

# React Practice - 3

송민석 @ 2019-09-09

# 무엇 하나?

- 이번에는...
- 로그인/회원가입
  - 폼 (Form), 모달(Modal, Dialog)
  - Validation (yup)
  - Firebase API 연동
  - redux, redux-react
  - redux-saga

# 폼

- FormControl
- Input
- InputLabel
- FormHelperText

```
<FormControl className={classes.margin} fullWidth>
  <InputLabel htmlFor='input-username'>Email</InputLabel>
  <Input id='input-username' name='email' type='email' />
  <FormHelperText id='input-username-helper-text'>이메일</FormHelperText>
</FormControl>
```

Email

이메일

# 폼

- Controlled component vs uncontrolled component
  - input의 value값을 react의 state로 관리 (single source of truth)  
`value, onChange`
- 장점
  - 실시간 validation 및 transformation 가능
  - Single Source of Truth
- 단점
  - 복잡한 코드
  - 모든 input에 적용할 수 있는 것은 아님
- 좀 더 복잡한 폼을 사용하려면...
  - Formik (<https://jaredpalmer.com/formik/>)

# Validation

- 직접 구현
  - 간단한 validation은 별도의 라이브러리 없이 구현

`validateInput()`  
`onChange, onBlur`

- Library
  - 복잡도가 증가하면 라이브러리 도움
    - 정확한 이해 필요.
    - 필요하다면 라이브러리 직접 수정 가능해야 함.
  - Yup (<https://github.com/jquense/yup>)
    - 스키마를 이용한 간단한 validation 처리

`npm install yup`

```
let yup = require('yup');

let schema = yup.object().shape({
  name: yup.string().required(),
  age: yup
    .number()
    .required()
    .positive()
    .integer(),
  email: yup.string().email(),
  website: yup.string().url(),
  createdAt: yup.date().default(function() {
    return new Date();
  }),
});

// check validity
schema
  .isValid({
    name: 'jimmy',
    age: 24,
  })
  .then(function(valid) {
    valid; // => true
  });
```

