

React Native Redux Middleware

Isumi
Batam, Jan 2020

GOALS

1. Intro about Redux Saga
2. How Redux Saga Works
3. Implementation Redux Saga



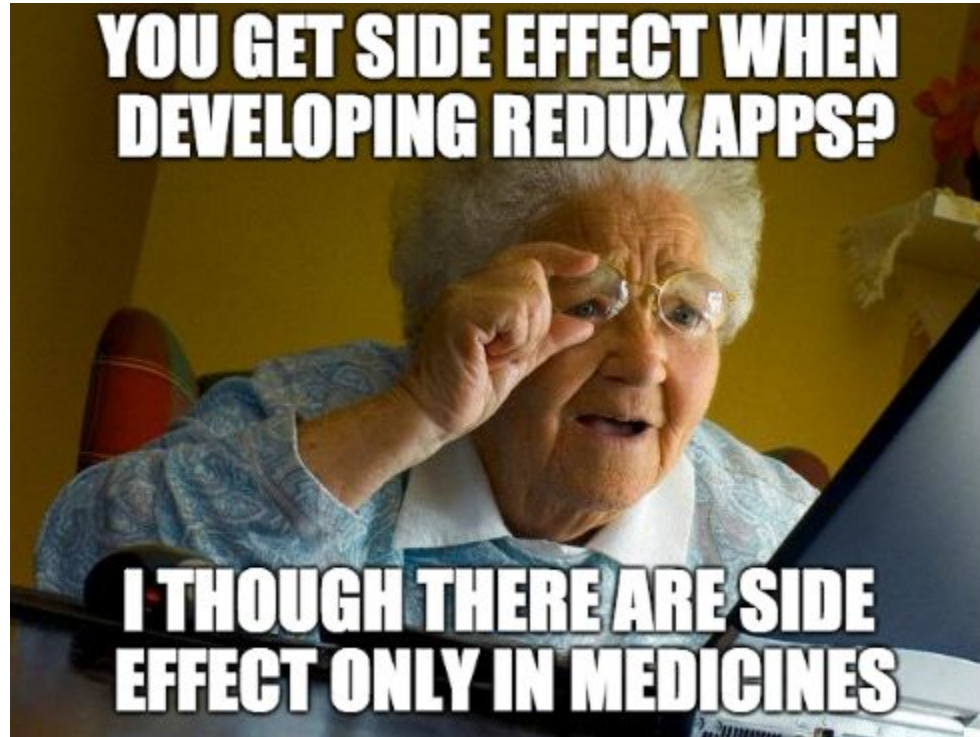
1. Intro about REDUX SAGA

Redux-saga is a redux middleware library, that is designed to make handling side effects in your redux app nice & simple. It achieves this by leveraging an ES6 feature called Generators, allowing us to write asynchronous code that looks synchronous, and is very easy to test.



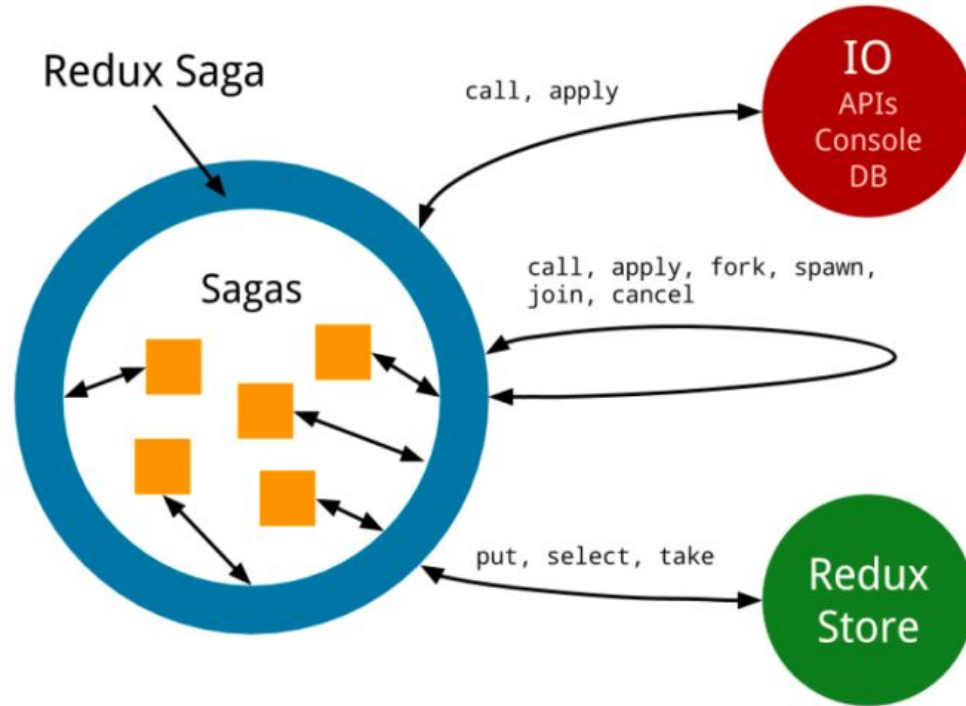
Benefits?

