

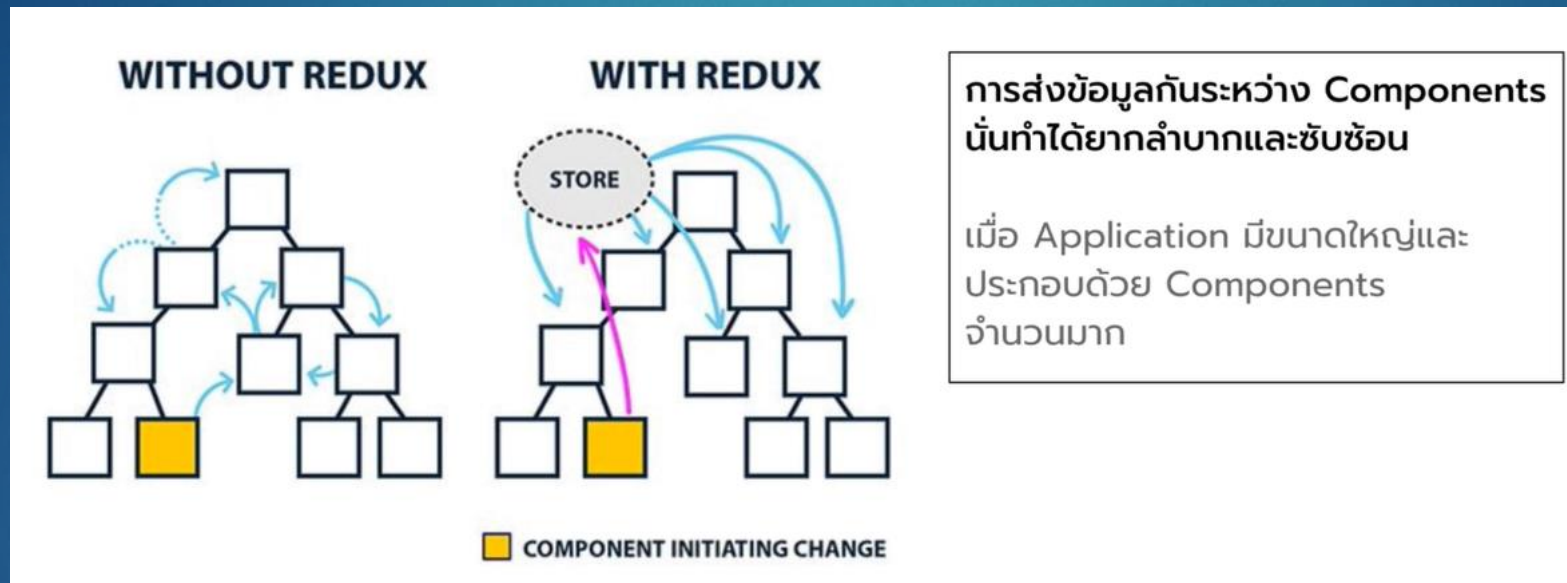


Redux Workshop

REACT + REDUX

Why Redux ?

- ▶ Redux ใช้ในการจัดการ State ที่ใช้สำหรับการ render component
- ▶ ปัญหาที่เกิดขึ้น ว่าทำไม เราถึงใช้ Redux



- ▶ แก้ปัญหาด้วย ด้วยการเก็บ State ไว้ที่ที่ เดียวคือ Store

Structure and flow control Redux!

- ▶ 1. Single source of truth เก็บข้อมูลไว้ที่เดียวคือ store
- ▶ 2. State is read-only การเปลี่ยนแปลงข้อมูลสามารถทำได้ผ่าน actions
- ▶ 3. Changes are made with pure functions สามารถเปลี่ยนแปลง แก้ไข ข้อมูลผ่านทาง pure functions (Reducer) โดยเมื่อได้รับ actions เข้ามา

STORE ACTION REDUCER

Store

```
import { createStore, applyMiddleware } from "redux";
import { composeWithDevTools } from "redux-devtools-extension";
import thunk from "redux-thunk";
import rootReducer from "../redux/reducers";

const initialState = {};

const middleware = [thunk];

const store = createStore(
  rootReducer,
  initialState,
  composeWithDevTools(applyMiddleware(...middleware))
);

export default store;
```

```
▼ cat (pin)
  ▼ cats (pin)
    ▼ 0 (pin)
      id (pin): "1"
      name (pin): "Persian"
      image (pin): "https://images.unsplash.com/photo-1494256997604-768d1f608cac?ixlib=rb-1.2.1&ixid=eyJhCHBfawQiojEYMDd9&auto=format&fit=crop&w=400&q=60"
      create_date (pin): "2019-10-16T03:00:00.000Z"
    ► 1 (pin): {}
    ▼ 2 (pin)
      id (pin): "3"
      name (pin): "Exotic Shorthair"
      image (pin): "https://images.unsplash.com/photo-1472491235688-bdc81a63246e?ixlib=rb-1.2.1&ixid=eyJhCHBfawQiojEYMDd9&auto=format&fit=crop&w=400&q=60"
      create_date (pin): "2019-10-18T03:00:00.000Z"
    ► 3 (pin): {}
    error (pin): null
```

Actions

- ▶ Trigger to change state with action
- ▶ How to send action with “dispatch”

```
dispatch({  
  type: ADD_CAT,  
  payload: response.data,  
});
```

```
dispatch({  
  type: FETCH_CATS,  
  payload: response.data,  
});
```


