



Ka San @quangtin12b5

Follow

★ 311

4

20

Published Aug 1st, 2018 10:21 AM - 3 min read

🕒 2.0K

💬 2

📎 2

Redux saga in react native

React Native

ReactJS

redux

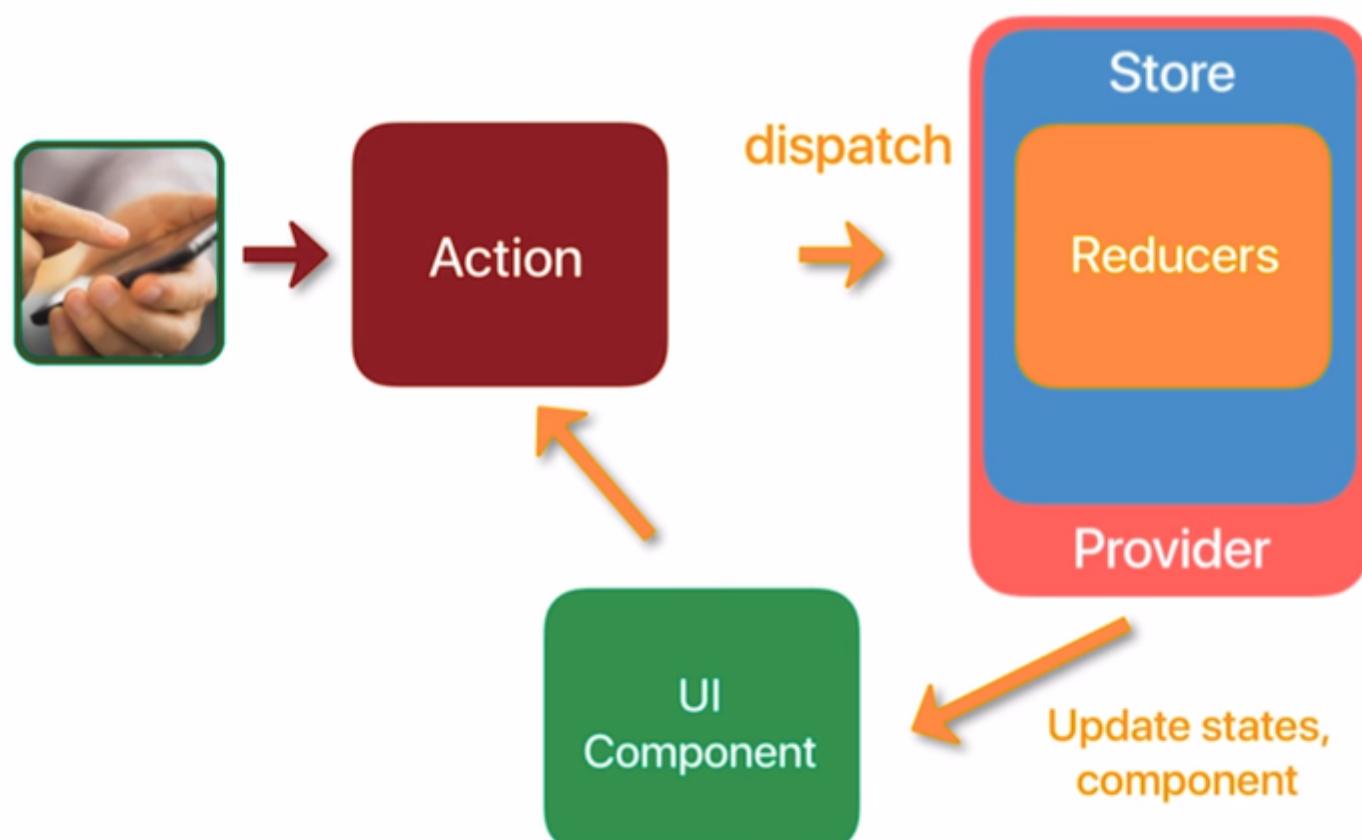
• • •

Redux

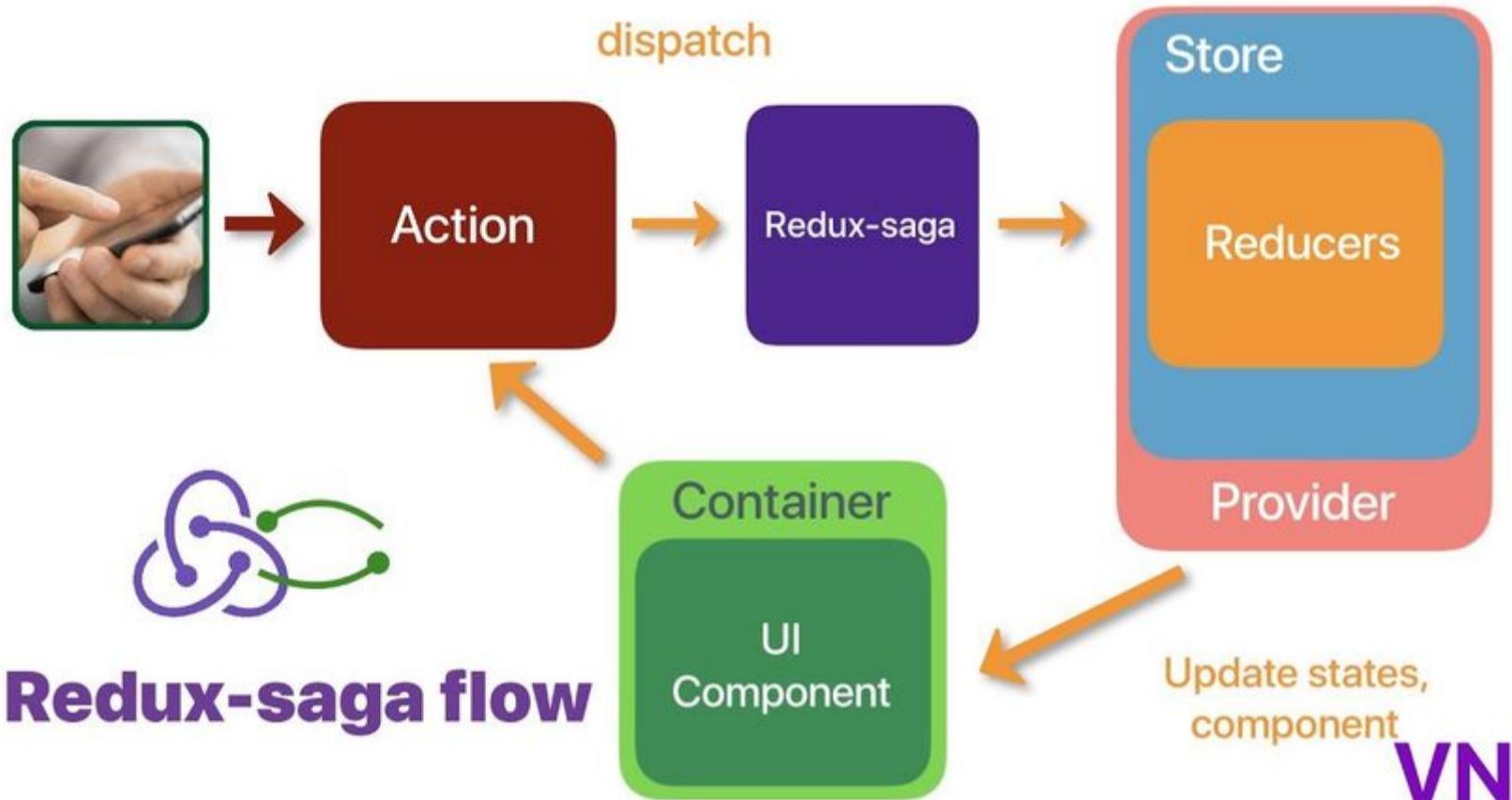
Redux là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng. Được dựa trên nền tảng tư tưởng của kiến trúc Flux do Facebook giới thiệu, do vậy Redux thường là bộ đôi kết hợp hoàn hảo với React (React Js và React Native) và đương nhiên khi nhắc đến reactjs điều đầu tiên người ta nhớ đến đó chính là redux.

