



# React Redux 아키텍처

플랫폼사업부문 개발 4unit  
모바일개발 1Cell

노원상 연구원

# React & Redux

A red circle with a white number 1 in the center, surrounded by a light red ring.

1. React & Redux 개념

A red circle with a white number 2 in the center, surrounded by a light red ring.

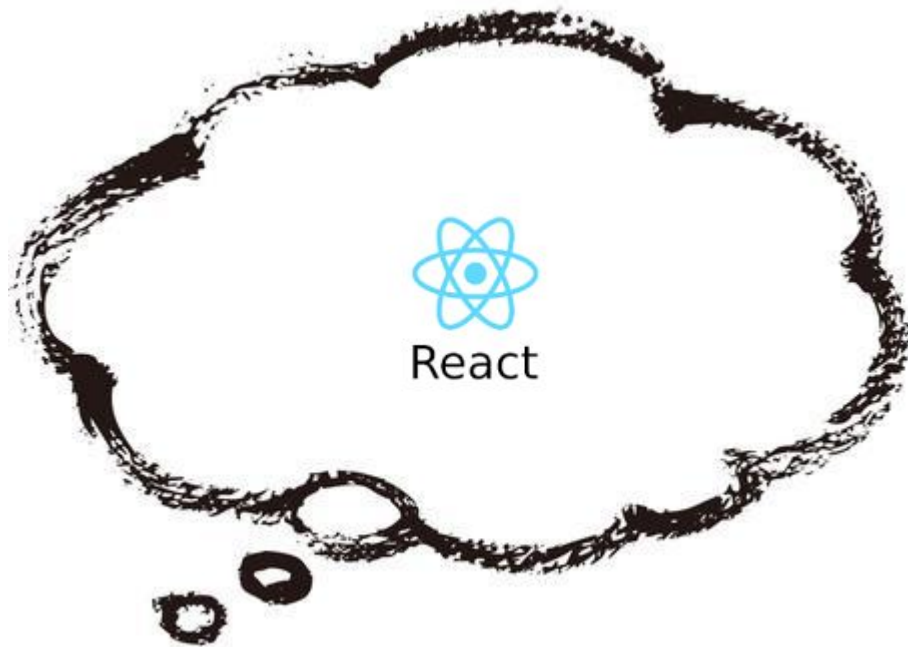
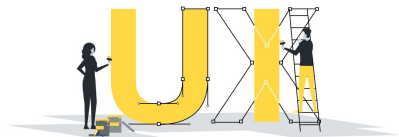
2. Redux 실전코드



# **PART 01**

## **React & Redux**

# React



우리가 쓰고 있는 React는  
무엇일까?

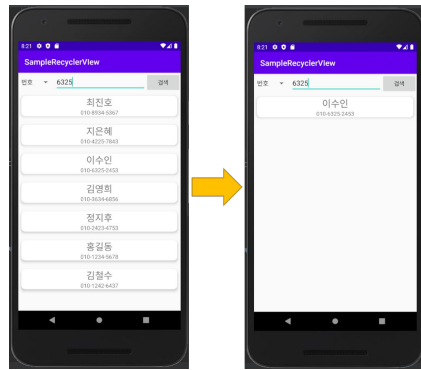
# React



모바일인 경우



React를 사용하지 않으면?



