React Native Redux

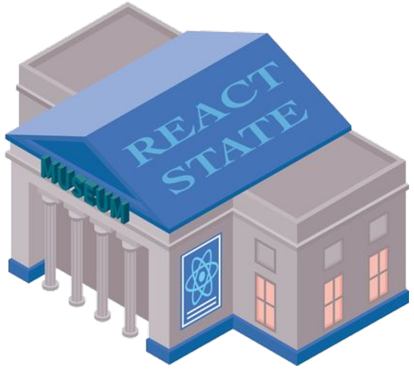
Isumi
Batam, Jan 2020

BINAR ACADEMY

GOALS

1. Intro about State Management
2. Concept Redux
3. Implementation Redux

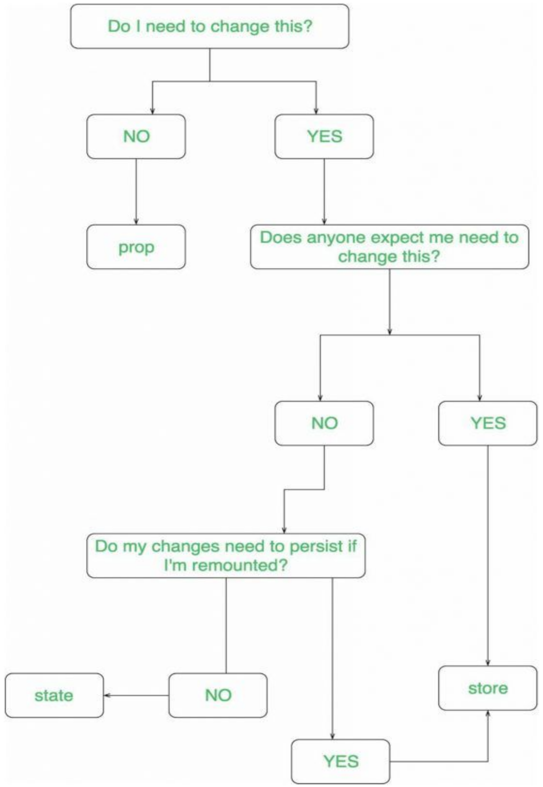
1. Intro about State Management



Notes:
The more you care about state management, the more complex your application will become. Keep it simple and change when you really have the need for more control over your state.

For more details:

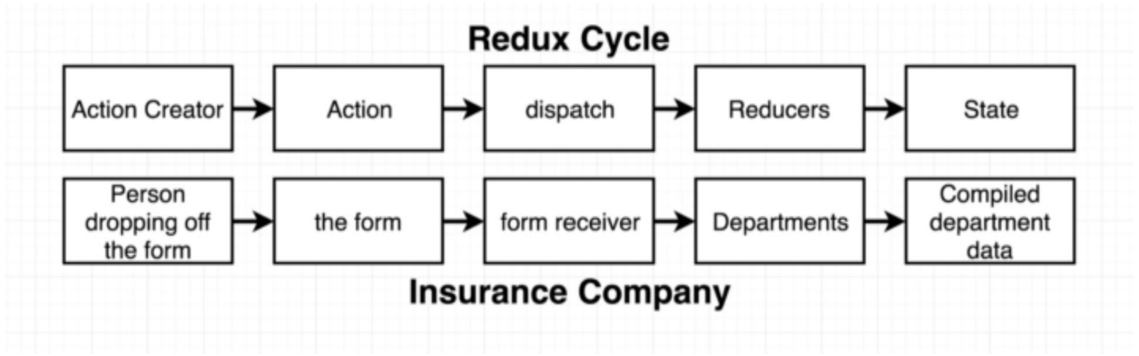
- https://www.geeksforgeeks.org/component-state-react-native/
- https://medium.com/dailyjs/comparison-of-state-management-solutions-for-react-2161a0b4af7b