Một số khái niệm:

- Actions: các khối thông tin (payloads) gửi dữ liệu từ ứng dụng của bạn đến Store. Chúng là nguồn thông tin duy nhất cho Store. Bạn có thể gửi chúng đến Store bằng cách sử dụng store.dispatch().
- Reducers: Lắng nghe các hành động (Actions) và thực hiện các thay đổi trên các giá trị của Store. Hãy nhớ rằng các hành động chỉ mô tả những gì đã xảy ra, nó không thay đổi dữ liệu trên Store.
- Store: Quản lý nhiều reducers.
- Provider: là component chứa nhiều store.
- Tham khảo : <https://redux.js.org/basics>



- Redux-Saga là một thư viện redux middleware, giúp quản lý những side effect trong ứng dụng redux trở nên đơn giản hơn. Bằng việc sử dụng tối đa tính năng Generators (function*) của ES6, nó cho phép ta viết async code nhìn giống như là synchronous.
- Hoạt động: nằm giữa Action và Provider, nó sẽ thực hiện một hành động nào đó (vd: gửi các request đến server nào đó) rồi nhận dữ liệu sau đó effects qua Provider.
- Tham khảo: <https://redux-saga.js.org/>



Build project

Bây giờ làm một ví dụ để hiểu rõ hơn nhé 😊

Cài đặt và tạo project mình đã hướng dẫn: <https://viblo.asia/p/react-native-LzD5d6kdZjY>

Mình sẽ làm chức năng đếm số, sử dụng một số thư viện sau:

```
"react-native-button": "^2.3.0",
"react-redux": "^5.0.7",
"redux": "^4.0.0",
"redux-saga": "^0.16.0"
```

- Tạo file actionTypes chứa hai hằng số tăng, giảm

src/actions/actionTypes.js

VIBLO

Search Viblo



Sign In/Sign up

- Tạo hàm increaseAction, decreaseAction src/actions/index.js

```

import { INCREMENT, DECREMENT } from './actionTypes';

// Tăng với giá trị step
export const increaseAction = (step) => {
    return {
        type: INCREMENT,
        step: step
    }
}

// Giảm với giá trị step
export const decreaseAction = (step) => {
    return {
        type: DECREMENT,
        step: step
    }
}

```

- Cho phép component CounterContainer có thể truy cập để dispatch như là props

src/containers/CounterContainer.js

```

import { connect } from 'react-redux';
import CounterComponent from '../components/CounterComponent';

//Actions
import { increaseAction } from '../actions';
import { decreaseAction } from '../actions';

const mapStateToProps = (state) => {
    return {
        times: state.counterReducers ? state.counterReducers : 0
    }
};

const mapDispatchToProps = (dispatch) => {
    return {
        onDecrement: (step) => {
            dispatch(decreaseAction(step));
        },
        onIncrement: (step) => {
            dispatch(increaseAction(step));
        }
    };
}
const CounterContainer = connect(mapStateToProps, mapDispatchToProps)(CounterComponent);
export default CounterContainer;

```

- Tạo một hàm để lắng nghe các actions và không thay đổi state

src/reducers/CounterReducers.js

```
import { DECREMENT, INCREMENT } from '../actions/actionTypes';

//state không thay đổi, chỉ trả về giá trị cuối cùng
const counterReducers = (times = 0, action) => {
    switch (action.type) {
        case INCREMENT:
            return times + action.step;
        case DECREMENT:
            return times - action.step;

        default:
            return times;
    }
}

export default counterReducers;
```

- Đưa tất cả reducers vào store bằng cách dùng combineReducers sẻ giúp việc gộp nhiều reducers lại

src/reducers/index.js

```
import { combineReducers } from 'redux';
import counterReducers from './CounterReducers';

const allReducers = combineReducers({
    counterReducers,
});
export default allReducers;
```

- Tạo Component chứa 2 button tăng, giảm giá trị

src/components/CounterComponent.js

```

import React, { Component } from 'react';
import Button from 'react-native-button';

import { Text, View, } from 'react-native';

export default class CounterComponent extends Component {
  render() {
    return (
      <View style={{ flex: 1, marginTop: 30 }}>
        <Text style={{ margin: 20, fontWeight: 'bold', color: 'forestgreen', fontSize: 20 }}>
          Redux Sagas
        </Text>
        <View style={{ height: 50, margin: 10, flexDirection: 'row' }}>
          <Button
            containerStyle={{ padding: 10, height: 45, borderRadius: 10, backgroundColor: 'darkviolet' }}
            style={{ fontSize: 18, color: 'white' }}
            onPress={() => {
              this.props.onDecrement(1);
            }}
          >
            Decrement -
          </Button>
          <Button
            containerStyle={{ padding: 10, height: 45, borderRadius: 10, backgroundColor: 'darkviolet' }}
            style={{ fontSize: 18, color: 'white' }}
            onPress={() => {
              this.props.onIncrement(1)
            }}
          >
            Increment +
          </Button>
        </View>
        <Text style={{ margin: 10, fontWeight: 'bold', color: 'darkblue', fontSize: 17 }}>
          Counts: {this.props.times}
        </Text>
      </View>
    );
  }
}