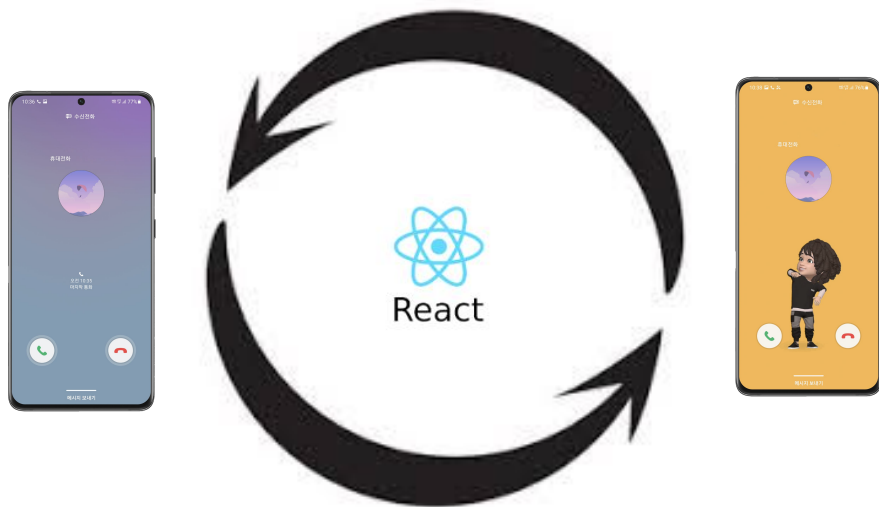
**notifyDataSetChanged()**



**setImageResource()**



# React



UI를 자동 업데이트 해주는 React



개발자는 대체적으로 편안함을 느낀다



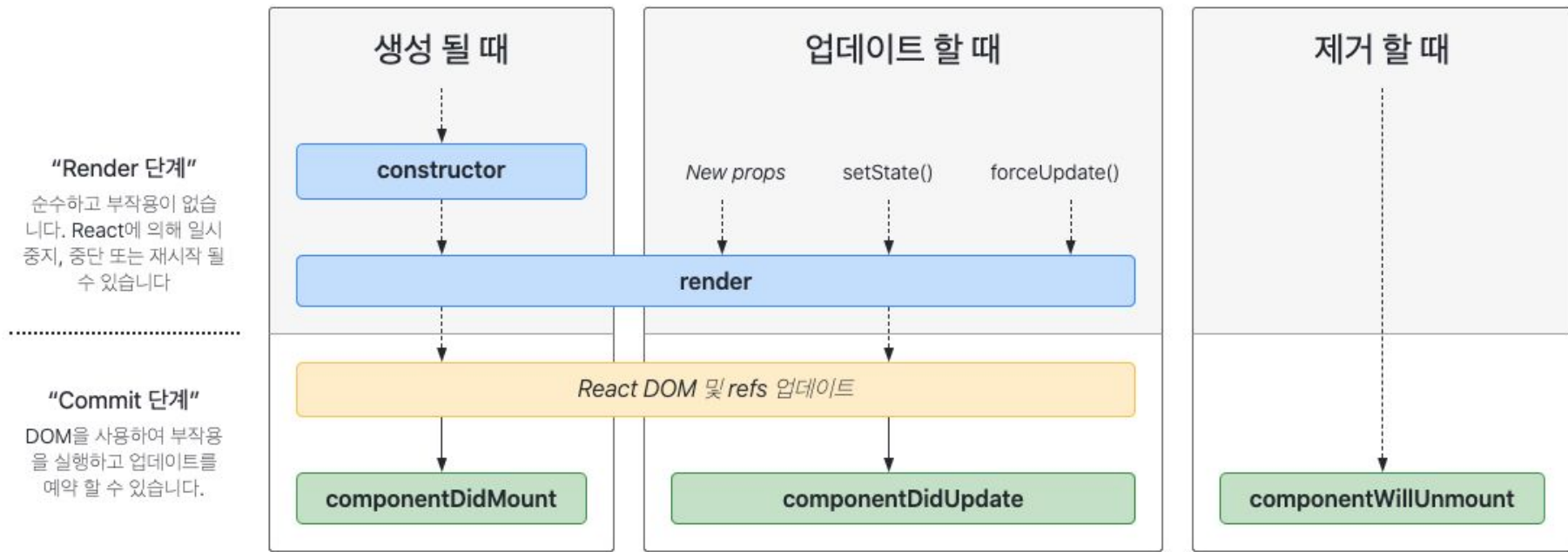
# React

---

실습 맛보기

---

# LifeCycle



저희가 쓰는 React lifeCycle 에 대해 알고 계신가요?



