Component

Provider

Connect

The Action Creator *selectSong* is binded to the dispatch function. *onClick()* invoke the dispatch function with the returned Action object as argument

The Component SongList uses the state: songs

class **SongList** extends React.Component {

    renderList(){

        return this.props.**songs**.map((**song**) => {

            return (

                <div className="item" key={**song**.title}>

                         <button className="ui button primary" onClick={() => this.props.**selectSong(song)}>**

                            Select

                        </button>

                     <div className="content">{**song**.title}</div>

                </div>

            );

        });

    }

    render(){

         return <div className="ui divded list">{this.renderList()}</div>

    }

}

const **mapStateToProps** = (state) => {

    return { songs: state.songs };

}

const **mapDispatchToProps** = {selectSong};

export default **connect**(**mapStateToProps**, **mapDispatchToProps**)( **SongList**);

The Component SongDetail uses the state: selectedSong

C/Henrik Høltzer Zealand 25.02.2020

React-Redux (Songs App - Overview )

export **const selectSong** = (song) => {

    // Return an action

    return {

        type: 'SONG\_SELECTED',

        payload: song

    };

};

ReactDOM.render(

    <**Provider** store={**createStore(reducers)**}>

        **<App** />

    **</Provider>,**

     document.getElementById('root')

);

const **App** = () => {

    return (

        <div className="ui container grid">

            <div className="ui row">
                <div className="column eight wide">

                    <**SongList** />

                </div>

                <div className="column eight wide">

                    <**SongDetail** />

                </div>

            </div>

        </div>

    );

};

const **SongDetail** = ({ song }) => {

    if (!song){

        return <div>Select a song</div>;

    }

    return (

        <div>

            <h3>Details for: </h3>

            <p>

               Title: {song.title} <br/>

               Duration: {song.duration}

            </p>

        </div>

    );

};

const **mapStateToProps** = (state) => {

    return { song : state.selectedSong };

}

export default **connect**(**mapStateToProps**)(**SongDetail**);

Action Creator

Action

selectedsong

Store

Reducers

songsReducer

selectedSongsReducer

songs

State

const **songsReducer** = () => {

    return [

        {title: 'No Scrubs', duration: '4:05'},

        //….

    ];

};

const **selectedSongReducer** = (selectedSong=null, action) => {

    if (action.type === 'SONG\_SELECTED'){

        return action.payload;

    }

    return selectedSong;

}

export default **combineReducers**({

    **songs**: songsReducer,

    **selectedSong**: selectedSongReducer

});