FPT Education

FPT POLYTECHNIC

cdio™

Conceive Design Implement Operate

THỰC HỌC – THỰC NGHIỆP

LẬP TRÌNH FRONT-END FRAMEWORK 2

REDUX – REACT VỚI REDUX REDUX THUNK

http://www.poly.edu.vn

Mục tiêu

- Redux
- React với redux
- Redux thunk

Phần 1
Redux
React với redux

REDUX

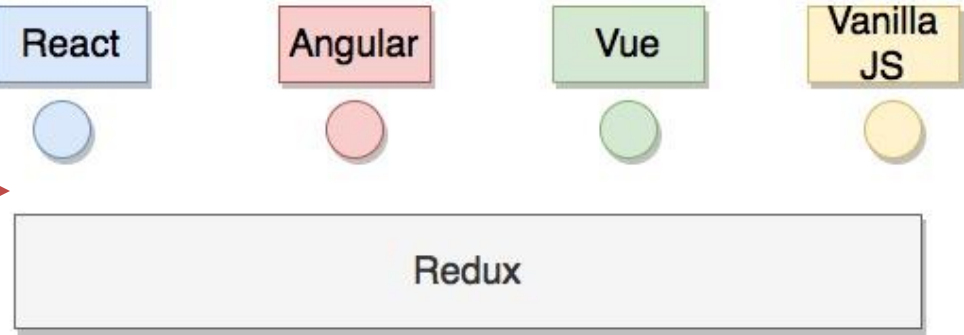
Redux là gì

Thư viện js quản lý state

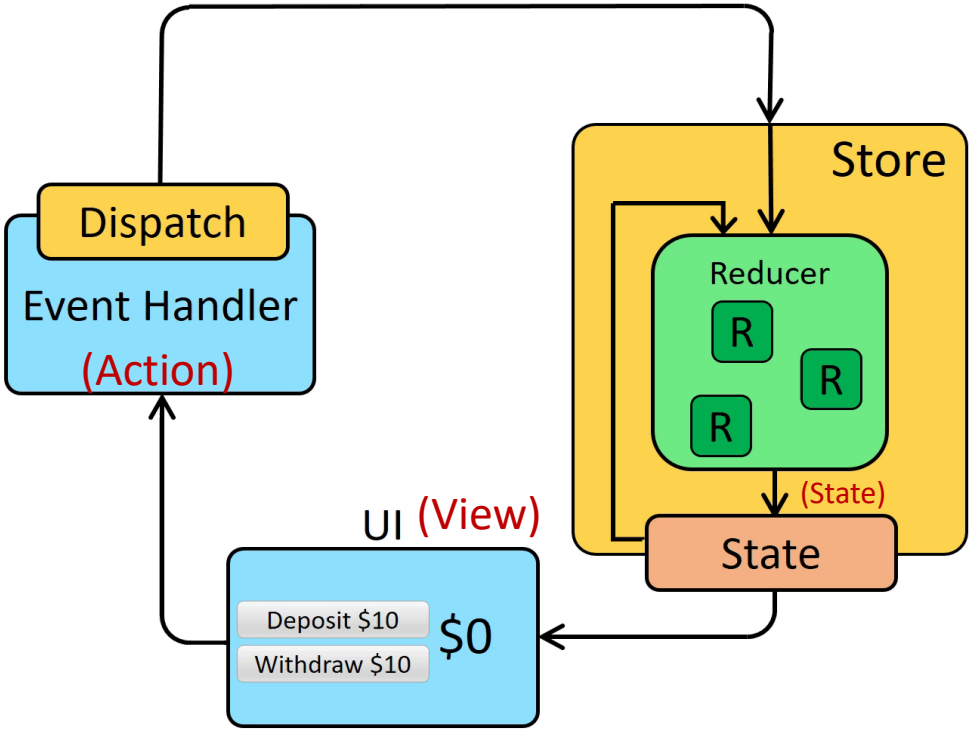
Sử dụng kiến trúc uni-directional data flow

Giúp tạo các ứng dụng phức tạp dễ dàng hơn

không được thiết kế chi tiết để làm việc với React



REDUX – DATA FLOW



View: mô tả giao diện người dùng dựa vào state hiện tại
Action: sự kiện xảy ra trong ứng dụng dựa vào user input

Actions

- Action như 1 event mô tả điều gì đó (what) xảy ra trong ứng dụng
- Là một javascript object thuần tuý, có thêm trường ***(field) type***.
- Ví dụ

```
const createBag = (withdraw) => {
  return {
    type: 'deposit',
    payload:{
      withdraw:withdraw
    }
  };
};
```

Reducers

- như là 1 event listener xử lý các event dựa vào ***received action (event) type***
- là 1 function, nhận current state và action object, cập nhật state nếu cần và trả về state mới `(state, action) => newState`
- Ví dụ

```
//reducers
const bagReducers = (moneyofbag = 100, action) => {
  if (action.type === 'deposit') {
    return action.payload.withdraw + moneyofbag;
  }

  return moneyofbag;
};
```

- Luật khi tạo reducers
  - Chỉ nên tính toán giá trị của new state dựa vào đối số state và action
  - ***Không được phép sửa đổi state hiện có***, sao chép state hiện có và thực hiện các thay đổi đối với giá trị sao chép

- Store
  - Được tạo ra bằng cách truyền vào reducer
  - Dùng phương thức getState để trả về current state.
- Dispatch
  - Là method của redux store
  - Cách duy nhất để cập nhật state: `store.dispatch()`
- Ví dụ

```
const {createStore, combineReducers} = Redux;
const claimDepartments = combineReducers ({
  moneyofbag: bagReducers,
});

const store = createStore(claimDepartments);
store.dispatch(createBag(100));
console.log(store.getState());
```