# Redux



# Redux

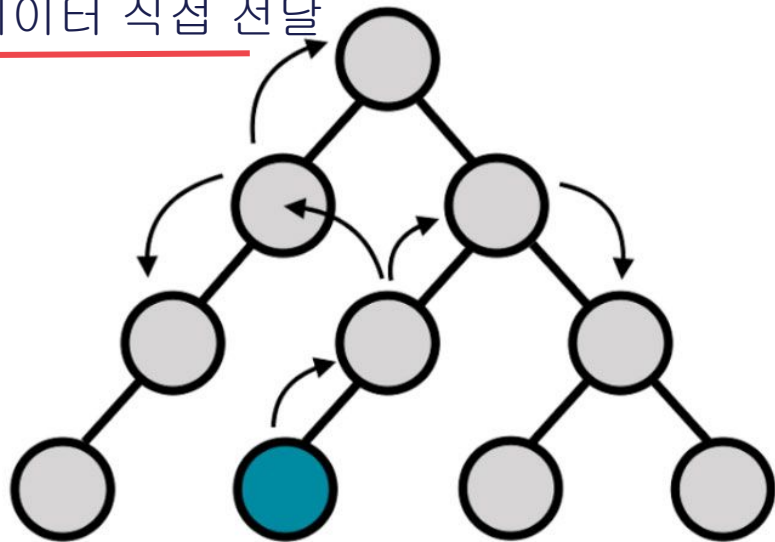
1.관리 라이브러리

2.스토어

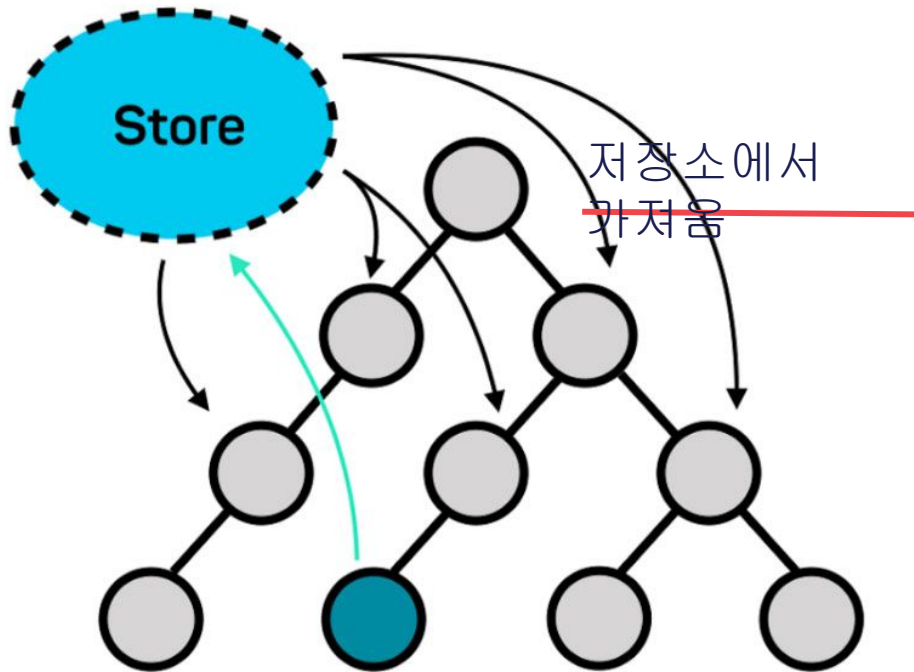
3.컴포넌트  
처리방식

## props

데이터 직접 전달



## Redux



**상태가 변경된 컴포넌트**

# Redux를 사용한 실전코드



JS (Babel)

```
1 //2021.05.11 Redux 실습내용(store 생성)
2
3 import { createStore ,combineReducers} from 'redux'
4 //Redux 패키지에서 createStore,combineReducers 모듈을 가져온다
5
6 const store = createStore(
7   combineReducers({ rootReducer }),
8   //createStore 스토어 생성 모듈
9   //combineReducer 리덕스 리듀서를 작성할때 가장 일반적인 용례를 단순화 하는 유틸리티 함수
10 );
11
```