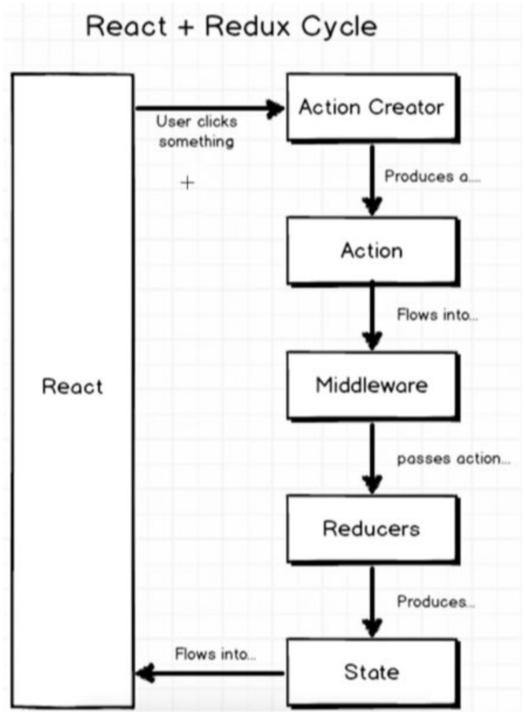

Do I need to change this?
NO
YES
prop
Does anyone expect me need to change this?
NO
YES
Do my changes need to persist if I'm remounted?
state
NO
YES
store

2. Concept

REDUX

1. State Management
2. Inspired by flux and elm
3. Can be used without react





For more details:
https://onoumenon.gitbook.io/wiki/programming/react/untitled-1
https://code-cartoons.com/a-cartoon-intro-to-redux-3afb775501a6

Benefits?

One general state in the store and each component has access to the state. This eliminates the need to continuously pass state from one component to another.

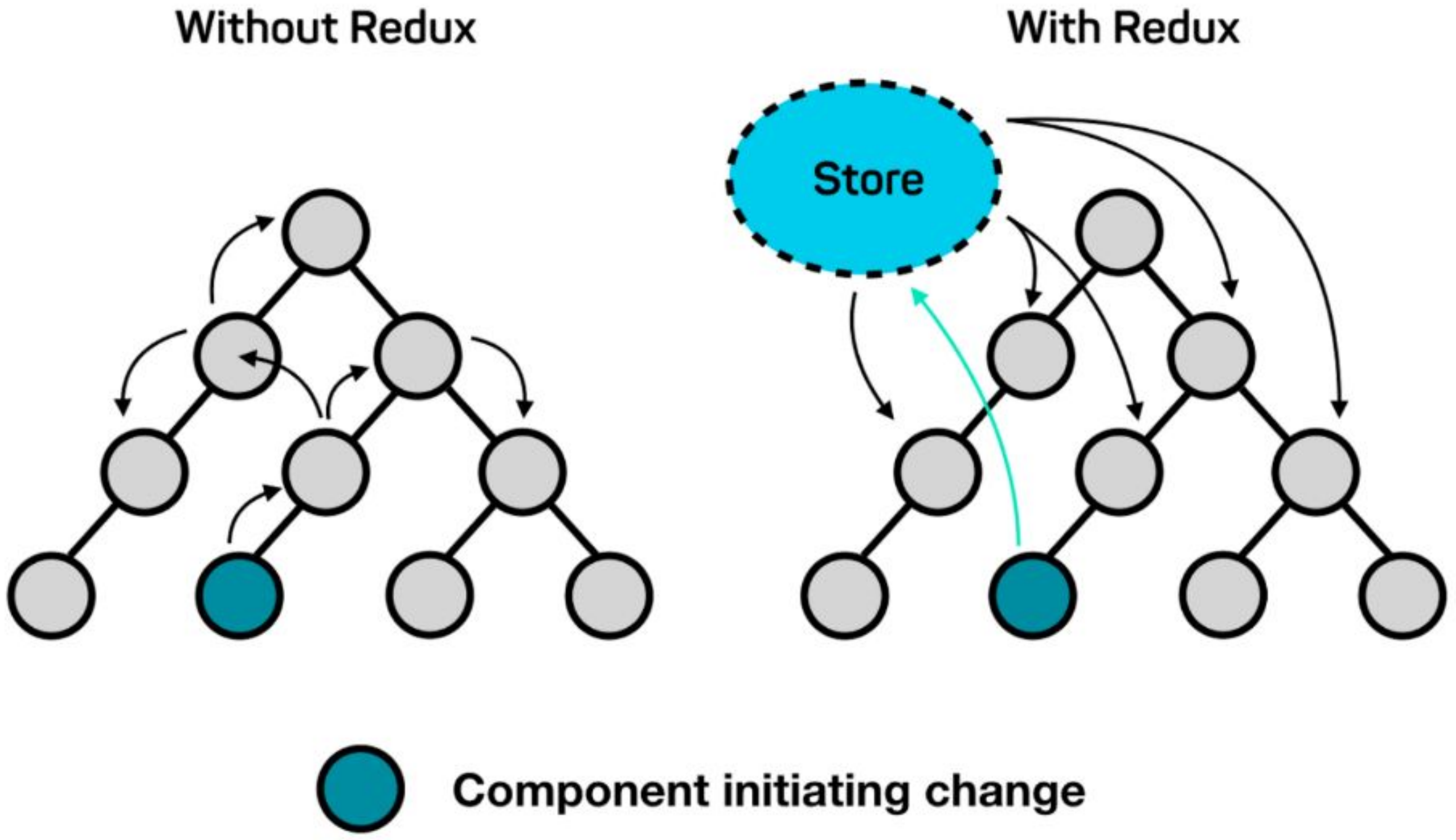
What else?