REACT VỚI REDUX

Để react và redux làm việc cùng nhau

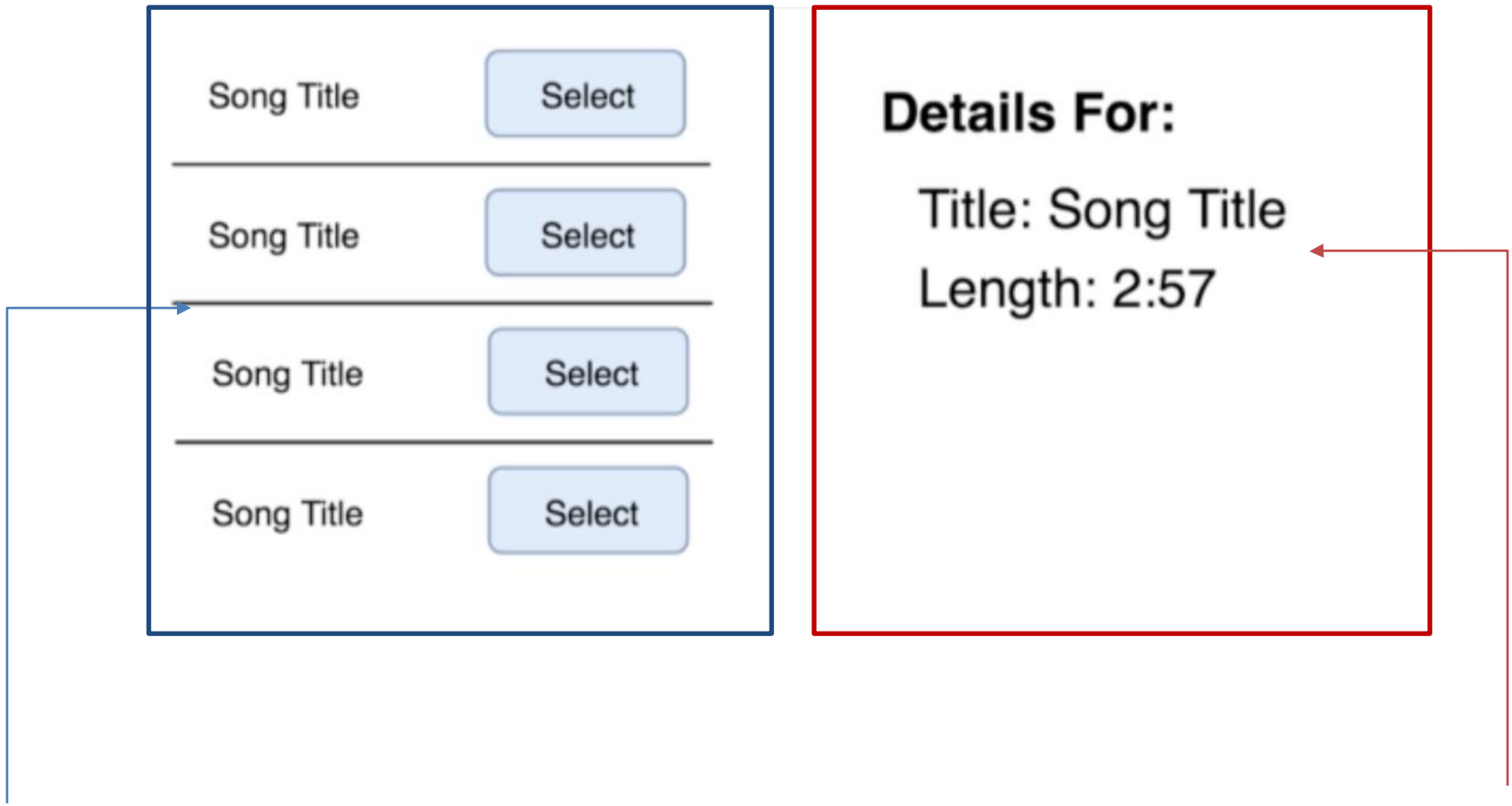
- React-Redux là một thư viện độc lập cho phép chúng ta sử dụng Redux với một ứng dụng React.
- Chúng thường được sử dụng cùng nhau để truy cập dữ liệu global state

❑ Cài đặt

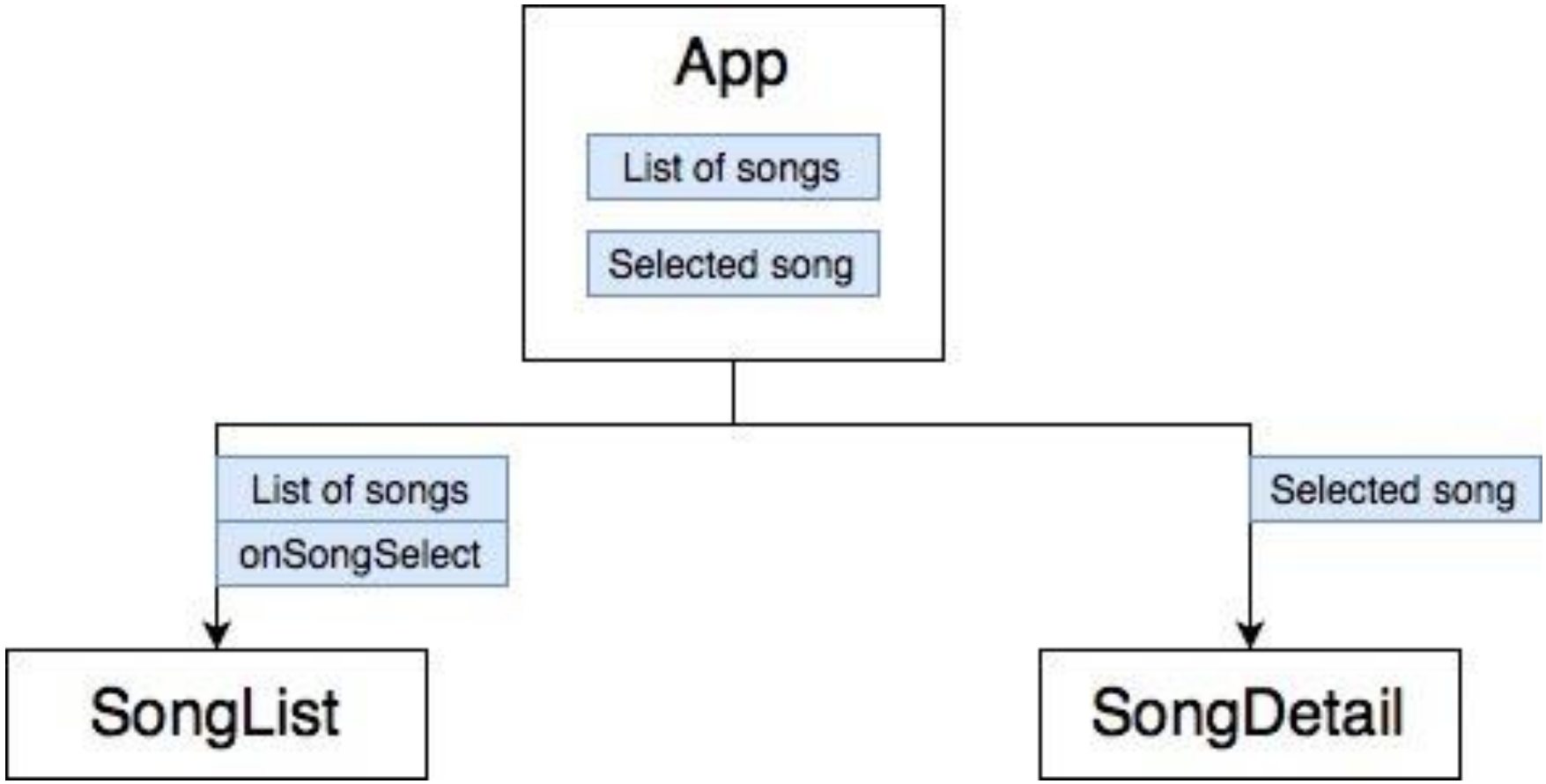
```
➜  songs git:(144-setup) ✗ npm install --save redux react-redux
+ react-redux@5.1.0
+ redux@4.0.1
added 8 packages from 5 contributors and audited 35672 packages in 9
.254s
found 0 vulnerabilities

➜  songs git:(144-setup) ✗ npm start
```