# Redux를 사용한 실전코드



JS (Babel)

```
1 //2021.05.11 Redux 실습내용(store 최상위 뷰에 연결)
2
3 import store from "../store"; //내가 생성한 스토어
4 import { Provider } from "react-redux"; //react-Redux 패키지에서 provider를 가져옴
5
6 //최상위 뷰 App.tsx
7 //provider를 사용해서 다른 컴포넌트가 store에 접근하여 사용할수 있게 해줌
8 render(){
9   <View>
10     <Provider store={store}>
11       ... 중략
12     </Provider>
13   </View>
14 }
```

# Redux를 사용한 실전코드

```
import { connect } from "react-redux"; //컴포넌트의 로직을 재사용 할 때 사용하는 패턴
```

A view ...종락

```
const mapStateToProps = (state: any) => { // mapStateToProps store에 있는 state 데이터를 가져옴
  return {
    A data: state.store.A data
  };
};
export default connect(mapStateToProps)(A View); // 특정 값을 props 로 받아와서 사용함
```

B View ...종락

```
const mapStateToProps = (state: any) => { // mapStateToProps store에 있는 state 데이터를 가져옴
  return {
    B data: state.store.B data
  };
};
export default connect(mapStateToProps)(A View); // 특정 값을 props 로 받아와서 사용함
```

앱의 모든 상태를  
오직 하나의 Store에 저장해~

어떻게 변경시킬까?^^

변경된 상태값을 전달하자!

Store

Selector(s)

props

Reducer(s)

middleware

비동기로 처리하는 일이나  
다른 도울일이 없을까?

스토어를 변경시키자

Action(s)

화면이야~

View

Map state to props

받은 상태값을 어떤

Props에 주입해서 사용할까?

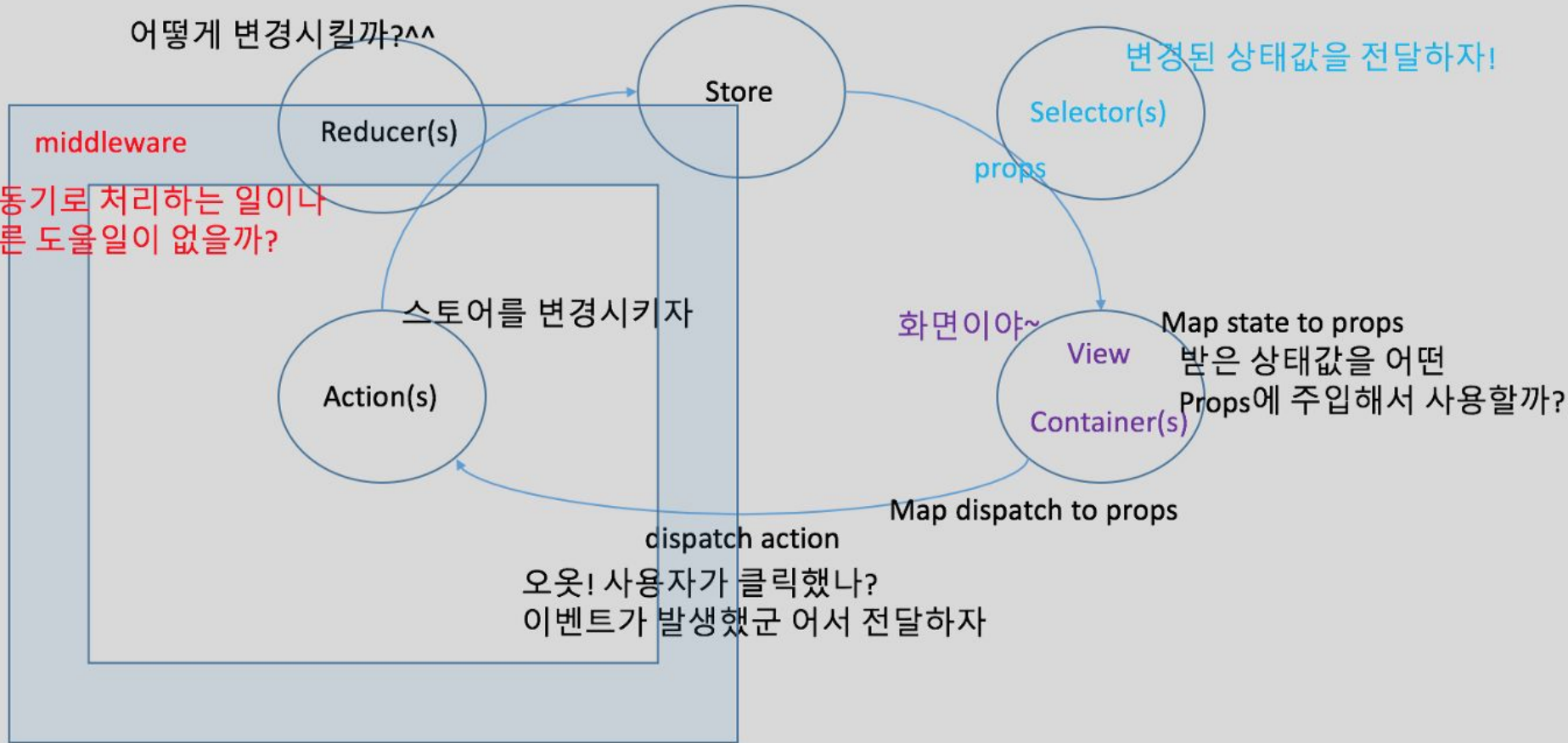
Container(s)