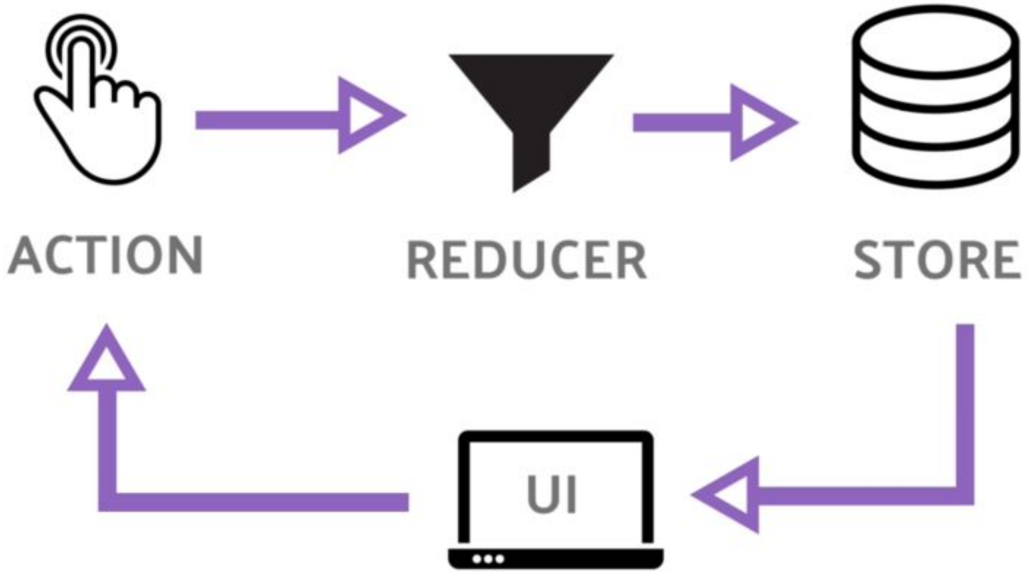
1. Redux makes the state predictable.

   In Redux, the state is always predictable. If the same state and action are passed to a reducer, the same result is always produced as reducers are pure functions. The state is also immutable and is never changed

2. Maintainability

   Redux is strict about how code should be organized so it makes it easier for someone with knowledge of Redux to understand the structure of any Redux application.