SONG LIST APP

Yêu cầu

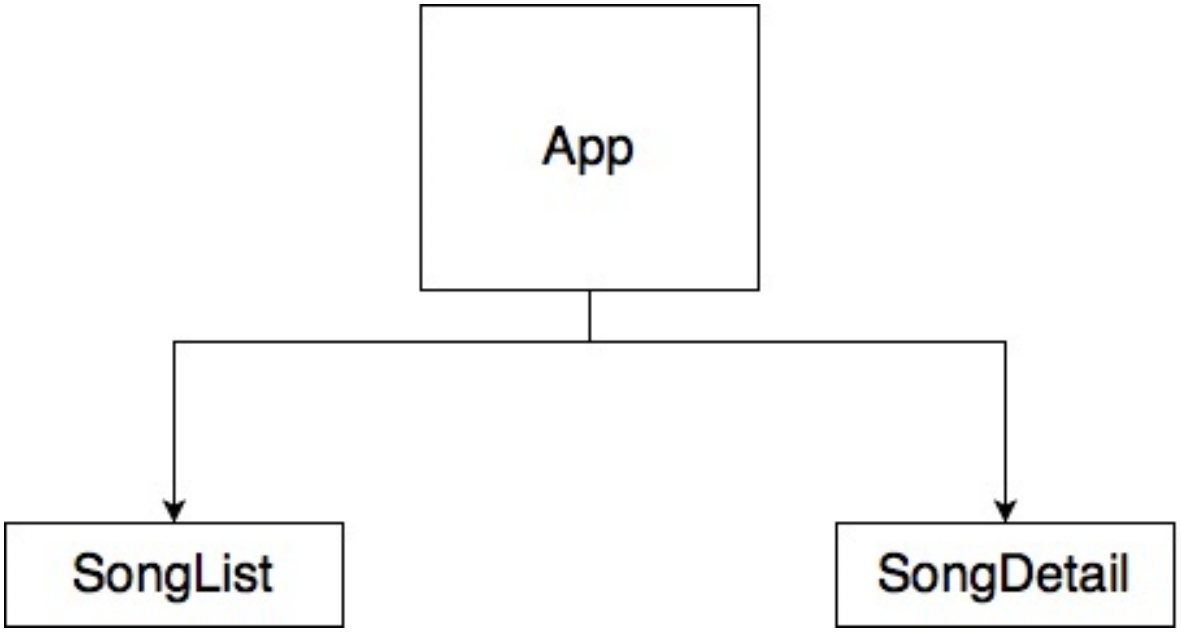

Song Title
Select
Song Title
Select
Song Title
Select
Song Title
Select
Details For:
Title: Song Title
Length: 2:57
SongList
SongDetail

