



Sarah
Mike
Lola
Bob

## Détail de Sarah

- Id:1
- Role:ROOT
- Actif:NON

- └ .vscode
  - { } settings.json
- └ node\_modules
- └ public
  - ★ favicon.ico
  - <> index.html
  - { } manifest.json
- └ src
  - └ actions
    - JS index.js
  - └ components
    - JS App.js
  - └ containers
    - JS User\_detail.js
    - JS User\_list.js
  - └ reducers
    - JS index.js
    - JS reducer\_users.js
    - JS reducers\_activeUsers.js
  - # App.css
  - JS App.test.js
  - # index.css
  - JS index.js
  - 🖼 logo.svg
  - JS registerServiceWorker.js
- 📄 .gitignore
- { } package-lock.json
- { } package.json
- 📖 README.md

# index.html

```
<> index.html x
2  <html lang="en">
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
6      <meta name="theme-color" content="#000000">
7      <!--
8      | manifest.json provides metadata used when your web app is added to the
9      | homescreen on Android. See https://developers.google.com/web/fundamentals/engage-and-retain/web-app-manifest/
10     -->
11     <link rel="manifest" href="%PUBLIC_URL%/manifest.json">
12     <link rel="shortcut icon" href="%PUBLIC_URL%/favicon.ico">
13     <link rel="stylesheet" href="/style/style.css">
14     <link rel="stylesheet" href="https://cdn.rawgit.com/twbs/bootstrap/48938155eb24b4ccdde09426066869504c6dab3c/dist/css/bootstrap.min.css">
15     <!--
16     | Notice the use of %PUBLIC_URL% in the tags above.
17     | It will be replaced with the URL of the `public` folder during the build.
18     | Only files inside the `public` folder can be referenced from the HTML.
19
20     | Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC_URL%/favicon.ico" will
21     | work correctly both with client-side routing and a non-root public URL.
22     | Learn how to configure a non-root public URL by running `npm run build`.
23     -->
24     <title>React App</title>
25   </head>
26   <body>
27     <noscript>
28       | You need to enable JavaScript to run this app.
29     </noscript>
30     <div class="container"></div>
31     <!--
```

- └ .vscode
  - { settings.json
- node\_modules
- └ public
  - ★ favicon.ico
  - <> index.html
  - { manifest.json
- └ src
  - └ actions
    - JS index.js
  - └ components
    - JS App.js
  - └ containers
    - JS User\_detail.js
    - JS User\_list.js
  - └ reducers
    - JS index.js
    - JS reducer\_users.js
    - JS reducers\_activeUsers.js
  - # App.css
  - JS App.test.js
  - # index.css
  - JS index.js**
  - 🖼 logo.svg
  - JS registerServiceWorker.js
- 📄 .gitignore
- { package-lock.json
- { package.json
- 📖 README.md

## index.js

```

JS index.js
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import App from './components/App';
4  import registerServiceWorker from './registerServiceWorker';
5
6  import reducers from './reducers';
7
8  //----Import Redux
9  import { Provider } from 'react-redux';
10 import { createStore, applyMiddleware } from 'redux';
11
12 const createStoreWithMiddleWare = applyMiddleware()(createStore);
13
14
15
16 ReactDOM.render(
17   <Provider store={createStoreWithMiddleWare(reducers, window.__REDUX_DEVTOOLS_EXTENSION__ && window.__REDUX_DEVTOOLS_EXTENSION__())}>
18     <App />
19   </Provider>,
20   document.querySelector('.container'));
21   registerServiceWorker();
22

```

- └ .vscode
  - { } settings.json
- └ node\_modules
- └ public
  - ★ favicon.ico
  - <> index.html
  - { } manifest.json
- └ src
  - └ actions
    - JS index.js
  - └ components
    - JS App.js
  - └ containers
    - JS User\_detail.js
    - JS User\_list.js
  - └ reducers
    - JS index.js
    - JS reducer\_users.js
    - JS reducers\_activeUsers.js
- # App.css
- JS App.test.js
- # index.css
- JS index.js
- 🖼 logo.svg
- JS registerServiceWorker.js
- 💎 .gitignore
- { } package-lock.json
- { } package.json
- 📖 README.md

## component/App.js

```
JS App.js x JS User_detail.js JS User_list.js JS index.js .../actions JS index.js .../reducers JS reducer_users.js JS reducers_activeUsers.js
1 //----Import composant react
2 import React, { Component } from 'react';
3
4 //---- Import fichier style
5 import '../App.css';
6
7 //----Import de composant
8 import User_list from '../containers/User_list';
9 import User_detail from '../containers/User_detail';
10
11 class App extends Component {
12   render() {
13     return (
14       <div>
15         <User_list/>
16         <User_detail/>
17       </div>
18     );
19   }
20 }
21
22 export default App;
23
```

- .vscode
  - {} settings.json
- node\_modules
- public
  - ★ favicon.ico
  - <> index.html
  - { } manifest.json
- src
  - actions
    - JS index.js
  - components
    - JS App.js
  - containers
    - JS User\_detail.js
    - JS User\_list.js
  - reducers
    - JS index.js
    - JS reducer\_users.js
    - JS reducers\_activeUsers.js
- # App.css
- JS App.test.js
- # index.css
- JS index.js
- 📁 logo.svg
- JS registerServiceWorker.js
- 🔍 .gitignore
- { } package-lock.json
- { } package.json
- 📖 README.md

