```
class StoreWorkerAdapter {
  constructor(worker) {
    this.worker = worker;
    this.lastKnownState = defaultState;
    this.listeners = [];

    this.worker.addEventListener("message", (message) => {
      this.lastKnownState = message.data;
      for (let listener of this.listeners) {
        listener();
      }
    })
  }

  dispatch = (action) => {
    this.worker.postMessage(action);
  }

  subscribe = (listener) => {
    this.listeners.push(listener)
  }

  getState = () => {
    return this.lastKnownState;
  }
}
```

Użycie adaptera

```
import React from 'react';
import ReactDOM from 'react-dom';
import AppConnector from './components/AppConnector';
import {Provider} from "react-redux";
import {createStore} from 'redux'
import {reducer} from "./reducer";

const store = createStore(reducer);

ReactDOM.render(
    <Provider store={store}>
        <AppConnector/>
    </Provider>,
    document.getElementById('root')
);
```