- Không sử dụng redux

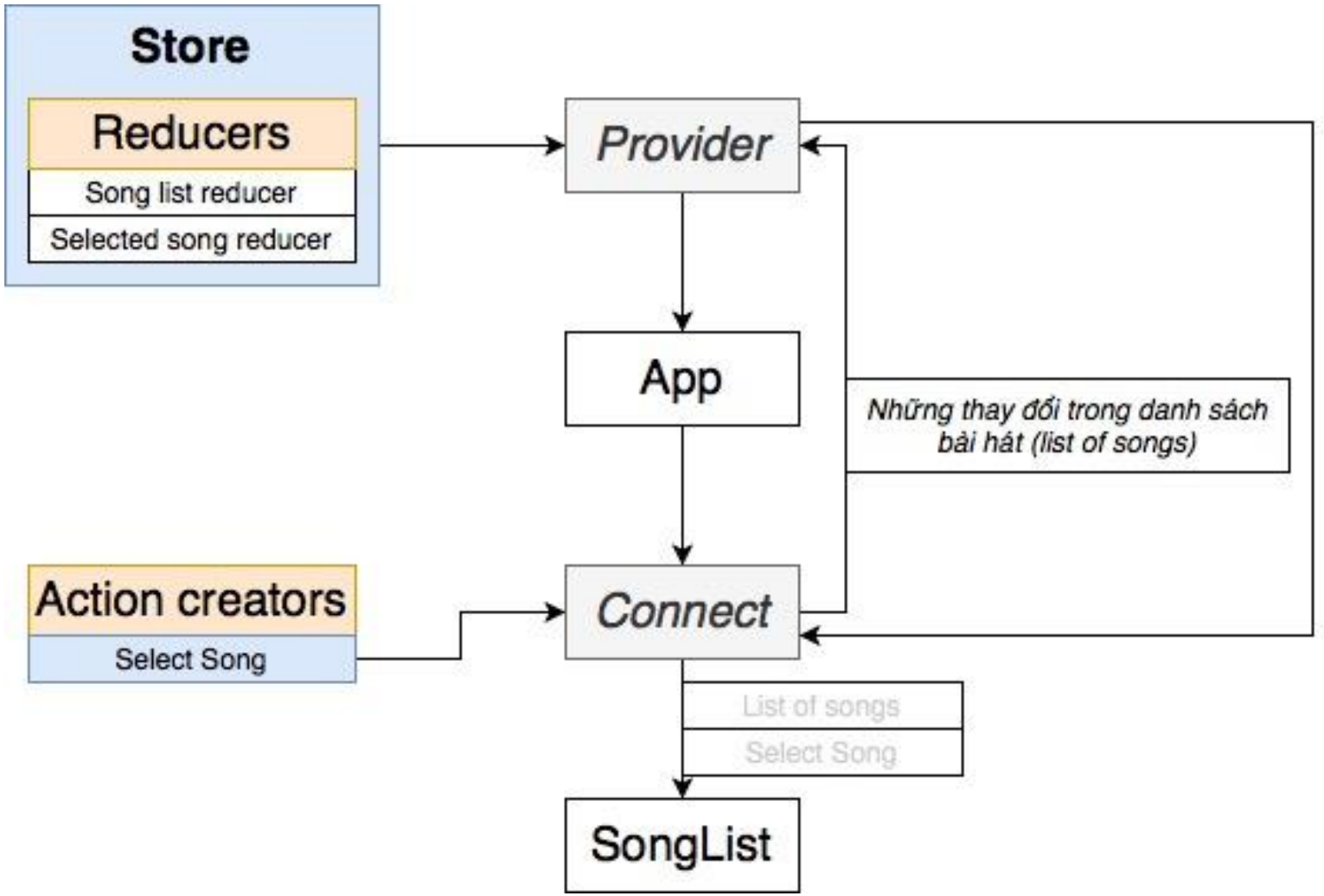


SONG LIST APP

- Sử dụng redux



Sử dụng redux



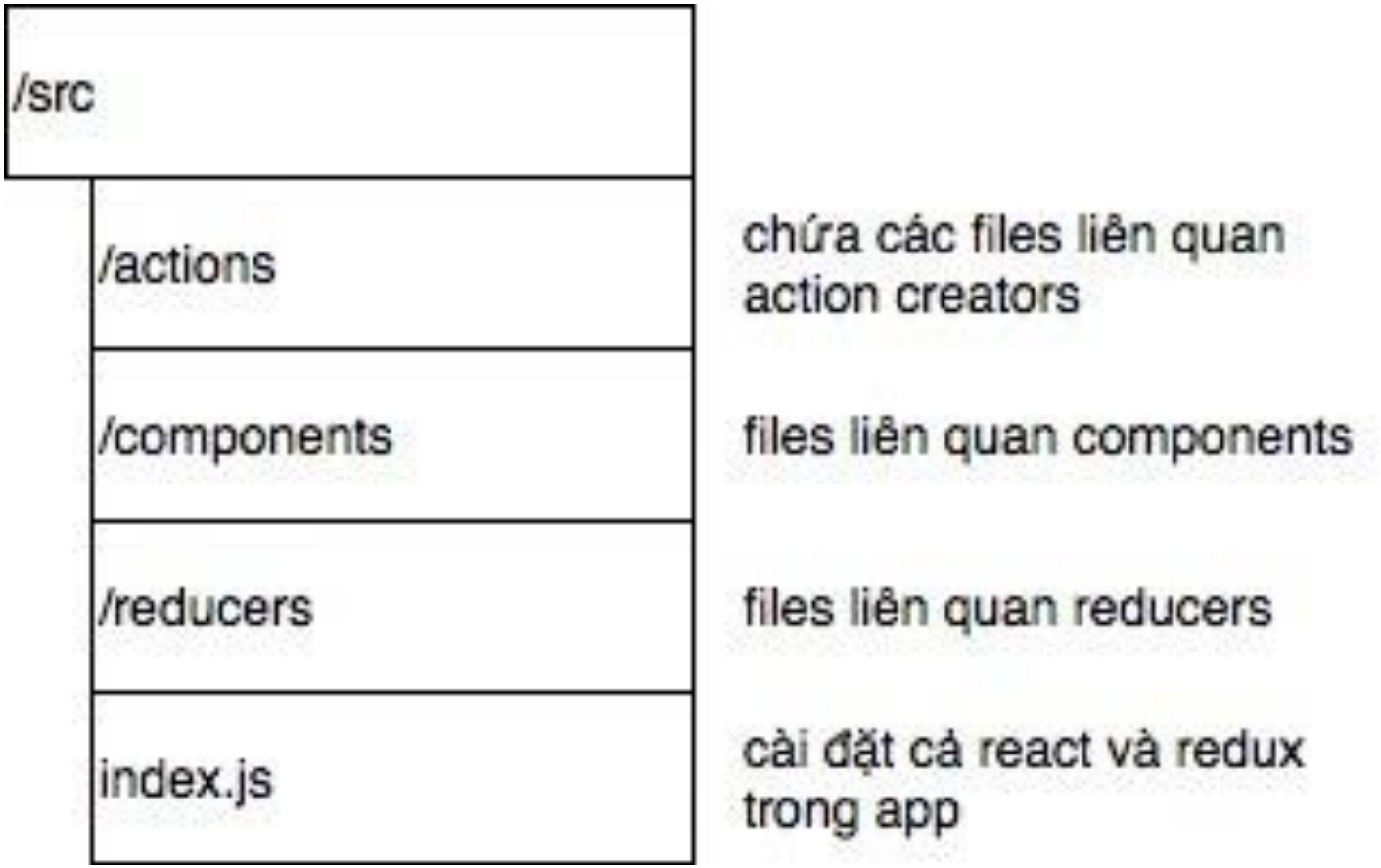
PROVIDER

- Provider
  - Là component của react
  - Được cung cấp bởi thư viện react-redux
  - Cung cấp store cho những component con của nó