Map dispatch to props

dispatch action

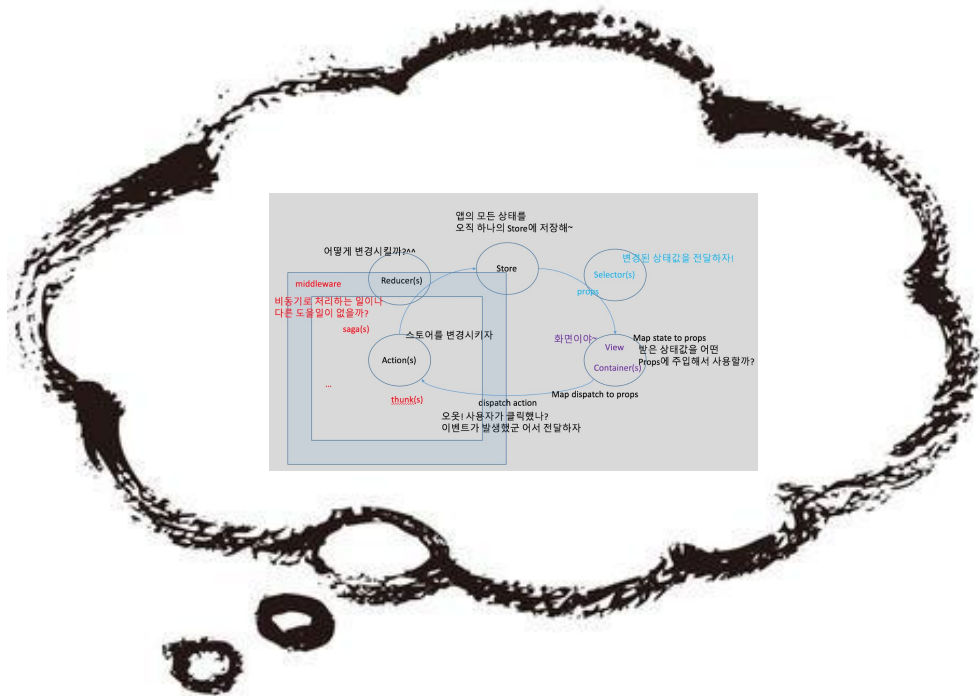
오옷! 사용자가 클릭했나?