# Firebase \*

- Firebase 사용
  - auth와 db의 백엔드 로직을 위해

```
$ npm install --save firebase
```

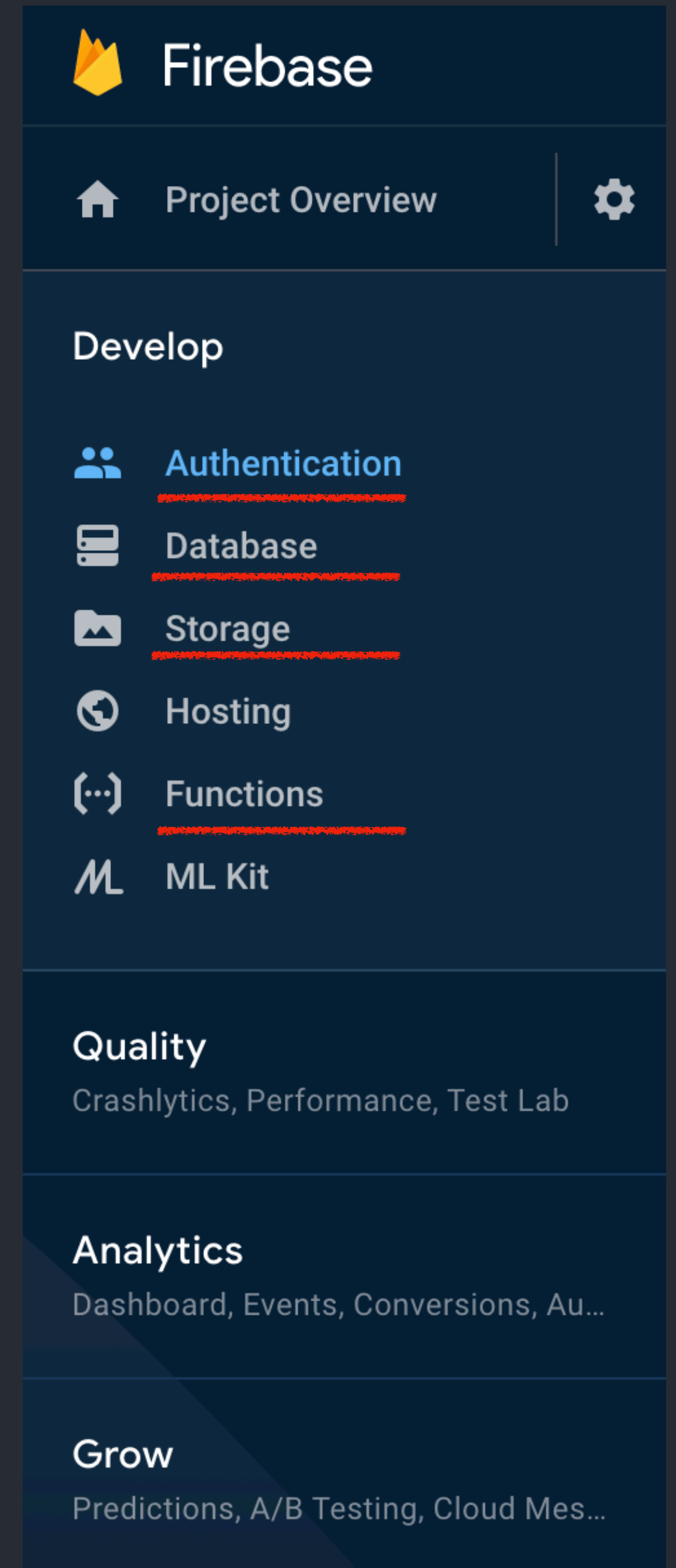
```
// Firebase App (the core Firebase SDK) is always required and must be listed first
import * as firebase from "firebase/app";
```

```
// Add the Firebase products that you want to use
import "firebase/auth";
import "firebase/database";
```

```
// TODO: Replace the following with your app's Firebase project configuration
const firebaseConfig = {
  // ...
};
```

```
// Initialize Firebase
firebase.initializeApp(firebaseConfig);
```

```
const firebaseConfig = {
  apiKey: "AIzaSyDE9-USX40Y8AfH5dGeDhX8N1Lt-28p6Ds",
  authDomain: "yakmoa-80569.firebaseio.com",
  databaseURL: "https://yakmoa-80569.firebaseio.com",
  projectId: "yakmoa-80569",
  storageBucket: "yakmoa-80569.appspot.com",
  messagingSenderId: "692085812753",
  appId: "1:692085812753:web:65aae7b8840d046e"
};
```



# 로그인 / 회원가입 연동

- 로그인

```
firebase.auth().signInWithEmailAndPassword(email, password).catch(function(error) {  
  // Handle Errors here.  
  var errorCode = error.code;  
  var errorMessage = error.message;  
  // ...  
});
```

- 회원가입

```
firebase.auth().createUserWithEmailAndPassword(email, password).catch(function(error) {  
  // Handle Errors here.  
  var errorCode = error.code;  
  var errorMessage = error.message;  
  // ...  
});
```

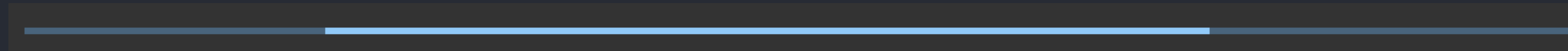
- 로그인 여부 체크

```
firebase.auth().onAuthStateChanged(function(user) {  
  if (user) {  
    // User is signed in.  
    var displayName = user.displayName;  
    var email = user.email;  
    var emailVerified = user.emailVerified;
```