CONNECT

- Connect ()
  - Là 1 method
  - Kết nối các components đến store
  - Cho phép truy xuất global state vào component
  - Chỉ có những component bên trong providers mới có thể connect
  - **mapStateToProps** `(state, [ownProps]) =>` stateProps : được sử dụng để lấy giá trị state (vào component hiện tại) như 1 props value.
  - **mapDispatchToProps**: được sử dụng để truy xuất dispatch

Cấu trúc project



SONG LIST APP - ACTION

JS index.js ×

src > actions > JS index.js > ...

```
1   // Action creator
2   export const selectSong = song => {
3     // Return an action
4     return {
5       type: 'SONG_SELECTED',
6       payload: song
7     };
8   };
```

SONG LIST APP - REDUCERS

JS index.js

src > reducers > JS index.js > ...

```
1   import { combineReducers } from 'redux';
2
3   const songsReducer = () => {
4     return [
5       { title: 'No Scrubs', duration: '4:05' },
6       { title: 'Macarena', duration: '2:30' },
7       { title: 'All Star', duration: '3:15' },
8       { title: 'I Want it That Way', duration: '1:45' }
9     ];
10  };
11
12  const selectedSongReducer = (selectedSong = null, action) => {
13    if (action.type === 'SONG_SELECTED') {
14      return action.payload;
15    }
16
17    return selectedSong;
18  };
19
20  export default combineReducers({
21    songs: songsReducer,
22    selectedSong: selectedSongReducer
23  });
```

Provider

JS index.js ×

src > JS index.js

```
1   import React from 'react';
2   import ReactDOM from 'react-dom';
3   import { Provider } from 'react-redux';
4   import { createStore } from 'redux';
5   import App from './components/App';
6   import reducers from './reducers';
7
8   ReactDOM.render(
9     <Provider store={createStore(reducers)}>
10      <App />
11    </Provider>,
12    document.querySelector('#root')
13  );
```

Connect

JS SongList.js ×

src > components > JS SongList.js > ...

```
1   import React, { Component } from 'react';
2   import { connect } from 'react-redux';
3
4   class SongList extends Component {
5     render() {
6       return <div>SongList</div>;
7     }
8   }
9
10  export default connect()(SongList);
11
```

CONNECT - MAPSTATETOPROPS

Connect

JS SongList.js ●

src > components > JS SongList.js > ...

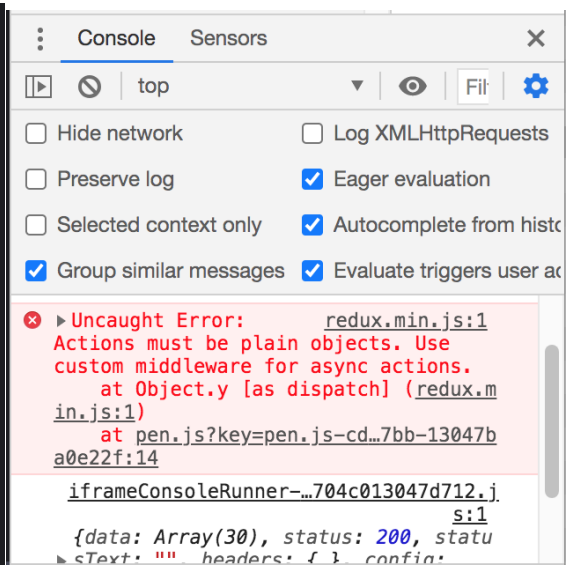
```
1   import React, { Component } from 'react';
2   import { connect } from 'react-redux';
3
4   class SongList extends Component {
5     render() {
6       console.log(this.props);
7       return <div>SongList</div>;
8     }
9   }
10
11  const mapStateToProps = state => {
12    return { songs: state.songs };
13  };
14
15  export default connect(mapStateToProps)(SongList);
16
```

Phần 2
Redux thunk

REDUX THUNK

codepen.io

```
const postsAction = async() => {
  const response = await axios.get('https://api.github.com/users');
  console.log(response);
  return {
    type: 'FETCH_USER',
    payload: response.data
  };
};
const{createStore, applyMiddleware, combineReducers} = Redux;
const thunk = ReduxThunk.default;
const store = createStore(applyMiddleware(thunk));
store.dispatch(postsAction());
```