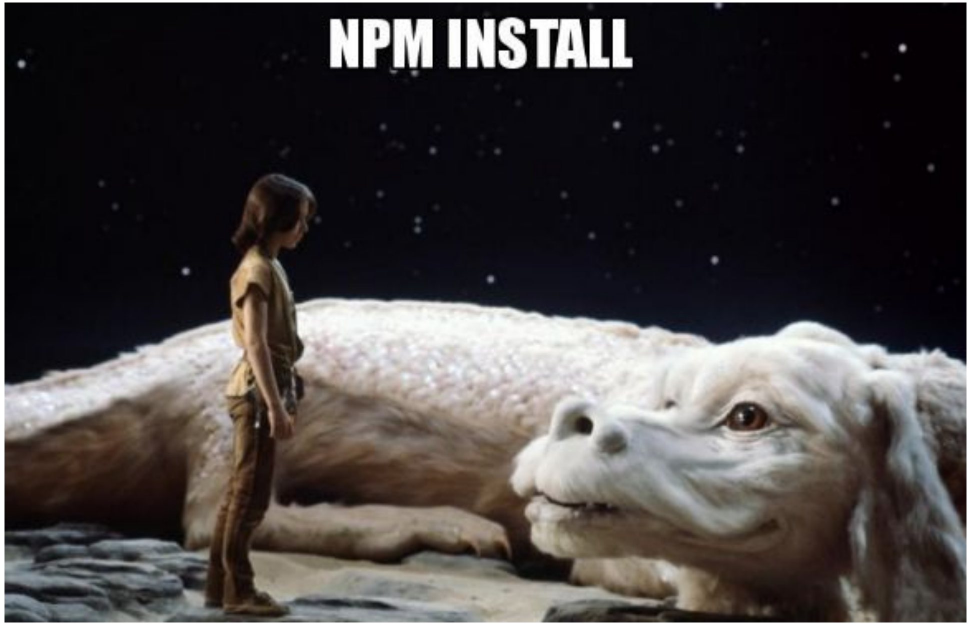

Without Redux
With Redux
Store
● Component initiating change


ACTION
REDUCER
STORE
UI

3. Implementation Redux

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

- redux
- react-redux
- redux-saga (optional => advanced)



For more details:
https://unbug.gitbooks.io/react-native-training/content/41_redux+react.html

./store.js

```
import { createStore } from 'redux';
import rootReducer from './src/redux/reducer/index';

const configureStore = () => {
 return {
   ...createStore(rootReducer)
 };
};

export default configureStore;
```

./src/redux/reducer/index.js

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

export default IndexReducer;
```

./src/redux/reducer/AuthReducer.js

```
import { FETCH_TOKEN } from '../type/AuthType';

const initialState = {
 token: null
};

export default (state = initialState, action) => {
 switch (action.type) {
   case FETCH_TOKEN:
     return {
       ...state,
       token: action.payload
     };
   default:
     return state;
 }
};
```

./src/redux/action/AuthAction.js

```
import {FETCH_TOKEN} from '../type/AuthType';

export const auth = data => {
 console.log(data)
 return {
   type: FETCH_TOKEN,
   payload: data
 };
};
```

./src/redux/type/AuthType.js

```
export const FETCH_TOKEN = 'FETCH_TOKEN';
```

./App.js

```
import React, { Component } from 'react';
import { Provider } from 'react-redux';
import configureStore from './store';
const store = configureStore();

export default class App extends Component {
 render() {
   return (
     <Provider store={store}>
       <AppContainer />
     </Provider>
   );
 }
}
```

./src/components/Login.js

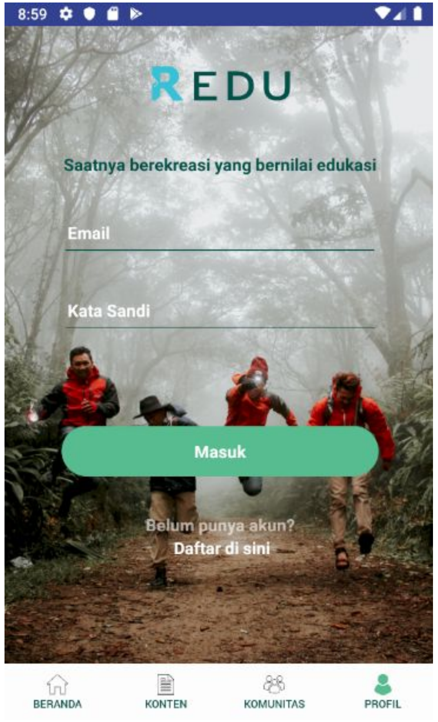
```
import { connect } from 'react-redux';
import { auth } from '../redux/action/AuthAction';