Reducer

- ▶ This pure function
- ▶ Accept action and process to change data state then return into store

```
dispatch({  
  type: FETCH_CATS,  
  payload: response.data,  
});
```

```
export default (state = initialState, action) => {  
  const { type, payload } = action;  
  
  switch (type) {  
    case FETCH_CATS:  
      return {  
        ...state,  
        cats: payload,  
      };  
    case ADD_CAT:  
      state.cats.push(payload);  
      return {  
        ...state,  
        cats: state.cats,  
      };  
    case ERROR_CAT:  
      return {  
        ...state,  
        error: payload,  
      };  
    default:  
      return state;  
  }  
};
```

Combine Store & Reducer

```
import { createStore, applyMiddleware } from "redux";
import { composeWithDevTools } from "redux-devtools-extension";
import thunk from "redux-thunk";
import rootReducer from "../reducer/reducers";

const initialState = {};

const middleware = [thunk];

const store = createStore(
  rootReducer,
  initialState,
  composeWithDevTools(applyMiddleware(...middleware))
);

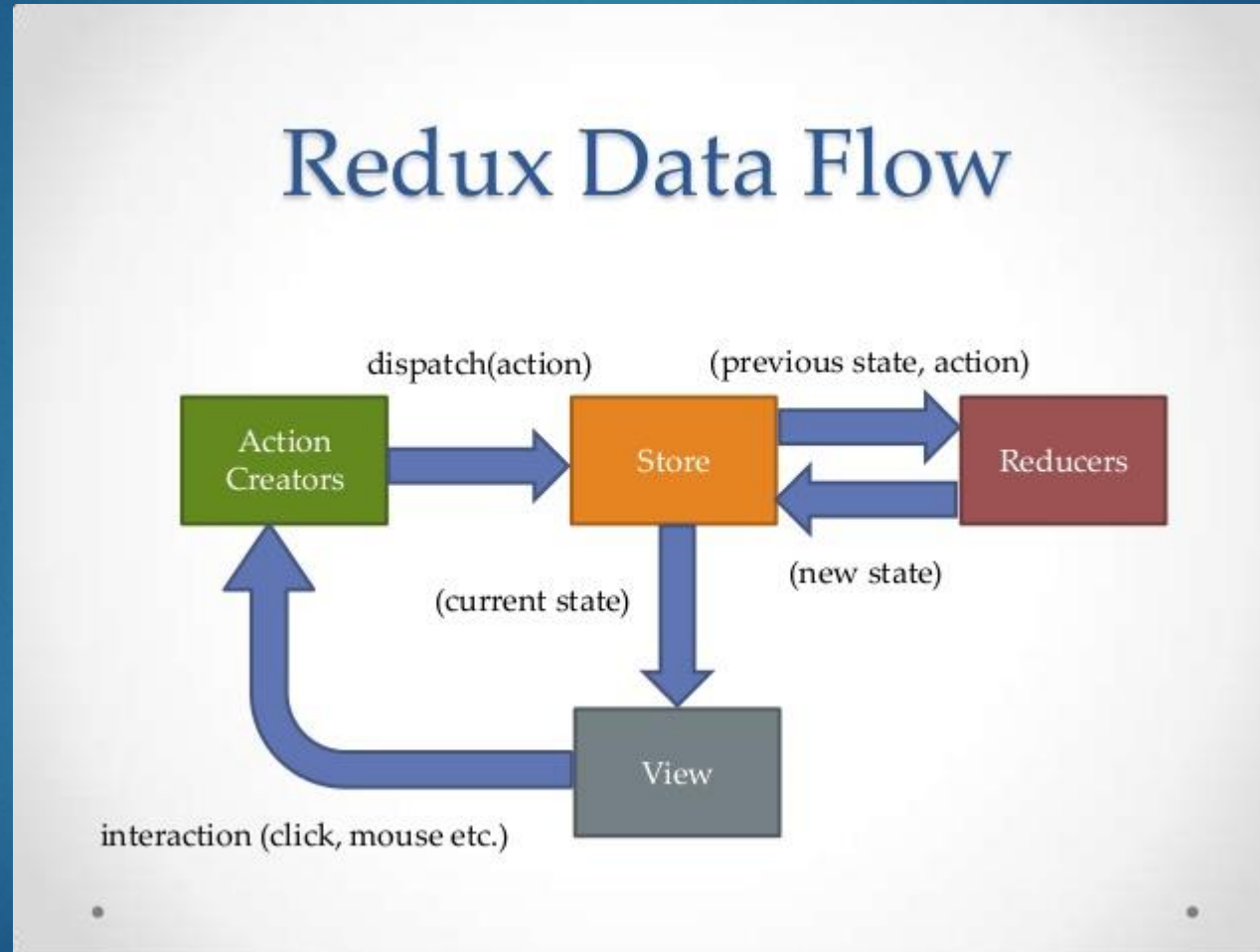
export default store;
```

Directory ../reducer/index.js

```
import { combineReducers } from "redux";
import cat from "../cat";

export default combineReducers({ cat });
```

Redux Data Flow



Library Tools

Relative Redux!!

- ▶ Yarn add react-redux redux redux-devtools-extension redux-thunk
- ▶ OR npm install react-redux redux redux-devtools-extension redux-thunk --save
- ▶ React-redux
- ▶ Redux
- ▶ Redux-devtools-extension
- ▶ Redux-thunk

Workshop

- ▶ How to query data from api
- ▶ How to on event data storage in mock database

Tools

- ▶ Json-server (npm install -g json-server)
- ▶ Axios - Promise based HTTP client for the browser
- ▶ Ref: <https://www.npmjs.com/package/json-server>
- ▶ <https://github.com/chattavat-v/redux>

Test

- ▶ Add event to click delete cat card
- ▶ User redux and json-server to DELETE action
- ▶ Trick
- ▶ Filter method
- ▶ Use id to remove in reducer



Persian

2019-10-16

DELETE