이벤트가 발생했군 어서 전달하자





# Redux



이벤트 하나 호출하는데 이러한 관리가?

# React 실전코드

A -> B -> C

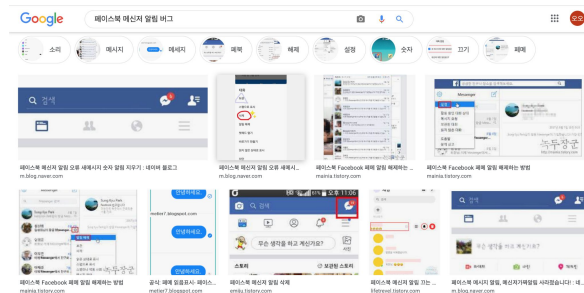
데이터 직접 전달

```
//A View
render(){
  <B View data={"A Data"}/>
}

//B View
interface{
  data:any
}
render(){
  //A data 처리
  <C View data={"B Data"}/>
}

//C View
interface{
  data:any
}
render(){
  //B 데이터 처리
  <D View data={"C Data"}
}
```

# React 작업으로 생길수 있는 문제



# React 작업으로 생길수 있는 문제



이상적인 알림 숫자 변경 시나리오

# React 작업으로 생길수 있는 문제



작업이 엇키면서 예상과 다른 숫자가 보입니다



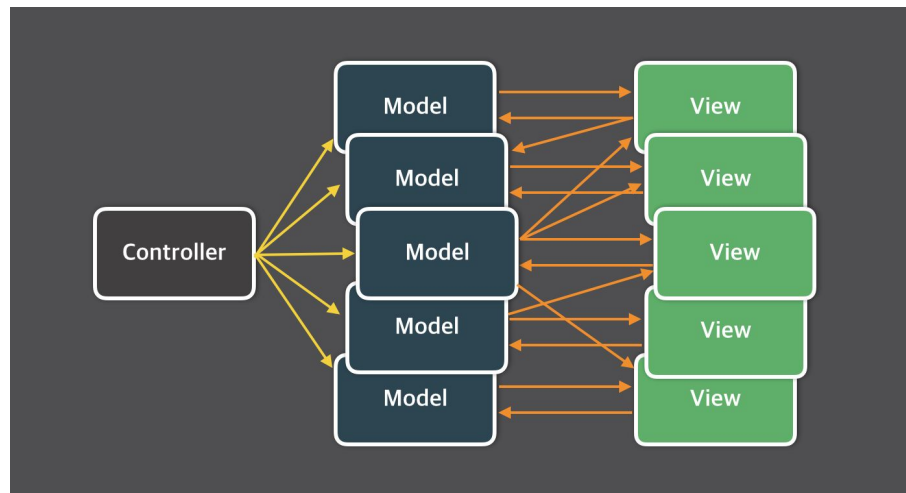
# React 작업으로 생길수 있는 문제

---

- 여러 곳에서 영향을 받는 **State**
- 언제 끝날지 모르는 비동기 작업
- 모델과 **View**가 따로 **State**를 관리함으로써 생기는 격차



# React 실전코드



어디서 문제가 발생했는지 모르겠어...