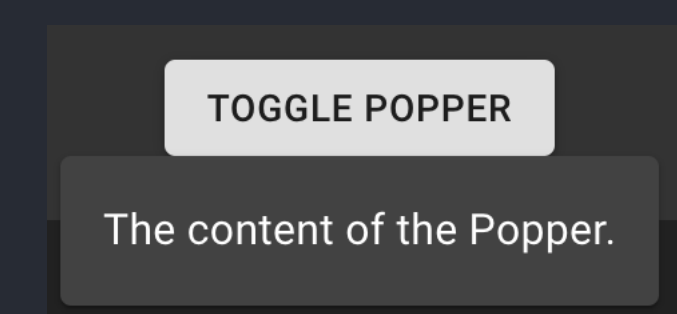
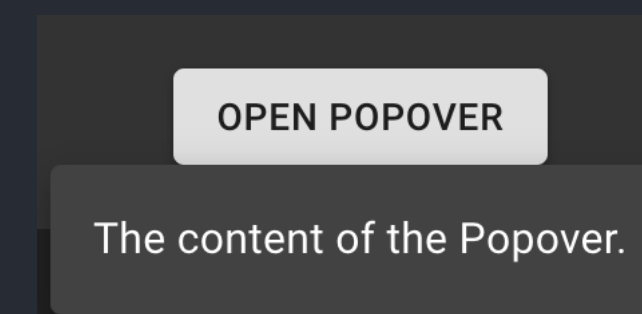
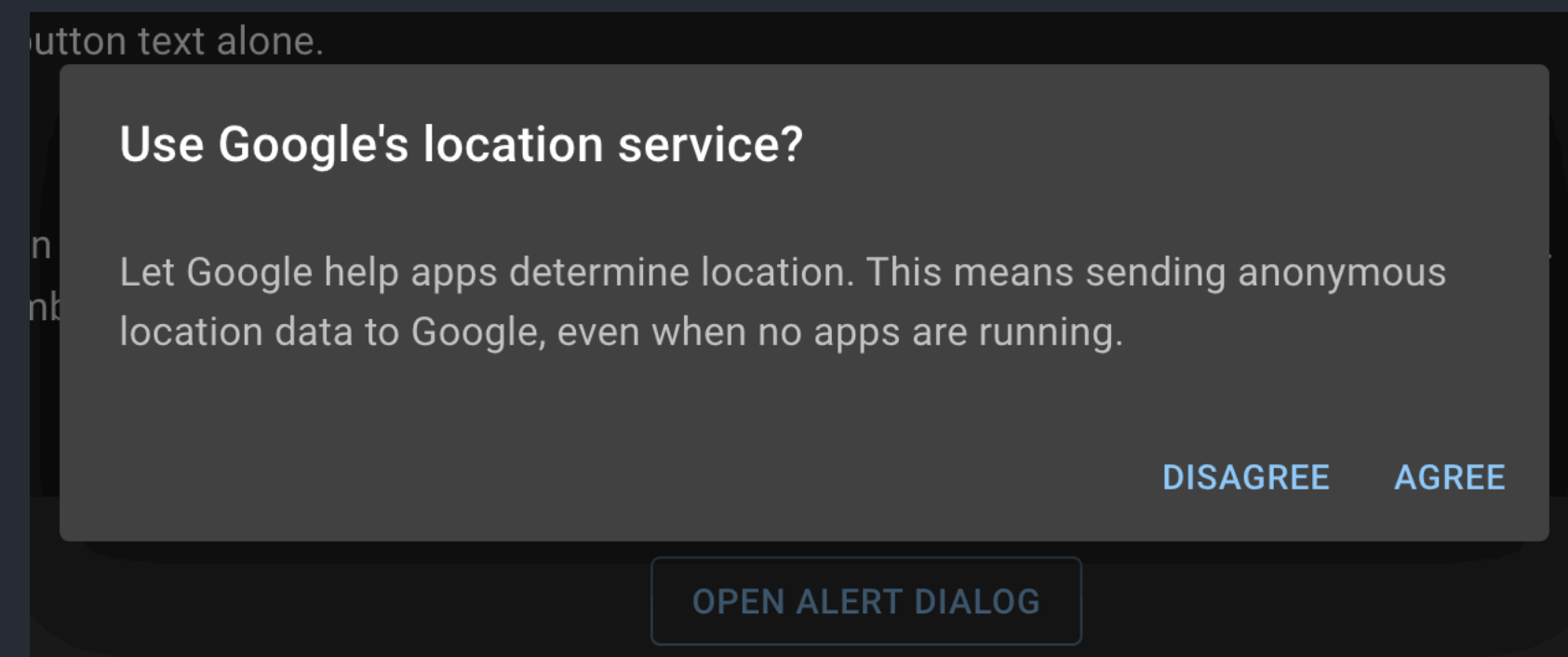


