



This repository

Search

Pull requests

Issues

Marketplace

Gist



 **gildata** / **redux-seeds**

 Unwatch ▾

2

★ Star

0

 Fork

0

↔ Code

! Issues 1

 Pull requests 0

 Projects 0

 Wiki

⚙ Settings

Insights ▾

Branch: master ▾

**redux-seeds** / **react-slingshot-docs** / **readme.md**

Find file

Copy path



**xgqfrms-GitHub** Update readme.md

81f410d just now

1 contributor

344 lines (253 sloc) | 8.48 KB

Raw

Blame

History



# react-slingshot-docs

docs/FAQ

<https://github.com/coryhouse/react-slingshot/blob/master/docs/FAQ.md>

<https://github.com/coryhouse/react-slingshot/blob/master/docs/FAQ.md#i-just-want-an-empty-starter-kit>

## actions

<https://github.com/coryhouse/react-slingshot/blob/master/src/actions/fuelSavingsActions.js>

```

import * as types from '../constants/actionTypes';
import {getFormattedDateTime} from '../utils/dateHelper';

// example of a thunk using the redux-thunk middleware
export function saveFuelSavings(settings) {
  return function (dispatch) {
    // thunks allow for pre-processing actions, calling apis, and dispatching multiple actions
    // in this case at this point we could call a service that would persist the fuel savings
    return dispatch({
      type: types.SAVE_FUEL_SAVINGS,
      dateModified: getFormattedDateTime(),
      settings
    });
  };
}

export function calculateFuelSavings(settings, fieldName, value) {
  return {
    type: types.CALCULATE_FUEL_SAVINGS,
    dateModified: getFormattedDateTime(),
    settings,
    fieldName,
    value
  };
}

```

## reducers

<https://github.com/coryhouse/react-slingshot/blob/master/src/reducers/fuelSavingsReducer.js>

```

import {SAVE_FUEL_SAVINGS, CALCULATE_FUEL_SAVINGS} from '../constants/actionTypes';

```

```

import calculator from '../utils/fuelSavingsCalculator';
// object-assign & Object.assign()
import objectAssign from 'object-assign';
import initialState from './initialState';

// IMPORTANT: Note that with Redux, state should NEVER be changed.
// State is considered immutable. Instead,
// create a copy of the state passed and set new values on the copy.
// Note that I'm using Object.assign to create a copy of current state
// and update values on the copy.
export default function fuelSavingsReducer(state = initialState.fuelSavings, action) {
  let newState;
  switch (action.type) {
    case SAVE_FUEL_SAVINGS:
      // For this example, just simulating a save by changing date modified.
      // In a real app using Redux, you might use redux-thunk and handle the async call in fuelSavingsActions.js
      return objectAssign({}, state, {dateModified: action.dateModified});
    case CALCULATE_FUEL_SAVINGS:
      newState = objectAssign({}, state);
      newState[action.fieldName] = action.value;
      newState.necessaryDataIsProvidedToCalculateSavings = calculator().necessaryDataIsProvidedToCalculateSavings(r
      newState.dateModified = action.dateModified;
      if (newState.necessaryDataIsProvidedToCalculateSavings) {
        newState.savings = calculator().calculateSavings(newState);
      }
      return newState;
    default:
      return state;
  }
}

```

<https://github.com/coryhouse/react-slingshot/blob/master/src/reducers/index.js>

```

import { combineReducers } from 'redux';

```

```
import fuelSavings from './fuelSavingsReducer';
import {routerReducer} from 'react-router-redux';

const rootReducer = combineReducers({
  fuelSavings,
  routing: routerReducer
});

export default rootReducer;
```

## store

---

<https://github.com/coryhouse/react-slingshot/blob/master/src/store/configureStore.js>

```
import {createStore, compose, applyMiddleware} from 'redux';
import reduxImmutableStateInvariant from 'redux-immutable-state-invariant';
import thunk from 'redux-thunk';
import rootReducer from '../reducers';

function configureStoreProd(initialState) {
  const middlewares = [
    // Add other middleware on this line...
    // thunk middleware can also accept an extra argument to be passed to each thunk action
    // https://github.com/gaearon/redux-thunk#injecting-a-custom-argument
    thunk,
  ];
  return createStore(
    rootReducer, initialState, compose(
      applyMiddleware(...middlewares)
    )
  );
}
```

```
function configureStoreDev(initialState) {
  const middlewares = [
    // Add other middleware on this line...
    // Redux middleware that spits an error on you when you try to mutate your state either inside a dispatch or
    reduxImmutableStateInvariant(),
    // thunk middleware can also accept an extra argument to be passed to each thunk action
    // https://github.com/gaearon/redux-thunk#injecting-a-custom-argument
    thunk,
  ];
  const composeEnhancers = window.__REDUX_DEVTOOLS_EXTENSION_COMPOSE__ || compose;
  // add support for Redux dev tools
  const store = createStore(
    rootReducer, initialState, composeEnhancers(
      applyMiddleware(...middlewares)
    )
  );
  if (module.hot) {
    // Enable Webpack hot module replacement for reducers
    module.hot.accept('../reducers', () => {
      const nextReducer = require('../reducers').default;
      // eslint-disable-line global-require
      store.replaceReducer(nextReducer);
    });
  }
  return store;
}

const configureStore = process.env.NODE_ENV === 'production' ? configureStoreProd : configureStoreDev;
export default configureStore;
```

