

Redux Saga

Middleware story continues...

Redux - Middleware - Redux Saga

- A library that aims to make ***side effects*** (i.e. asynchronous things like data fetching and impure things like accessing the browser cache) in React/Redux applications easier and better
- The mental model is that a saga is like a separate thread in your application that's solely responsible for side effects. ***redux-saga is a redux middleware***, which means this thread can be started, paused and cancelled from the main application with normal redux actions, it has access to the full ***redux application state*** and it ***can dispatch redux actions*** as well
- It uses an ES6 feature called ***Generators*** to make those ***asynchronous flows easy to read, write and test***

Redux - Middleware - Redux Saga Setup



```
npm install redux-saga --save
```

Redux - Middleware - Redux Saga In Action

- Create sagas.js

Create saga middleware

```
import createSagaMiddleware from 'redux-saga'  
const sagaMiddleware = createSagaMiddleware();
```

- Run middleware by configuring the watchers

```
import {getReposWatcher} from './sagas'  
sagaMiddleware.run(getReposWatcher);
```

Saga API

- Middleware API
 - createSagaMiddleware(options)
 - middleware.run(saga, ...args)
- Saga Helpers
 - takeEvery(pattern, saga, ...args)
 - takeLatest(pattern, saga, ..args)
 - throttle(ms, pattern, saga, ..args)
- Effect creators
 - take(pattern)
 - put(action)
 - call(fn, ...args)
 - fork(fn, ...args)
 - spawn(fn, ...args)
 - cancel(task)
 - select(selector, ...args)
- Effect combinators
 - race(effects)
 - all(effects)