agit d'un outil qui permet de générer le squelette de base de react en installant tous les packages qui nous avons besoin.

Pour l'installer il faut aller dans le terminal Node prompt et faire les commandes suivantes

```
Node.js command prompt
Your environment has been set up for using Node.js 12.14.0 (x64) and npm.

C:\Users\sautr>node -v
v12.14.0

C:\Users\sautr>npm -v
6.13.4

C:\Users\sautr>npm install --global npm@latest
C:\Users\sautr\AppData\Roaming\npm\npm -> C:\Users\sautr\AppData\Roaming\npm\node_modules\npm\bin\npm-cli.js
C:\Users\sautr\AppData\Roaming\npm\npx -> C:\Users\sautr\AppData\Roaming\npm\node_modules\npm\bin\npx-cli.js
+ npm@6.13.7
added 434 packages from 860 contributors in 19.49s

C:\Users\sautr>npm install --global create-react-app
C:\Users\sautr\AppData\Roaming\npm\create-react-app -> C:\Users\sautr\AppData\Roaming\npm\node_modules\create-react-app\index.js
+ create-react-app@3.4.0
added 99 packages from 47 contributors in 68.248s

C:\Users\sautr>
```

Nommé votre repo

```
C:\Users\sautr>create-react-app todo-list

Creating a new React app in C:\Users\sautr\todo-list.

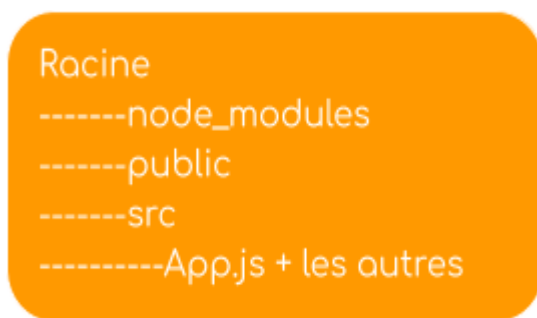
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

[.....] | fetchMetadata: sill resolveWithNewModule scheduler@0.18.0 c
```

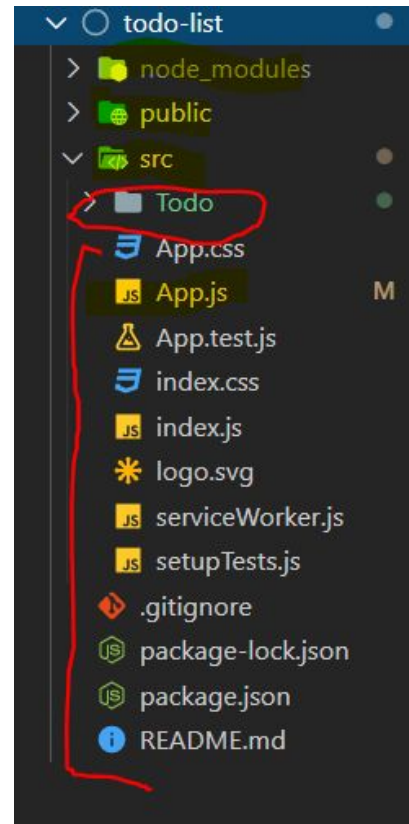
Dans le dossier créé vous pouvez faire une **npm start** pour lancer le projet

2) Ecrire du react

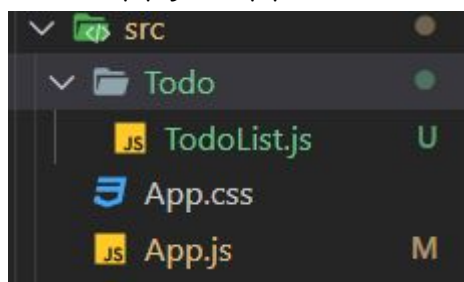
Voici le squelette de base



Nous pouvons écrire notre code dans la racine du dossier src comme indiqué
Ici le dossier créé est nommé "Todo"



