Sử dụng redux saga trong reactjs

nguyen.le ,  15 min read

Nội dung bài viết

* [**Giới thiệu nội dung bài viết**](https://levunguyen.com/laptrinhjavascript/2021/12/15/su-dung-redux-saga-trong-reactjs/#gi%E1%BB%9Bi-thi%E1%BB%87u-n%E1%BB%99i-dung-b%C3%A0i-vi%E1%BA%BFt)
* [**1. Set up redux và tích hợp redux-saga**](https://levunguyen.com/laptrinhjavascript/2021/12/15/su-dung-redux-saga-trong-reactjs/#1-set-up-redux-v%C3%A0-t%C3%ADch-h%E1%BB%A3p-redux-saga)
* [**2. Tích hợp Axios để call APi**](https://levunguyen.com/laptrinhjavascript/2021/12/15/su-dung-redux-saga-trong-reactjs/#2-t%C3%ADch-h%E1%BB%A3p-axios-%C4%91%E1%BB%83-call-api)
* [**3. Render kết quả**](https://levunguyen.com/laptrinhjavascript/2021/12/15/su-dung-redux-saga-trong-reactjs/#3-render-k%E1%BA%BFt-qu%E1%BA%A3)
* [**4. Video Demo**](https://levunguyen.com/laptrinhjavascript/2021/12/15/su-dung-redux-saga-trong-reactjs/#4-video-demo)

**Giới thiệu nội dung bài viết**

**Chào các em, hôm nay anh sẽ hướng dẫn mọi người hiểu về Redux Saga trong reactjs.**

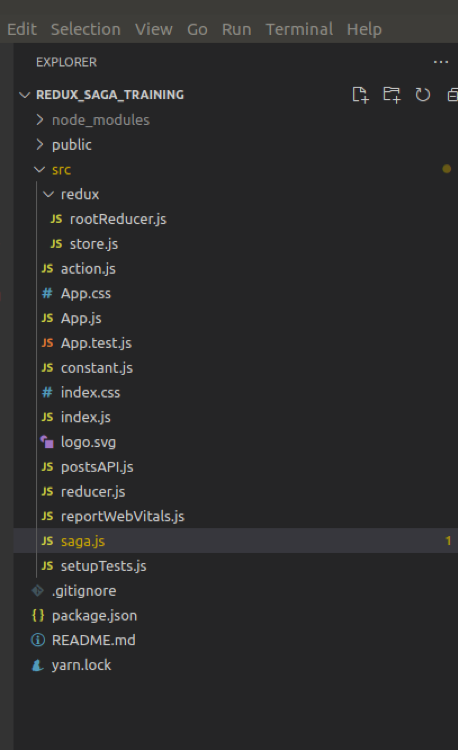
* **Redux saga về bản chất nó là một middleware library để handle các side effect trong trong ReactJS (Side effect các bạn có thể hiểu đơn giản là các xử lý bất đồng bộ như là call API, sử dụng setTimeout, setInterval,…).**
* **Redux saga được xây dựng dựa trên các javascript generator function, vì thế các bạn có thể tìm hiểu thêm ở link này để hiểu thêm về loại function này nhé, nó khác gì so với các function bình thường.**
* **Đối với generator function, nó có một vài helper mà mình cần phải biết trước khi bắt tay vào thực hành.**
* **takeEvery() : thực thi và trả lại kết quả của mọi actions được gọi.**
* **takeLastest() : có nghĩa là nếu chúng ta thực hiện một loạt các actions, nó sẽ chỉ thực thi và trả lại kết quả của của actions cuối cùng.**
* **take() : tạm dừng cho đến khi nhận được action.**
* **put() : dispatch một action.**
* **call(): gọi function. Nếu nó return về một promise, tạm dừng saga cho đến khi promise được giải quyết.**
* **race() : chạy nhiều effect đồng thời, tuy nhiên chỉ lấy kết quả của effect nhanh nhất, và hủy kết quả của những effect còn lại. Các bạn có thể hình dung như một cuộc đua, và nó chỉ qua tâm đến người thắng cuộc.**

**Ok mình sẽ bắt tay vào thực hành!**

**1. Set up redux và tích hợp redux-saga**

**Bây giờ chúng ta sẽ tạo ra một ứng dụng nho nhỏ show ra list các bài Post lấy data từ APi sử dụng Redux-saga kết hợp với axios nhé.**

**Mình sẽ lần lượt tạo các file như bên dưới. Đầu tiên mình sẽ tạo ra 3 file đại diện cho 3 thành phần cơ bản của redux nhé, bao gồm: action.js, constant.js, reducer.js.**

****

* **File action.js**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | import {GET\_LIST\_POST, GET\_LIST\_POST\_SUCCESS} from './constant';  export const getListPost = (payload) => {  return {  type: GET\_LIST\_POST,  payload,  }  }  export const getListPostSuccess = (payload) => {  return {  type: GET\_LIST\_POST\_SUCCESS,  payload,  }  } |

* **File constant.js**

|  |  |
| --- | --- |
| 1  2 | export const GET\_LIST\_POST = 'GET\_LIST\_POST';  export const GET\_LIST\_POST\_SUCCESS = 'GET\_LIST\_POST\_SUCCESS'; |

* **File reducer.js**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | const postsReducer = (state = INITIAL\_STATE, action) => {  switch(action.type) {  case GET\_LIST\_POST:  return {  ...state,  load: true,  };  case GET\_LIST\_POST\_SUCCESS:  const {data} = action.payload;  return {  ...state,  posts: data,  load: false,  };  default:  return state;  }  }  export default postsReducer; |