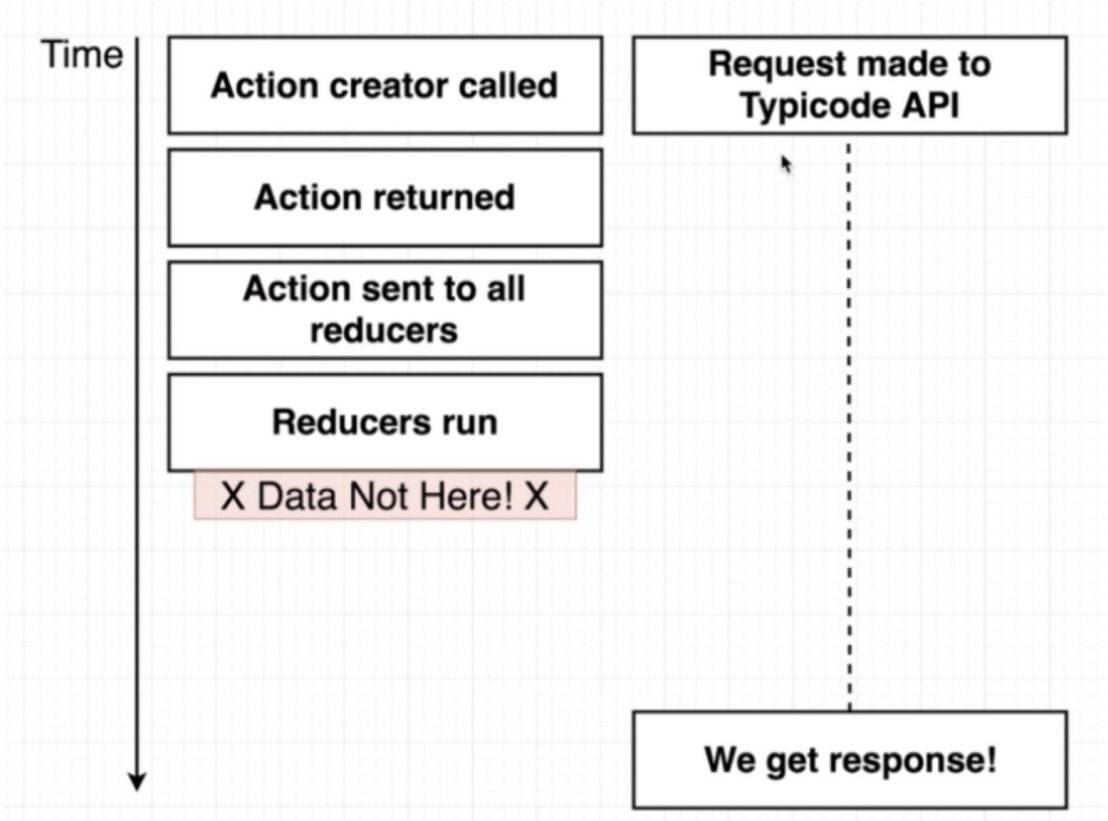


Lỗi gì xảy ra ở postsAction?

action creators phải trả về js thuần tuý.
postsAction thì không.