# 프로그레스, 모달

- Progress
  - Circular
  - Linear



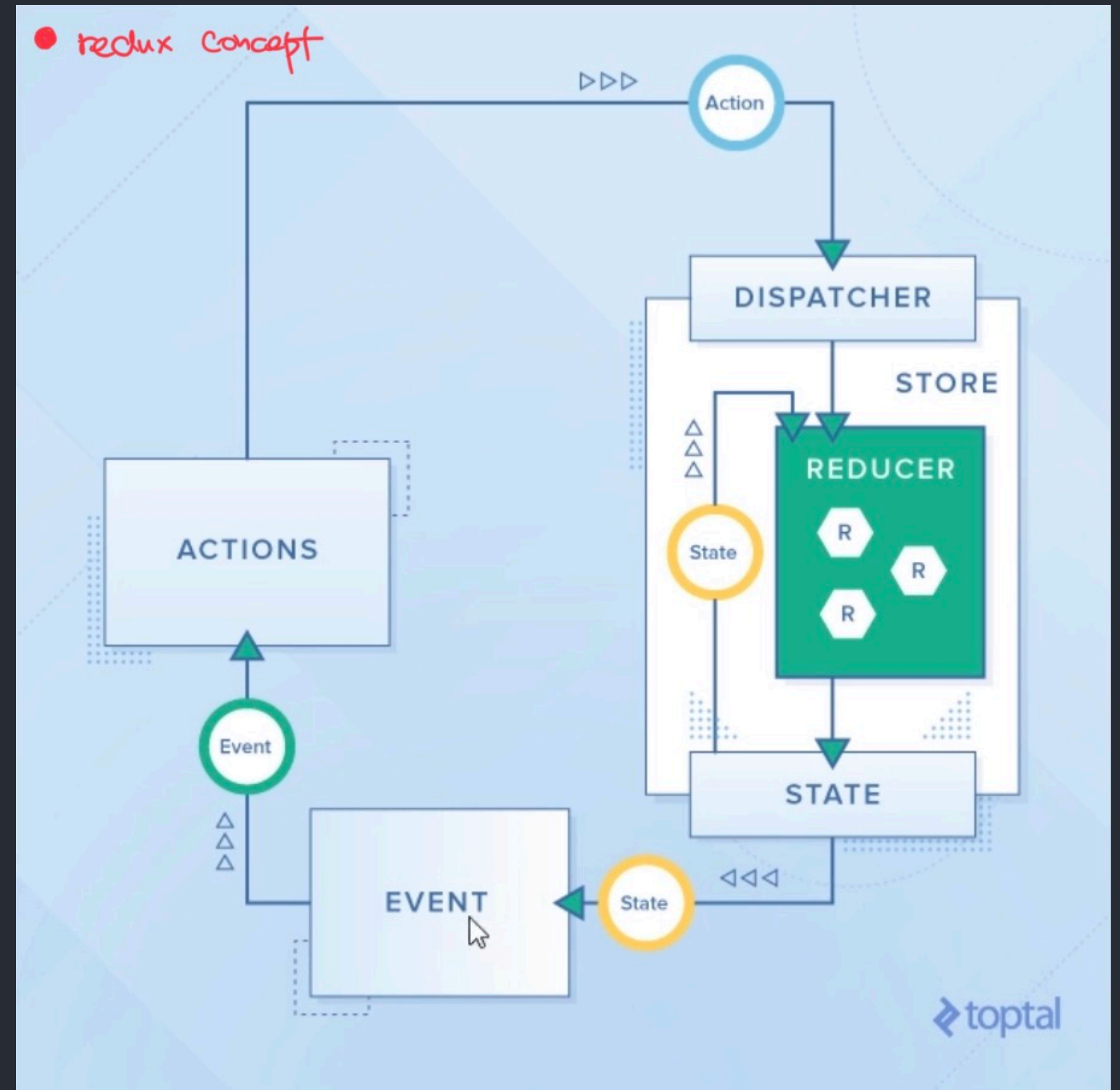
- Modal
  - Dialog
  - Popper
  - Popover
    - The scroll and click away are blocked





# Redux

- Review: React에서의 state 관리
    - props
    - state
    - global state
  - redux: state 관리 javascript 프레임워크
    - actions
    - reducer
    - store
  - Auth 상태를 redux에서 관리
- ```
npm install redux
```
- ```
npm install redux-react
```
- Redux dev tools



# Redux-saga

- 사이드이펙트를 효율적으로 관리하기 위한 redux 미들웨어
- Middleware
  - redux에 추가할 수 있는 미들웨어
- ES6의 Generator 활용: `function*`

```
npm install --save redux-saga
```

# 앞으로는 뭘 더 하나?

- <https://github.com/songminseok/yakmoa.git>
- 다음에는
- 약 관리: 등록/삭제/수정
- 복약이력: 달력 (library)