JS App.js   JS User\_list.js   JS User\_detail.js   JS index.js .../actions   JS index.js .../reducers   JS reducer\_users.js   JS reducers\_activeUsers.js

```

1  import React, { Component } from 'react';
2  import { connect } from 'react-redux';
3  import { bindActionCreators } from 'redux';
4
5  //----import de composant
6  import {selectUser} from '../actions/index';
7
8  class User_list extends Component {
9
10     render () {
11         selectUser();
12         return (
13             <div>
14                 <ul className="col-md-4">
15                     {
16                         this.props.myUsers.map((user)=>{
17                             return(
18                                 <li className="list-group-item" key={user.id} onClick={()=>{this.props.selectUserAction(user)}}>
19                                     {user.name}
20                                 </li>
21                             )
22                         })
23                     }
24                 </ul>
25             </div>
26         )
27     }
28 }
29
30 //NOTE: La fonction mapStateToProps prend (state) en argument qui est le retour
31 // du state concerner, tout se qui sort de cet fonction est mis dans les props.
32 //Donc myUsers est disponible et utilisable en tant que props.
33 function mapStateToProps(state){
34     return {
35         myUsers : state.users
36     }
37 }
38
39 function mapDispatchToProps(dispatch){
40     return bindActionCreators({selectUserAction : selectUser},dispatch)
41 }
42
43 export default connect(mapStateToProps, mapDispatchToProps)(User_list);

```

En plus de {connect} notre composant, Bien pensé a bind notre composant a actionCreators pour map les ACTIONS si on a besoin des actions.

containers/User\_list.js

## containers/User\_detail.js

Dans ce composant nous avons pas besoin de bind le actionCreators car nous avons seulement besoin des props du store, nous avons pas besoin des actions, donc on utilise uniquement {connect} pour avoir accès aux props du store

```
JS App.js JS User_list.js JS User_detail.js x JS index.js .../actions
1 import React, { Component } from 'react'
2 import {connect} from 'react-redux';
3
4
5 class User_detail extends Component {
6   render () {
7     const {myActiveUser} = this.props;
8     if(!myActiveUser){
9       return <div>Select a user for begin</div>
10    }
11    return (
12      <div>
13        <h3>Détail de {myActiveUser.name}</h3>
14        <ul>
15          <li>Id:{myActiveUser.id}</li>
16          <li>Role:{myActiveUser.role}</li>
17          <li>Actif:{myActiveUser.active}</li>
18        </ul>
19      </div>
20    )
21  }
22 }
23
24 function mapStateToProps(state){
25   return {
26     myActiveUser : state.activeUser
27   }
28 }
29
30 export default connect(mapStateToProps)(User_detail)
```

## action/index.js

- └ .vscode
  - { } settings.json
- node\_modules
- └ public
  - ★ favicon.ico
  - <> index.html
  - { } manifest.json
- └ src
  - └ actions
  - JS index.js**
  - └ components
    - JS App.js
  - └ containers
    - JS User\_detail.js
    - JS User\_list.js
  - └ reducers
    - JS index.js
    - JS reducer\_users.js
    - JS reducers\_activeUsers.js
  - # App.css
  - JS App.test.js
  - # index.css
  - JS index.js
  - 🖼 logo.svg
  - JS registerServiceWorker.js
- 💎 .gitignore
- { } package-lock.json
- { } package.json
- 📖 README.md

```
JS App.js    JS User_list.js    JS User_detail.js    JS index.js .../actions ✕

1  export const USER_SELECTED = 'USER_SELECTED';
2
3  export function selectUser(user){
4      console.log('selected: ',user);
5      return{
6          type: USER_SELECTED,
7          payload: user
8      }
9  }
```



## reducers/index.js

JS App.js

JS User\_list.js

JS User\_detail.js

JS index.js .../actions

JS index.js .../reduc

```
1 import {combineReducers} from 'redux';
2 import reducer_users from './reducer_users';
3 import reducer_activeUsers from './reducers_activeUsers';
4
5
6
7 // NOTE: {CombineReducers} permet de combiner l'état des reducers avec REDUX.
8 // users est donc un état courant dans le store REDUX,
9 //c'est le fichier de config des reducers
10 const rootReducer = combineReducers({
11   users : reducer_users,
12   activeUser : reducer_activeUsers
13 });
14
15 export default rootReducer;
```



## reducers/reducers\_Users.js

- └ .vscode
  - { } settings.json
- node\_modules
- └ public
  - ★ favicon.ico
  - <> index.html
  - { } manifest.json
- └ src
  - └ actions
    - JS index.js
  - └ components
    - JS App.js
  - └ containers
    - JS User\_detail.js
    - JS User\_list.js
  - └ reducers
    - JS index.js
    - JS reducer\_users.js
    - JS reducers\_activeUsers.js
  - # App.css
  - JS App.test.js
  - # index.css
  - JS index.js
  - 🖼 logo.svg
  - JS registerServiceWorker.js
- 💎 .gitignore
- { } package-lock.json
- { } package.json
- 📖 README.md

JS reducer\_users.js x

JS reducers\_activeUsers.js

```
1 export default function(){
2   //NOTE: ceci est un réducuer.
3   return[
4     {name:'Sarah', id:1, role:'ROOT', active:'NON'},
5     {name:'Mike', id:2, role:'USER', active:'OUI'},
6     {name:'Lola', id:3, role:'USER', active:'NON'},
7     {name:'Bob', id:4, role:'ADMIN', active:'OUI'},
8   ]
9 };
```

reducers/reducers\_activeUsers.js

## JS reducers\_activeUsers.js x

```
1  import {USER_SELECTED} from '../actions/index';
2
3  export default function (state = null, action){
4      switch(action.type){
5          case USER_SELECTED :
6              return action.payload;
7          default:
8              return state;
9      }
10
11 }
```