khi action hết thời gian đến reducer,
postsAction không fetch dữ liệu

REDUX THUNK



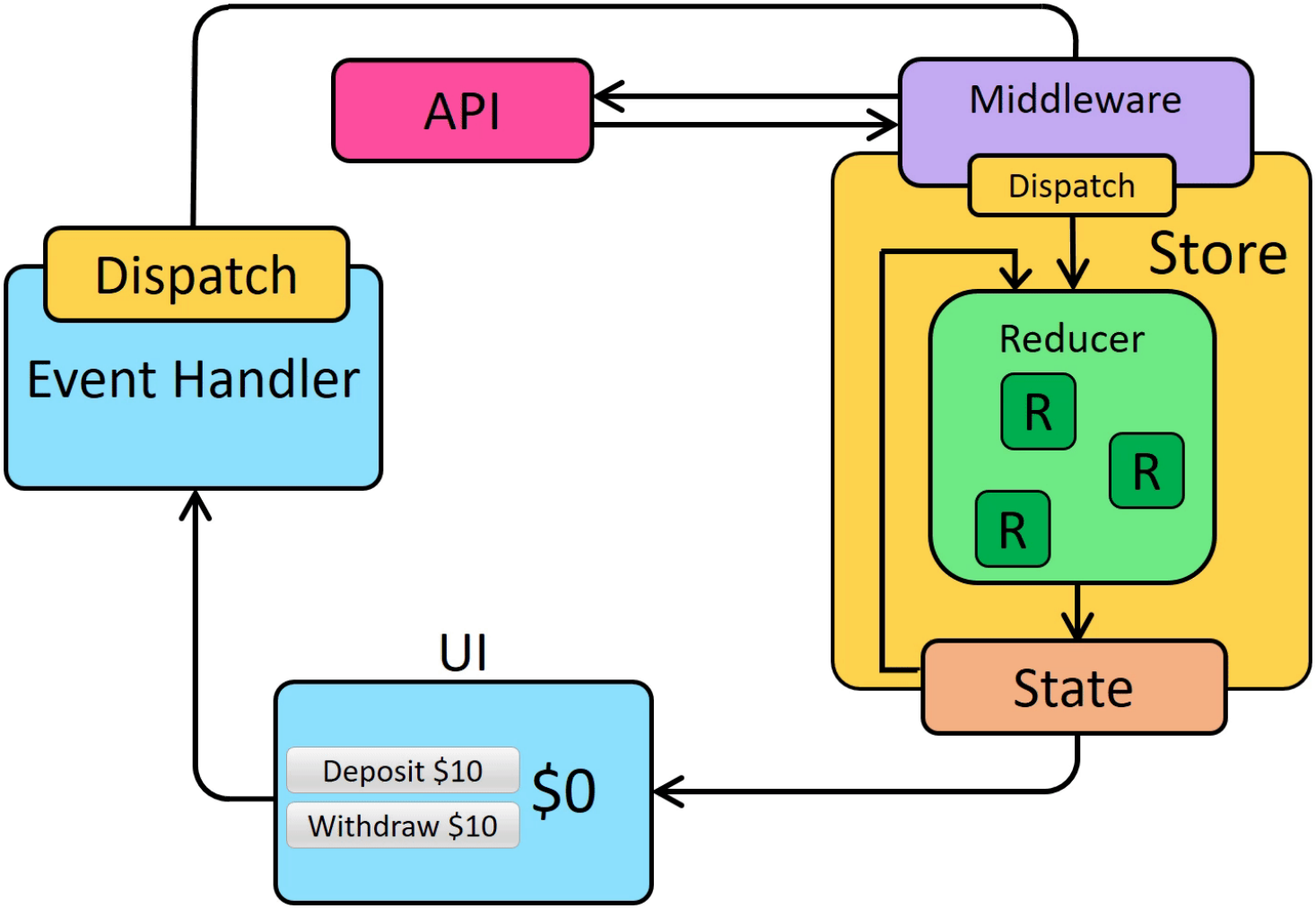
Synchronous action creator

ngay lập tức trả về 1 action với data đã sẵn sàng

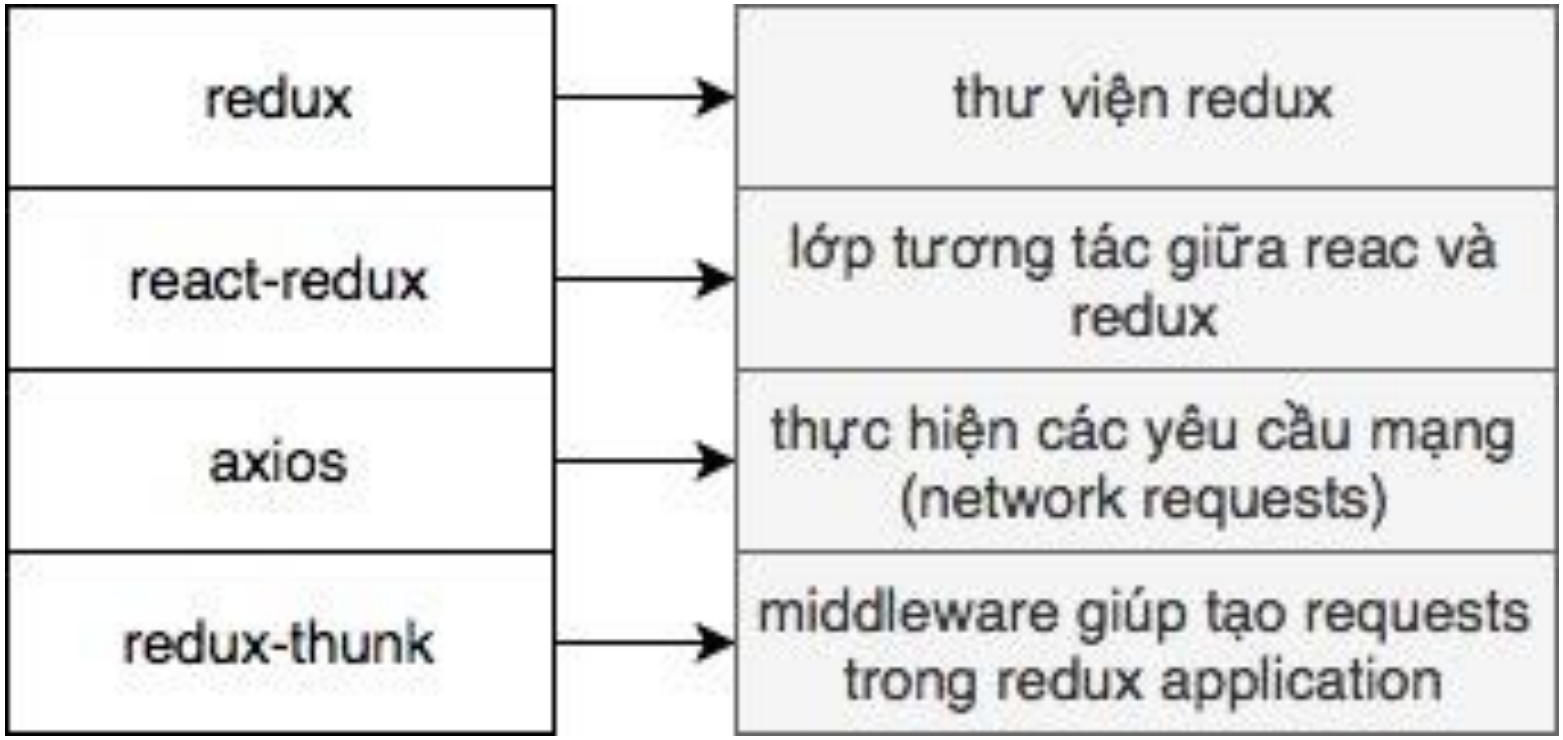
Asynchronous action creator

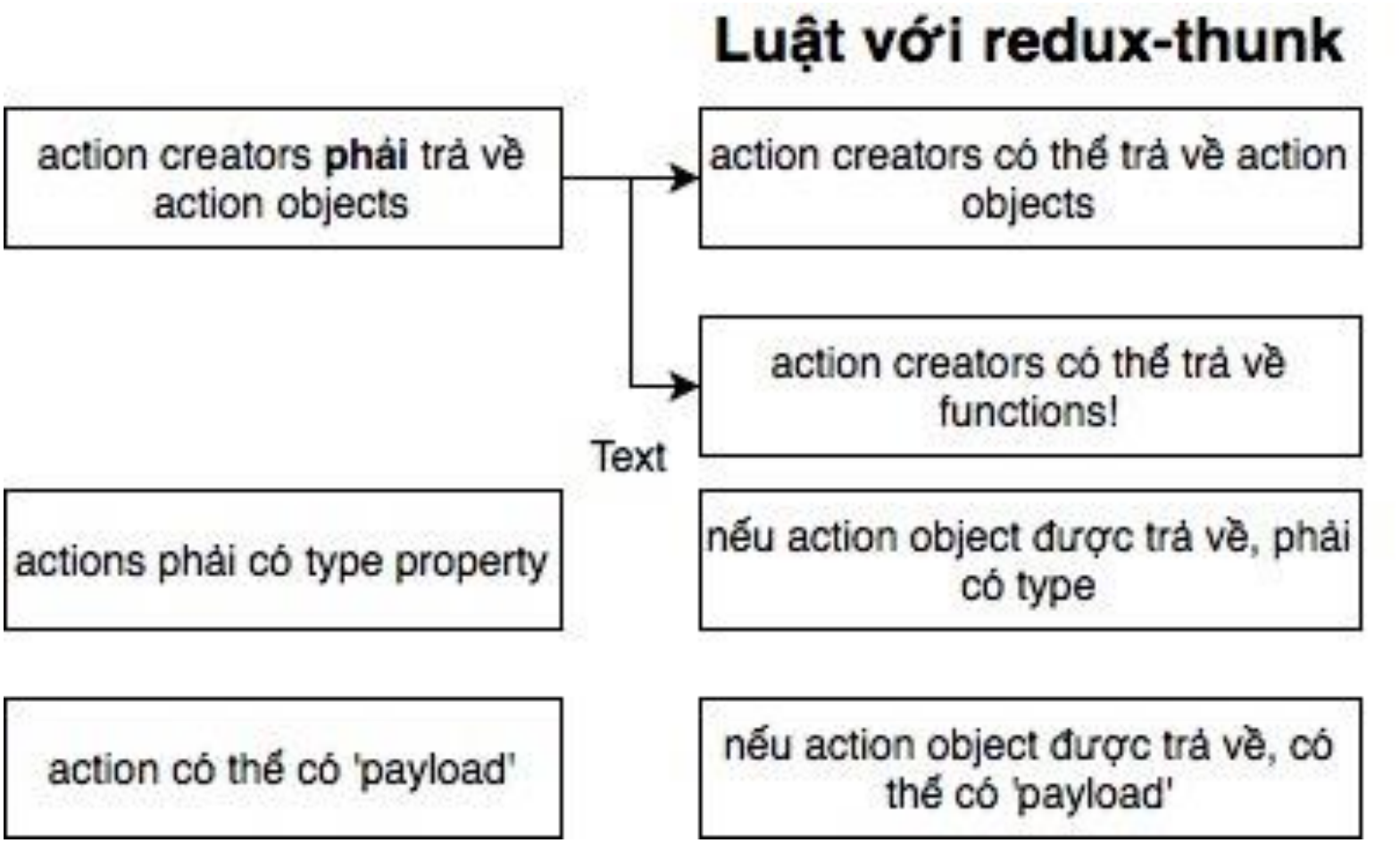
ngay lập tức trả về 1 action với data đã sẵn sàng