**Tiếp theo, mình sẽ tạo phần config cho redux, các bạn tạo một thư mục là redux, trong đó chứa 2 file là store.js và rootReducer.js**

* **File store.js**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | import { createStore, applyMiddleware } from 'redux';  import createSagaMiddleware from 'redux-saga';  import rootReducer from './rootReducer';  import postsSaga from '../saga';  const sagaMiddleware = createSagaMiddleware();  const store = createStore(rootReducer, applyMiddleware(sagaMiddleware));  sagaMiddleware.run(postsSaga);  export default store; |

* **File rootReducer.js**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | import { combineReducers } from 'redux';  import postsReducer from '../reducer';  const rootReducer = combineReducers({  posts: postsReducer,  });  export default rootReducer; |

**Ở file store js mình đã tiến hành tích hợp redux saga vào, các bạn ra teminal chạy lệnh npm install redux-saga hoặc lệnh yarn add redus-saga để cài đặt redux-saga nhé. Tiếp theo tại file index.js mình import thằng Provider của Redux vào và wrap nó ở component App như bình thường nhé!**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | import React from 'react';  import ReactDOM from 'react-dom';  import './index.css';  import App from './App';  import reportWebVitals from './reportWebVitals';  import { Provider } from 'react-redux';  import store from './redux/store';  ReactDOM.render(  <Provider store={store}>  <React.StrictMode>  <App />  </React.StrictMode>  </Provider>,  document.getElementById('root')  );  // If you want to start measuring performance in your app, pass a function  // to log results (for example: reportWebVitals(console.log))  // or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals  reportWebVitals(); |

* **Tiếp theo, tạo file saga.js**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | import { call, put, takeLatest } from 'redux-saga/effects';  import {getPostData} from './postsAPI';  import {getListPostSuccess} from './action';  function\* getListPostSaga(action) {  try {  const data = yield call(getPostData);  yield put(getListPostSuccess(data));  } catch (error) {  //handle error  }  }  function\* postsSaga() {  yield takeLatest('GET\_LIST\_POST', getListPostSaga);  }  export default postsSaga; |

**Nhiệm vụ của file này là lắng nghe nếu có action getListPost được dispatch, thì nó bắt lấy để handle việc call API, khi call API thì thường sẽ có 2 trường hợp xảy ra trường hợp như khối try catch ở trên, nếu call success thì nó vô khối try, lúc này mình dispatch một action mới là getListPostSuccess để cập nhật data get được từ APi vô redux store, trong trường hợp call error thì nó sẽ vô khối catch (vấn đề mình xin trình bày trong một bài khác).**

**2. Tích hợp Axios để call APi**

**Để handle việc call API, chúng ta sẽ tạo ra một file là postAPI.js**

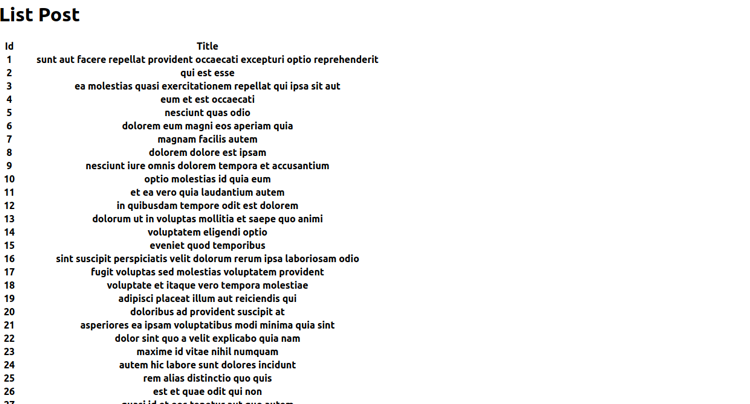
|  |  |
| --- | --- |
| 1  2  3  4  5 | import axios from 'axios';  export const getPostData = () => {  return axios.get('https://jsonplaceholder.typicode.com/posts');  } |

**Trong file này mình sử dụng một HTTP client dựa trên promise để call API đó là Axios, Chạy lệnh npm install –save axios hoặc yarn add axios để cài đặt thư viện.**

**3. Render kết quả**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52 | import React from 'react';  import { connect } from 'react-redux';  import {getListPost as getListPostAction} from './action';  class App extends React.Component {  render() {  const {posts, load} = this.props.posts;  if(load) {  return (  <h1>Data is loading from API...</h1>  )  }  return (  <h1>List Post</h1>  <table>  <tbody>  <tr>  <th>Id</th>  <th>Title</th>  </tr>  {  posts.map((post) => (  <tr>  <th>{post.id}</th>  <th>{post.title}</th>  </tr>  ))  }  </tbody>  </table>  </>  );  }  componentDidMount() {  this.props.getListPost();  }  }  const mapStateToProps = (state) => {  return {  posts: state.posts,  }  }  const mapDispatchToProps = (dispatch) => {  return {  getListPost: (params) => dispatch(getListPostAction(params)),  }  }  export default connect(mapStateToProps, mapDispatchToProps)(App); |

**Ở phần render ra kết quả này mọi người có thể thấy nếu đang call API tức là biến load đang là true thì nó sẽ show ra một đoạn text “Data is loading from API…”, trong trường hợp call API xong rồi thì biến load sẽ được set thành false và show ra kết quả như bên dưới.**

****

**Source code: https://github.com/Thanhdong0909/redux\_saga\_training**

**4. Video Demo**