```

- Đưa tất cả store vào Provider

App.js

```

import React, { Component } from 'react';
//Redux
import { createStore } from 'redux';
import { Provider } from 'react-redux';

import allReducers from './src/reducers';
import CounterContainer from './src/containers/CounterContainer';

let store = createStore(allReducers);
const App = () => (
  <Provider store={store}>
    <CounterContainer />
  </Provider>
);

export default class AppSaga extends Component {
  render() {
    return <App/>;
  }
}

```

index.js

```
import { AppRegistry } from 'react-native';
import App from './App';

AppRegistry.registerComponent('demo_redux_saga', () => App);
```

Đó là cách dùng redux, bây giờ mình sẽ sử dụng redux saga

- Tạo các hàm saga, để ý hàm thì có thêm * nhé

src/sagas/counterSagas.js

```
import { INCREMENT, DECREMENT} from '../actions/actionTypes';

//takeEvery khi thực hiện một hành động nào đó nó sẽ gọi một hàm trước khi dispatch vào reducers
import { takeEvery } from 'redux-saga/effects';

function* increment() {
  console.log(`This is increment saga`);
}

export function* watchIncrement() {
  yield takeEvery(INCREMENT, increment);
}

function* decrement() {
  console.log(`This is decrement saga`);
}

export function* watchDecrement() {
  yield takeEvery(DECREMENT, decrement);
}
```

- Tạo root chứa tất cả hàm saga

src/sagas/rootSaga.js

```
import { all } from 'redux-saga/effects';

import { sayHello } from './counterSagas';
import { watchIncrement, watchDecrement } from './counterSagas';

export default function* rootSaga() {
  yield all([
    watchIncrement(),
    watchDecrement(),
  ]);
}
```