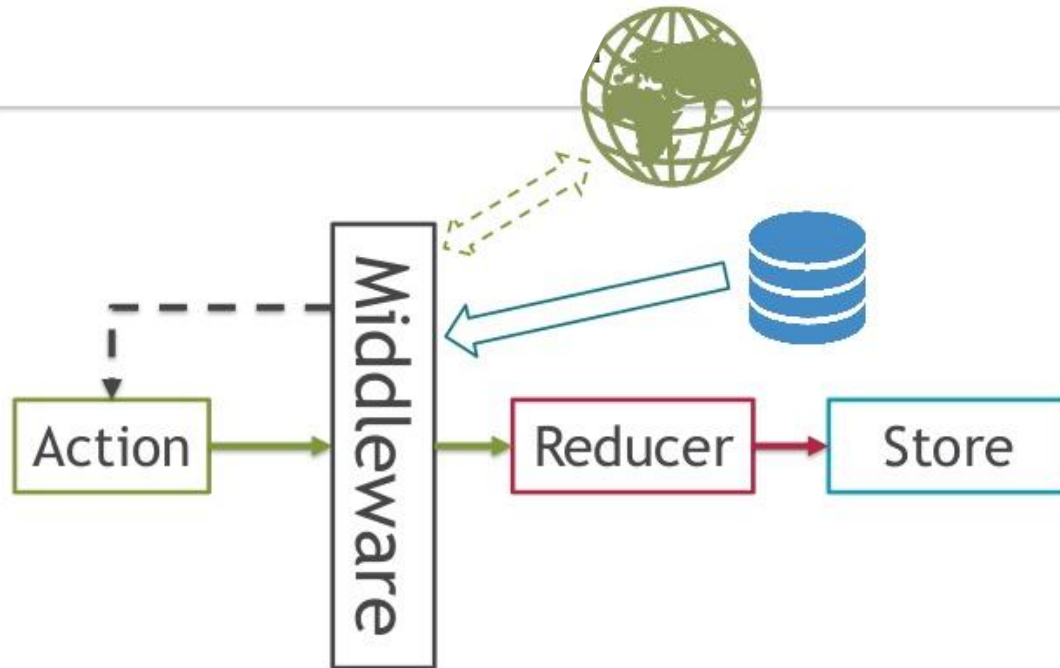
- 1. Do synchronous job smoothly**
- 2. Maintainability**





2. How Redux Saga Works







3. Implementation Redux

Install (Redux Basic):
`npm i redux react-redux`

- redux
- react-redux
- redux-saga





./store.js

```
import { createStore, applyMiddleware } from 'redux';
import createSagaMiddleware from 'redux-saga';
import rootReducer from './src/redux/reducer/index';
import IndexSaga from './src/redux/saga/index';

const configureStore = () => {
  const sagaMiddleware = createSagaMiddleware();
  return {
    ...createStore(rootReducer,
    applyMiddleware(sagaMiddleware)),
    runSaga: sagaMiddleware.run(IndexSaga)
  };
};

export default configureStore;
```



./src/redux/saga/index.js

```
import { all, takeLatest } from 'redux-saga/effects';
import { REQUEST_USER_DATA } from '../type/UserType';
import { getUserData } from '../user';

export default function* IndexSaga() {
  yield all([takeLatest(REQUEST_USER_DATA, getUserData)]);
}
```



./src/redux/saga/user.js

```
import { put, call } from 'redux-saga/effects'
import { SHOW_USER_DATA, FAILED_GET_USER_DATA, SUCCESS_USER_DATA }
  from '../type/UserType';
import { apiGetUser } from '../api/apiUser';

export function* getUserData(action) {
  try {
    const { token } = action;
    const users = yield call(apiGetUser, token)
    console.log(action)
    yield put({ type: SHOW_USER_DATA, payload: users })
    yield put({ type: SUCCESS_USER_DATA, payload: 'Successfully get ' })
  } catch (error) {
    yield put({ type: FAILED_GET_USER_DATA, payload: 'Fatal ERROR' })
    console.log(error)
  }
}
```



./src/redux/saga/api/apiUser.js

```
import axios from 'axios';

export const apiGetUser = token => {
  console.log('token nya ', token);
  let headers = {
    Authorization: token
  };
  return axios
    .get(
      'http://ec2-3-0-98-199.ap-southeast-1.compute.amazonaws.com:3000/users/',
      { headers: headers }
    )
    .then(function(res) {
      return res.data.results;
    });
};
```



./src/redux/type/userType.js

```
export const REQUEST_USER_DATA = 'REQUEST_USER_DATA'
export const SHOW_USER_DATA = 'SHOW_USER_DATA'
export const FAILED_GET_USER_DATA = 'FAILED_GET_USER_DATA'
export const SUCCESS_USER_DATA = 'SUCCESS_USER_DATA'
```



./src/redux/action/user.js (create file)

```
import { REQUEST_USER_DATA } from '../type/UserType'

export const getUser = (data) => {
  return {
    type : REQUEST_USER_DATA,
    token : data
  }
}
```



./src/redux/reducer/index.js (edit file)

```
import { combineReducers } from 'redux'
import user from './user'
import auth from './auth'
const IndexReducer = combineReducers ({
  user : user,
  auth : auth
})

export default IndexReducer
```



./src/redux/reducer/user.js

```
import {
  REQUEST_USER_DATA,
  SHOW_USER_DATA,
  FAILED_GET_USER_DATA, SUCCESS_USER_DATA } from '../type/UserType'

const initialState = {
  loading : true,
  data : [],
  message : null
}

export default (state = initialState, action) => {
  switch (action.type) {
    case REQUEST_USER_DATA :
      return {
        ...state,
        loading : true
      }
  }
}
```

//to be continued...



./src/redux/reducer/user.js (continue)

```
case SHOW_USER_DATA :  
    return {  
        ...state,  
        loading : false,  
        data : action.payload  
    }  
case FAILED_GET_USER_DATA :  
    return {  
        ...state,  
        loading : false,  
        message : action.payload  
    }
```

```
case SUCCESS_USER_DATA :  
    return {  
        ...state,  
        loading : false,  
        message : action.payload  
    }  
default :  
    return state  
}
```



Udah deh tinggal DISPATCH aja Kk..



dan rasakanlah efek sampingnya...

Exercise

Refactor your Profile App using Redux Saga ;)

