

# 리액트로 프론트엔드 레벨업

Alex Kwon

애플리케이션 제작하기 9/16



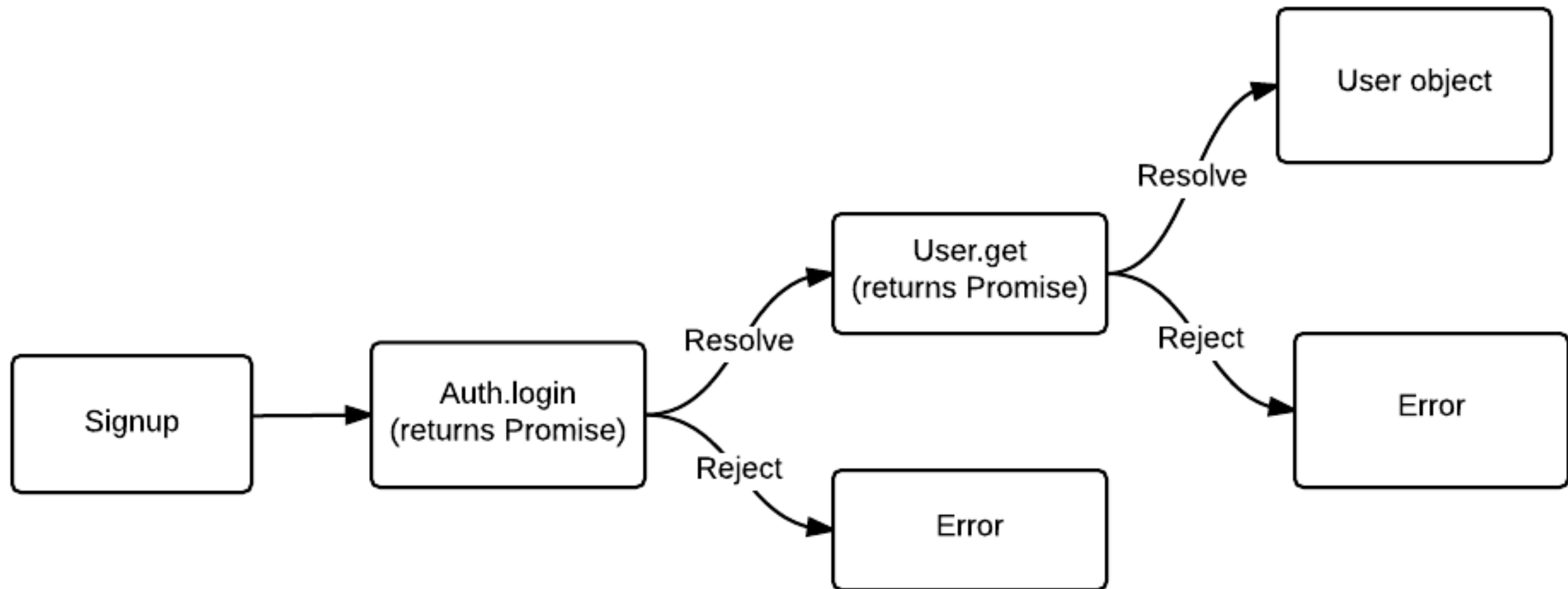
# 애플리케이션 제작하기 9/16

---

- 실습: nyan-forum 만들기
- JS 비동기 프로그래밍
- react-router 를 사용하여 라우팅하기
- firebase, firestore nosql db연동하기

- 비동기 작업을 수행하기 위해 사용
- 작업의 성공 및 실패 결과에 따라 대응
- callback -> promise -> async/await

# JS promise



# 실습: 냥포럼 만들기

---

- 데모: <http://nyan-forum-react.surge.sh/>
- CRA로 새 프로젝트 구성
- `npm install react-router-dom`
- `npm install firebase`

- 라우팅?
- 리액트용 클라이언트 사이드 라우팅
- 컴포넌트 단위의 라우팅

# react-router의 API

---

- <BrowserRouter> - HTML5 history API
- <Route> - 컴포넌트 렌더링
- <Switch> - 한 개의 Route를 렌더링
- <Link> - 라우트 이동
- history - 브라우저의 history 를 제어
- location, match - 현재 url의 정보
- withRouter - history, location, match의 정보를 컴포넌트에 전달

- Firestore - 실시간 db, 인증, 호스팅, 파일 업로드 등의 서비스 제공
- Firestore - nosql기반의 실시간 db



- Document => 한 개의 데이터 객체
- Collection => Document의 집합
- Reference => 특정 데이터의 위치만 참조
- Snapshot => 특정 데이터 값 포함

# Firestore querying

---

- Document
  - `db.collection('posts').doc('1')` => `documentRef`
  - `documentRef.get()` => `documentSnapshot`
  - `documentSnapshot.data()` => `{ id: '1', title: '...' }`
- Collection
  - `db.collection('posts')` => `collectionRef`
  - `collectionRef.get()` => `collectionSnapshot`
  - `collectionSnapshot.docs` => `[ documentSnapshot ]`

# Firestore data creation

---

- `db.collection('posts').set({ title: '...', content: '...' }) => 새 post 생성`
- `db.collection('posts').doc('1').update({ title: '...' }) => 1 번 post 업데이트`

# Firestore - nyan-forum db 모델링

---

- boards
  - id : string
  - name: string
  - posts: [postId]
- Comments
  - id: string
  - author: string
  - content: string
- posts
  - id: string
  - author: string
  - title: string
  - content: string
  - comments: [commentId]



다음에  
또!  
같이!  
만나요!