- Sửa lại file App.js

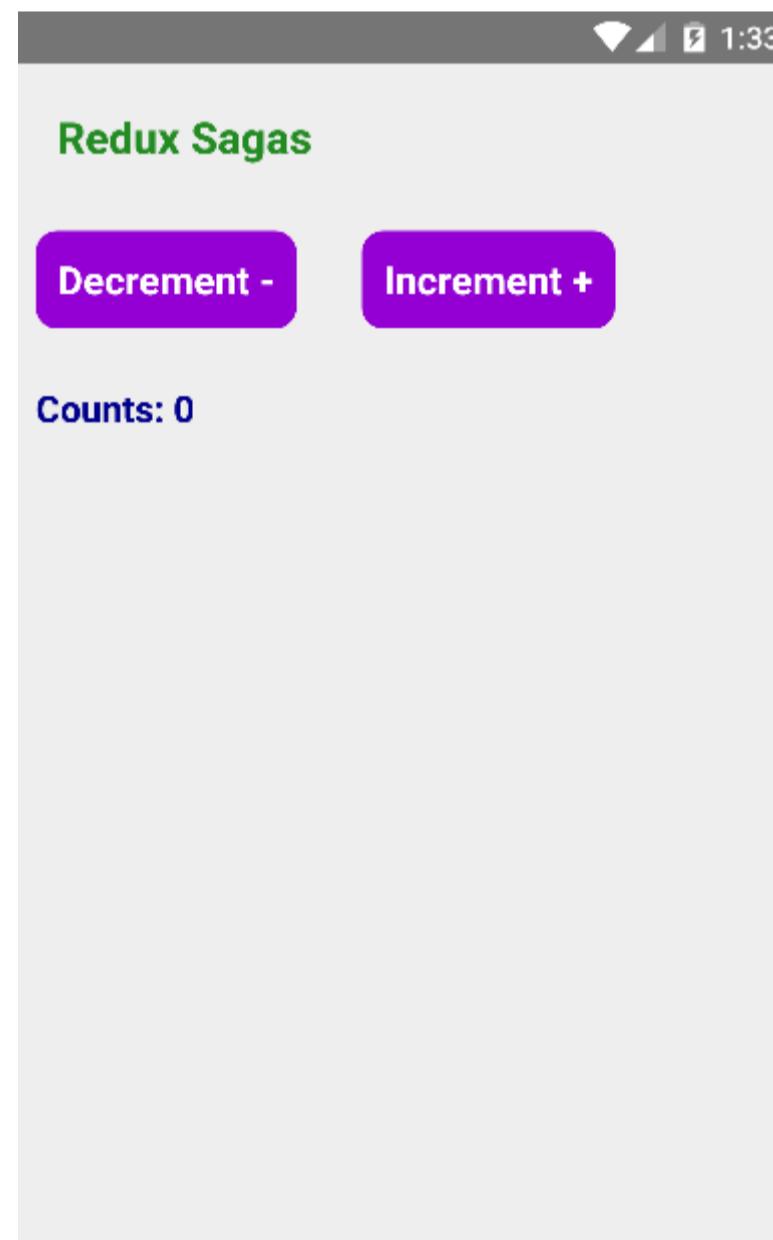
```
import React, { Component } from 'react';
//Redux
import { createStore, applyMiddleware } from 'redux';
import { Provider } from 'react-redux';

import allReducers from './src/reducers';
import CounterContainer from './src/containers/CounterContainer';
//Redux saga
import createSagaMiddleware from 'redux-saga';
import rootSaga from './src/sagas/rootSaga';
//Middleware
const sagaMiddleware = createSagaMiddleware();
//Tùy applyMiddleware vào Reducers thì tạo một store, sagaMiddleware nằm giữa Action và Reducers.
let store = createStore(allReducers, applyMiddleware(sagaMiddleware));
const App = () => (
  <Provider store={store}>
    <CounterContainer />
  </Provider>
);

export default class AppSaga extends Component {
  render() {
    return <App/>;
  }
}

sagaMiddleware.run(rootSaga); //Chạy xuyên suốt các hàm rootSaga trong app
```

Kết quả



Vậy là đã xong, mình mới tìm hiểu, có chỗ nào không đúng mong các bạn góp ý..

Chúc các bạn làm thành công (~.~)

<https://github.com/reduxjs/react-redux/blob/master/docs/api.md>

<https://github.com/redux-saga/redux-saga/blob/master/docs/api/README.md>



Related

[Redux cho người mới bắt đầu - Part 2 First Project](#)

Pham Van Duc

6 min read

👁 15004 🗣 27 💬 3 ⚑ 49

[Redux cho người mới bắt đầu - Part 3 Middleware](#)

Pham Van Duc

5 min read

👁 12158 🗣 22 💬 7 ⚑ 26

[Bắt đầu thực hiện một ví dụ CRUD với Redux](#)

Bùi Hiếu

5 min read

👁 1088 🗣 4 💬 18 ⚑ 11

[\[RN\]\[Redux\] Using redux in react-native programing](#)

Nguyen Viet Manh

3 min read

👁 1932 🗣 3 💬 0 ⚑ 4

More from Ka San

[Tab, Drawer Navigator in react native](#)

Ka San

6 min read

👁 1773 🗣 0 💬 0 ⚑ 1

[App Movies with react native](#)

Ka San

1 min read

👁 438 🗣 2 💬 2 ⚑ 6

[React Native Firebase](#)

Ka San

1 min read

👁 1150 🗣 3 💬 2 ⚑ 4

[Install React Native](#)

Ka San

1 min read

👁 566 🗣 0 💬 0 ⚑ 7

Comments

Login to comment

[Tran Cong Trinh](#) @trantrinh1102

Aug 1st, 2018 6:18 PM

Hay quá em ơi

▲ 0 ▼ | Reply Share ⚑ 0

Ka San @quangtin12b5

Aug 1st, 2018 10:50 PM



▲ 0 ▼ | Reply Share ⚑ 0

[Videos](#)[Authors](#)[Discussions](#)[Recommend System](#)[Tools](#)[Machine Learning](#)[System Status](#)**LINKS**

© 2019 Viblo. All rights reserved.

[Feedback](#) [Help](#) [FAQs](#) [RSS](#) [Terms](#)