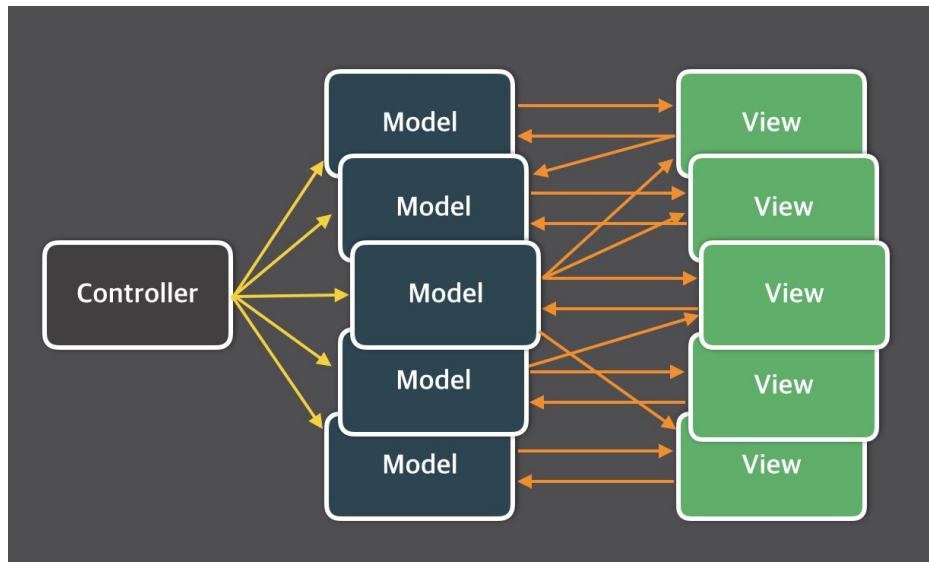


# Redux

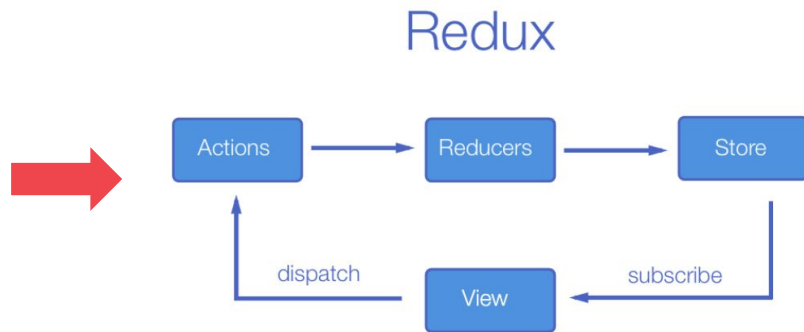
---

- View에 영향을 주는 State는 한 방향으로만 수정할 수 있다.
- 앞 액션이 끝난 후 뒤 액션을 실행한다.
- Model은 State를 변화시키고 View는 State를 참조만 한다.

# React 실전코드



어디서 데이터를 전달??



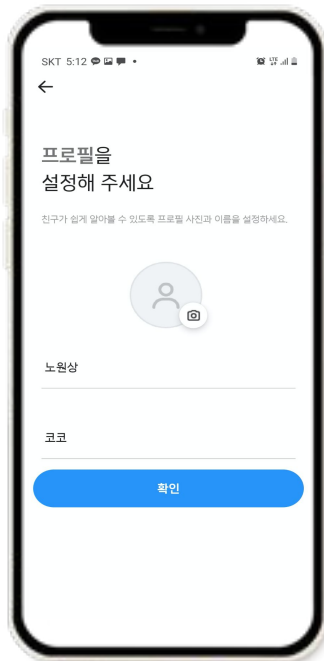
특정 데이터만 가져올수 있어  
편하다



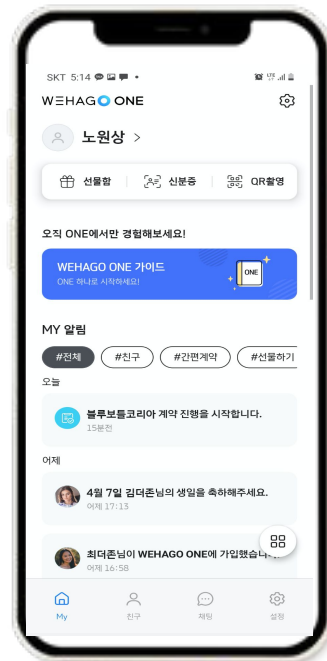
# **PART 02**

## **Redux 실전연습**

# Redux 실습



wehagoOne Redux





# Redux

---

감사합니다!!