Dans App.js supprimer le contenu: exemple de structure simple



```
import React, { Component } from 'react';
//import logo from './logo.svg';
import './App.css';
//On importe le fichier /src/todo/todolist.js
import TodoList from './Todo/TodoList';

class App extends Component {

  render() {
    return(
//retour une réponse dans le vue ici la page va afficher Ma todo
```

```

    <TodoList />

    );

  }

}

```

Dans Todoliste.js

```

export default App;
//1 importe le module react
import React,{ Component } from 'react';

//2 on donne un class ici c todolist
class TodoList extends Component{
//on donne un comportement ici c render ->rendre un donné

  render(){

//ecrire un donnée qui sera affiché via App.js
    return (
      <div>
        <h1>Ma todo</h1>
      </div>
    );
  };
}
//on importe la class nommé précédemment Todolist
export default TodoList;

```

3) Méthode construtor

Elément construtor doit être obligatoirement accompagné d'un élément super

Stat= État c'est un élément qui peut être changer contrairement au

Props= accessoir qui eux sont des élément statique qui ne change pas

```
constructor() {  
  super();  
  this.state={  
    userInput: '',  
    items: []  
  }  
}
```

```
addTodo(event) {  
  
}
```

```
{this.addTodo.bind(this)}
```

Bind(this) fait référent à la event addtodo

```
▼ {userInput: "oceane", items: Array(1)} ⓘ  
  userInput: "oceane"  
  ▶ items: ["oceane"]  
  ▶ __proto__: Object  
> cc
```

```
className="form-control mb-2"
```

ClassName permet de donnée une class css dans react afin utiliser du bootstrap comme ceci

Ma todo

intégrer le lien bootstrap dans index.html le fichier se trouve dans public de react

```
<head>  
  <meta charset="utf-8" />  
  <link rel="icon" href="%PUBLIC_URL%/favicon.ico" />
```

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.
min.css"
integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9Mu
hOf23Q9Ifjh" crossorigin="anonymous">
```

TP todo liste:

App.js

```
import React, { Component } from 'react';
import './App.css';
import TodoList from './Todo/TodoList';

class App extends Component {

  render() {
    return(
      <TodoList />
    );
  }
}

export default App;
```

Todoliste.js

```
import React,{ Component } from 'react';

class TodoList extends Component{
  constructor() {
    super();
    this.state={
      userInput:'',
      items:[]
    };
  }
  onChange(event) {
```

```

        //evenement du clavier
        // console.log(event.target.value);
        this.setState({
            userInput:event.target.value
        },
        //verif si c egal userinput=valeur qui ecrit dans le input
        ()=> console.log(this.state.userInput));
    }

//ajout item dans la todo
addTodo(event) {
    //empêche le relogue de la page car il s'agit d'un formulaire
    event.preventDefault();
    this.setState({
        userInput:'',
        //copie et ajout d'un new element dans le tableau
        items:[...this.state.items,this.state.userInput]
    },
    ()=>console.log(this.state));
}

deleteTodo(item) {
    //creation variable
    const array = this.state.items;
    const index = array.indexOf(item);
    //fonction qui supprime le tableau de la todo
    array.splice(index, 1);
    //mon item = est egal mon array apres la supprimer la todo
    this.setState({
        items: array
    });
}

//voir la todo

renderTodos () {

```

```

        return this.state.items.map((item)=> {
            return(
                <div className="list-group-item"key={item}>
                    {item} <button className="btn btn-danger"
                        onClick={this.deleteTodo.bind(this,
item)}>Supprimer</button>
                </div>
            );
        });
    }

    render() {
        return (
            <div>
                <h1 align="center">Ma todo</h1>
                <form className="form-row align-items-center">
                    <input
                        value={this.state.userInput}
                        type="text"
                        placeholder="Ecrire votre tâche"
                        //action qui fait appel à la fonction onChange
                        onChange={this.onChange.bind(this)}
                        //className= class css
                        className="form-control mb-2"
                    />
                    <button
                        onClick={this.addToDo.bind(this)}
                        className="btn btn-success"
                    >
                        Ajouter
                    </button>
                </form>
                <div className="list-group">
                    {this.renderTodos()}
                </div>
            </div>
        );
    }
};

```

```
}  
export default TodoList;
```