class Login extends Component {
// passing props auth at lifecycle component
     componentDidMount()  {
        console.log('auth redux', this.props.auth);
     }
}

const mapStateToProps = state => ({
 auth: state.auth
});

// redux store in here
const mapDispatchToProps = dispatch => {
 return {
   FetchToken: token => dispatch(auth(token))
 };
};

export default connect( mapStateToProps, mapDispatchToProps )(Login);
```

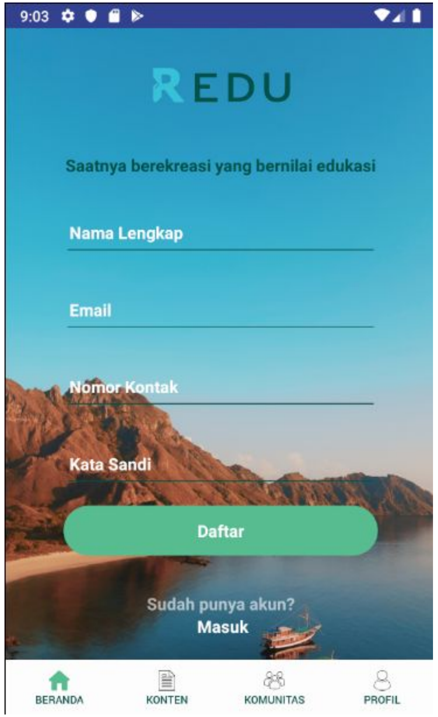
```
Don't forget:
// passing props FetchToken at handleLogin()  func
this.props.FetchToken(apiLogin.data.token);
```

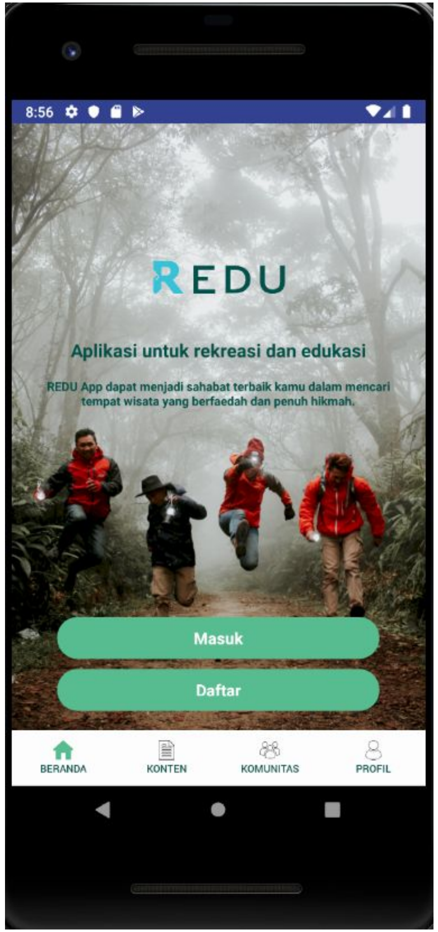
REACT, REDUX, REACT ROUTER. WHATS NEXT
REACT-ROUTER-REDUX
imgflip.com

Exercise

Refactor your Login Todo App using Redux ;)






8:56
REDU
Aplikasi untuk rekreasi dan edukasi
REDU App dapat menjadi sahabat terbaik kamu dalam mencari tempat wisata yang berfaedah dan penuh hikmah.
Masuk
Daftar
BERANDA
KONTEN
KOMUNITAS
PROFIL