## constants/

<https://github.com/coryhouse/react-slingshot/blob/master/src/constants/actionTypes.js>

```
export const SAVE_FUEL_SAVINGS = 'SAVE_FUEL_SAVINGS';
export const CALCULATE_FUEL_SAVINGS = 'CALCULATE_FUEL_SAVINGS';

// obj & auto assign
const ConstantsOBJ = {
  SAVE_FUEL_SAVINGS,
  CALCULATE_FUEL_SAVINGS
};
export ConstantsOBJ;
```

## containers

---

<https://github.com/coryhouse/react-slingshot/blob/master/src/containers/FuelSavingsPage.js>

```
import React from 'react';
import PropTypes from 'prop-types';
import {connect} from 'react-redux';
import {bindActionCreators} from 'redux';
import * as actions from '../actions/fuelSavingsActions';
import FuelSavingsForm from '../components/FuelSavingsForm';

export const FuelSavingsPage = (props) => {
  return (
    <FuelSavingsForm
      saveFuelSavings={props.actions.saveFuelSavings}
      calculateFuelSavings={props.actions.calculateFuelSavings}
      fuelSavings={props.fuelSavings}
    />
  );
};
```

```

FuelSavingsPage.propTypes = {
  actions: PropTypes.object.isRequired,
  fuelSavings: PropTypes.object.isRequired
};

function mapStateToProps(state) {
  return {
    fuelSavings: state.fuelSavings
  };
}

function mapDispatchToProps(dispatch) {
  return {
    actions: bindActionCreators(actions, dispatch)
  };
}

export default connect(
  mapStateToProps,
  mapDispatchToProps
)(FuelSavingsPage);

```

## utils

---

<https://github.com/coryhouse/react-slingshot/blob/master/src/utils/dateHelper.js>

```

export function getFormattedDateTime(date = new Date()) {
  return `${date.getMonth() + 1}/${date.getDate()} ${date.getHours()}:${padLeadingZero(date.getMinutes())}:${padLeadingZero(date.getSeconds())}`;
}

export function padLeadingZero(value) {
  return value > 9 ? value : `0${value}`;
}

```

## routes

---

<https://github.com/coryhouse/react-slingshot/blob/master/src/routes.js>

```
import React from 'react';
import { Route, IndexRoute } from 'react-router';

import App from './components/App';
import HomePage from './components/HomePage';
import FuelSavingsPage from './containers/FuelSavingsPage'; // eslint-disable-line import/no-named-as-default
import AboutPage from './components/AboutPage';
import NotFoundPage from './components/NotFoundPage';

export default (
  <Route path="/" component={App}>
    <IndexRoute component={HomePage}/>
    <Route path="fuel-savings" component={FuelSavingsPage}/>
    <Route path="about" component={AboutPage}/>
    <Route path="*" component={NotFoundPage}/>
  </Route>
);
```

## router v4 ???

no `IndexRoute` any more!

## styles

---

<https://github.com/coryhouse/react-slingshot/blob/master/src/styles/styles.scss>



```
/* Variables */
$vin-blue: #5bb7db;
$vin-green: #60b044;
$vin-red: #ff0000;

/*

// native css var

_css-var-color: rgba(0, 0, 0, 0.7);

*/

/* Styles */
body {
  font: 14px 'Helvetica Neue', Helvetica, Arial, sans-serif;
  line-height: 1.4em;
  color: #4d4d4d;
  min-width: 230px;
  max-width: 550px;
  margin: 0 auto;
  -webkit-font-smoothing: antialiased;
  -moz-font-smoothing: antialiased;
  font-smoothing: antialiased;
  font-weight: 300;
}
td {
  padding: 12px;
}
h2 {
  color: $vin-blue;
}

.savings {
  color: $vin-green;
```

```
}  
.loss {  
  color: $vin-red;  
}  
input.small {  
  width: 46px;  
}  
td.fuel-savings-label {  
  width: 175px;  
}
```

## components

---

UI

<https://github.com/coryhouse/react-slingshot/tree/master/src/components>

