

Nach dem man npm install redux & npm install react-redux ausgeführt hat geht es los.

--Den Store erstellen--

Zuerst erstellst du eine Store Datei welche den Zustand aller Componente Verwaltet z.b.:

MeinStore.js

In dieser Datei importierst du den store von redux:

```
import {createStore} from 'redux';
```

Dann erstellst du einen initialen Zustand welches ein oder mehrere Objekte beinhaltet

und übergibst diesem Objekt/Objekte einen Wert:

```
export const initialState = {  
  farbe: 'green'
```

```
};
```

Im Store importieren wir auch den reducer den wir erstellt haben aus der Datei MeinReducer.js:

```
import { reducer } from './MeinReducer';
```

Jetzt erstellen wir ein Objekt mit dem Namen store und

übergeben diesem die createStore Funktion von Redux, diese enthalten zusätzlich

den reducer und den initialState:

```
export const store = createStore(reducer, initialState);
```

```
import {createStore} from 'redux';  
import { reducer } from './MeinReducer';  
  
export const initialState = {  
  farbe: 'green'  
};  
  
export const store = createStore(reducer, initialState, window.__REDUX_DEVTOOLS_EXTENSION__ && window.__REDUX_DEVTOOLS_EXTENSION__());
```

--Den Reducer erstellen z.b. MeinReducer.js--

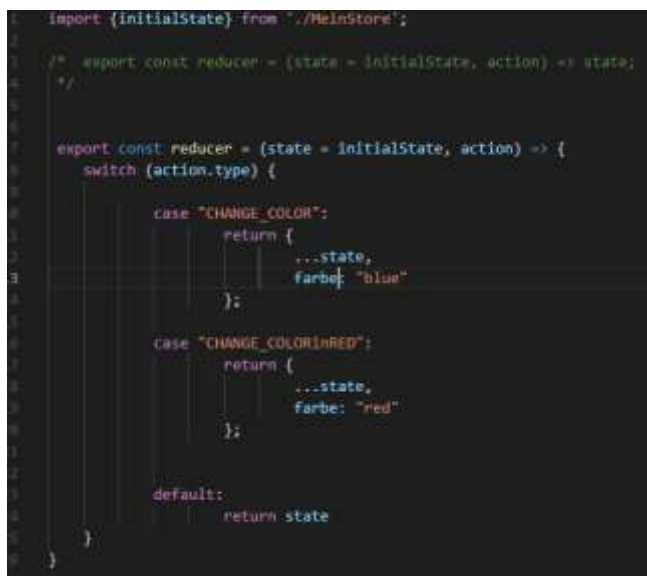
Als erstes importieren wir den initialenState aus der Datei MeinStore.js:

```
import {initialState} from './MeinStore';
```

Dann erstellen wir einen reducer, dieser beinhaltet den Zustand & die action,

dem Zustand wird der initialeState übergeben, mit Switch geben wir die jeweiligen Einstellungen weiter und zum Schluß wird nochmal der state zurückgegeben:

```
export const reducer = (state = initialState, action) => {  
  switch (action.type) {  
    case "CHANGE_COLOR":  
      return {  
        ...state,  
        farbe: "blue"  
      };  
    case "CHANGE_COLORinRED":  
      return {  
        ...state,  
        farbe: "red"  
      };  
    default:  
      return state  
  }  
}
```

A screenshot of a code editor with a dark background and light-colored text. The code is the same as the one shown in the previous block, but with syntax highlighting. The import statement is in green, the function definition is in blue, the switch statement is in purple, and the return statements are in red. The code is wrapped in a pre tag.

```
import {initialState} from './MeinStore';  
  
/* export const reducer = (state = initialState, action) => state;  
*/  
  
export const reducer = (state = initialState, action) => {  
  switch (action.type) {  
    case "CHANGE_COLOR":  
      return {  
        ...state,  
        farbe: "blue"  
      };  
    case "CHANGE_COLORinRED":  
      return {  
        ...state,  
        farbe: "red"  
      };  
    default:  
      return state  
  }  
}
```