REDUX ASYNC DATA FLOW



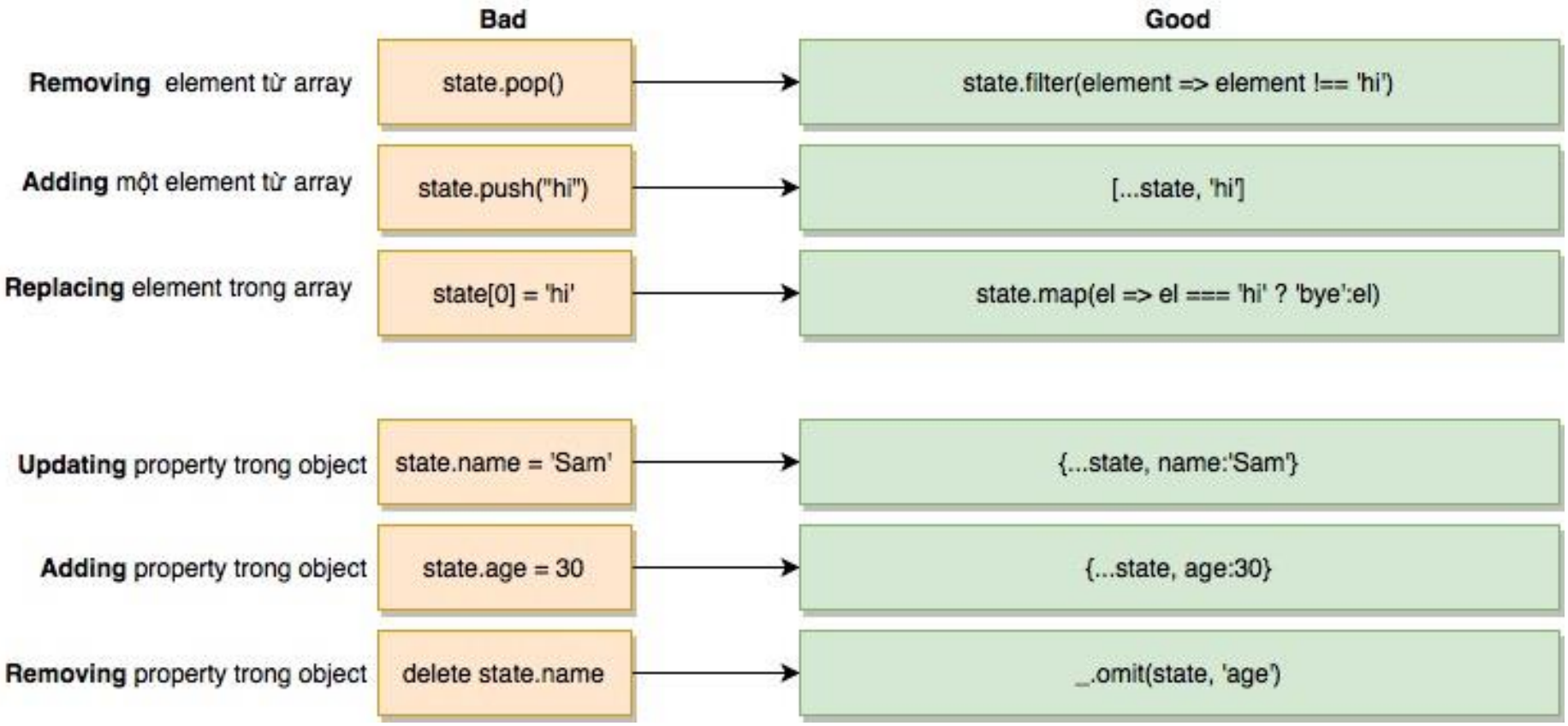
REDUX-THUNK


redux
react-redux
axios
redux-thunk
thư viện redux
lớp tương tác giữa reac và redux
thực hiện các yêu cầu mạng (network requests)
middleware giúp tạo requests trong redux application


Luật với redux-thunk
action creators phải trả về action objects
action creators có thể trả về action objects
action creators có thể trả về functions!
Text
actions phải có type property
nếu action object được trả về, phải có type
action có thể có 'payload'
nếu action object được trả về, có thể có 'payload'

Blog app

CẬP NHẬT STATE




thank
you!