--Die Action erstellen z.b. MeineActions.js--

wir erstellen ein actionCreator diesen übergeben wir später der connect Funktion,
den type benötigen wir um ihn mit dem reducer zu verbinden:

```
export const actionCreators = {  
  nameOffTheAction: (event) => {  
    return {type: "CHANGE_COLOR"};  
  },  
  nameOffTheActionZwei: (event) => {  
    return {type: "CHANGE_COLORinRED"};  
  }  
}
```



```
1 |  
2 | export const actionCreators = {  
3 |   nameOffTheAction: (event) => {  
4 |     return {type: "CHANGE_COLOR"};  
5 |   },  
6 |  
7 |   nameOffTheActionZwei: (event) => {  
8 |     return {type: "CHANGE_COLORinRED"};  
9 |   }  
10 |  
11 |  
12 | }  
13 |
```

--Den Provider in die index.js u. da in die ReactDOM einfügen--

Zuerst muss man den Provider von React Redux importieren:

```
import {Provider} from 'react-redux';
```

Dann in die DOM einfügen und den store festlegen:

```
ReactDOM.render(
```

```
<Provider store={store}>
```

```
<AppContainer />
```

```
</Provider>
```

```
, document.getElementById('root'));
```

```
registerServiceWorker();
```

```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4
5  import {AppContainer} from './App';
6  import {store} from './MeinStore';
7  import {Provider} from 'react-redux';
8
9
10 import registerServiceWorker from './registerServiceWorker';
11
12 ReactDOM.render(
13   <Provider store={store}>
14     <AppContainer />
15   </Provider>
16   , document.getElementById('root'));
17   registerServiceWorker();
18
```

--Daten aus dem Redux Store extrahieren und in die Container Komponente über die Props einfügen u. actionCreators importieren--

Zuerst muss ich connect von React Redux importieren, danach die actionCreators:

```
import {connect} from 'react-redux';
```

```
import {actionCreators} from './MeineActions';
```

Dann erzeugt man eine neue Komponente und übergibt dieser dann die Funktion connect mit mapStateToProps u. actionCreators & App:

```
export const AppContainer = connect(mapStateToProps, actionCreators)(App);
```

WICHTIG: Diese müssen wir in der index.js Datei einfügen:

```
<Provider store={store}>
```

```
<AppContainer />
```

```
</Provider>
```

Dann erstellen wir eine Funktion mapStateToProps, diese holt sich den gesamten Zustand aus dem Redux Store u. gibt einen Teil zurück:

```
function mapStateToProps (state) {  
  return {farbe: state.farbe};  
}
```

--Einfügen der Props in die Anwendung mit dem initialenState--

```
<h2 style={{color:this.props.farbe}}>
```

```
  Das ist der aktuelle Zustand</h2>
```

Und dann importieren wir die props in die Events:

```
<button style={{margin: 20}}>
```

```
  onClick={this.props.nameOffTheAction}
```

```
    >Ändere die Farbe mit Redux
```

```
</button>
```

```
<button style={{margin: 20}}>
```

```
  onClick={this.props.nameOffTheActionZwei}
```

```
    >Ändere die Farbe mit Redux
```

```
</button>
```

```

1  import React, {Component} from 'react';
2  import logo from './logo.svg';
3  import './App.css';
4
5
6  import {connect} from 'react-redux';
7  import {actionCreators} from './MeineActions';
8
9
10
11 class App extends Component {
12
13
14
15     render() {
16         return (
17             <div className="App">
18                 <header className="App-header">
19                     <img src={logo} className="App-logo" alt="logo" />
20                     <h1 className="App-title">Welcome to React YO</h1>
21                     <h2 style={{color:this.props.farbe}}>
22                         Das ist der aktuelle Zustand</h2>
23                     <button style={{margin: 20}}
24                         onClick={this.props.nameOffTheAction}
25                         >Ändere die Farbe mit Redux</button>
26                     <button style={{margin: 20}}
27                         onClick={this.props.nameOffTheActionZwei}
28                         >Ändere die Farbe mit Redux</button>
29                 </header>
30                 <p className="App-intro">
31                     To get started, edit <code>src/App.js</code> and save to reload.
32                 </p>
33             </div>
34         );
35     }
36 }
37
38
39
40
41 function mapStateToProps (state) {
42
43     return {farbe: state.farbe}
44 }
45 export const AppContainer = connect(mapStateToProps, actionCreators)(App);
46
47 export default App;
48
49

```