세상의 속도를
따라잡고 싶다면

Do it!

깡샘의 안드로이드 앱 프로그래밍 with 코틀린

이지스퍼블리싱(주)

20

파이어베이스 연동과 인증 기능

20-1 파이어베이스 이해하기

20-2 파이어베이스 연동하기

20-3 인증 기능 이용하기

20-4 회원가입과 로그인 기능 만들기

이지스퍼블리싱(주)

20-1 파이어베이스 이해하기

파이어베이스란?

- 안드로이드 앱에서 파이어베이스를 이용하면 서버리스 컴퓨팅을 구현할 수 있습니다.

- 파이어베이스는 모바일 앱과 웹 애플리케이션을 개발하는 데 필요한 여러 가지 기능을 제공
  - Authentication: 인증, 회원가입 및 로그인 처리
  - Cloud Message: 알림 전송
  - Firebase Database: 앱 데이터 저장 및 동기화
  - Realtime Database: 실시간 데이터 저장 및 동기화
  - Storage: 파일 저장소
  - Hosting: 웹 호스팅
  - Functions: 서버 관리 없이 모바일 백엔드 코드 실행
  - Machine Learning: 모바일 개발자용 머신러닝

20-1 파이어베이스 이해하기

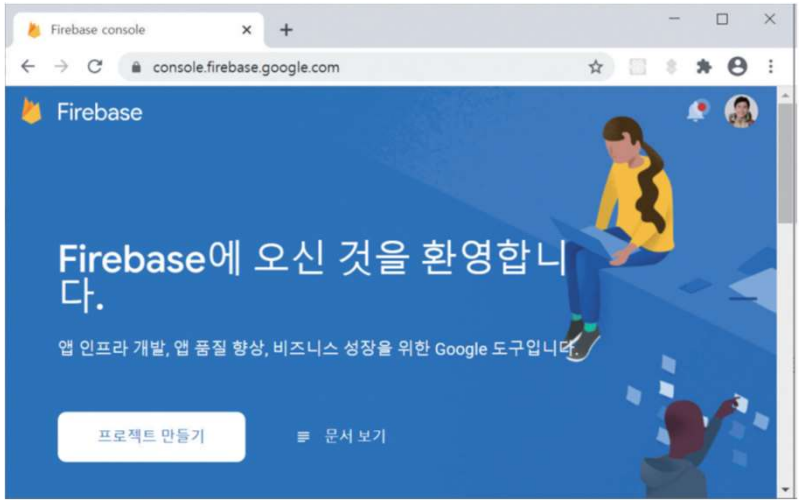
파이어베이스 요구 사항

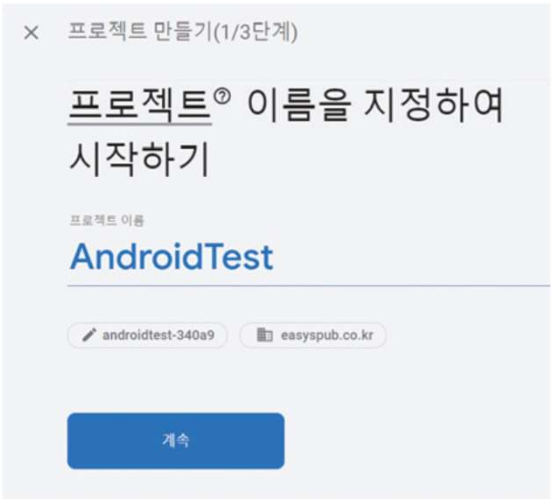
- 안드로이드 앱에서 파이어베이스와 연동을 위한 환경
  - 구글 계정
  - 안드로이드 스튜디오
  - 그래들 4.1 이상
  - minSdk 16 이상

20-2 파이어베이스 연동하기

파이어베이스 프로젝트 생성하기

- console.firebase.google.com에 접속
- 프로젝트 만들기





20-2 파이어베이스 연동하기

파이어베이스에 앱 등록하기

- 안드로이드 앱을 등록
- <google-services.json 다운로드>를 클릭하여 파일을 내려받은 후 안드로이드 앱 모듈의 루트 디렉터리에 복사

20-2 파이어베이스 연동하기

빌드 그래들에 파이어베이스 라이브러리 추가하기

- 프로젝트 수준의 build.gradle 파일

• 프로젝트 수준의 빌드 그래들

```
buildscript {
    (... 생략 ...)
    dependencies {
        classpath 'com.google.gms:google-services:4.3.10'
        (... 생략 ...)
    }
}
```

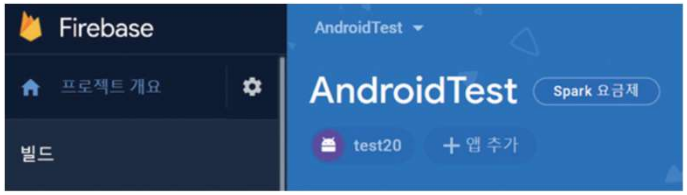
- 모듈 수준의 build.gradle 파일

• 모듈 수준의 빌드 그래들

```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
    id 'com.google.gms.google-services'
}
(... 생략 ...)
dependencies {
    (... 생략 ...)
    implementation platform('com.google.firebase:firebase-bom:29.0.0')
}
```

20-2 파이어베이스 연동하기

파이어베이스 콘솔에서 앱 등록 완료



20-3 인증 기능 이용하기

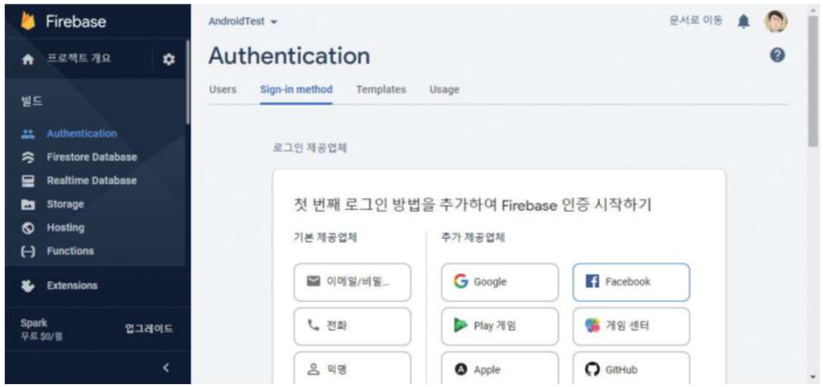
이메일/비밀번호 인증

- 이메일/비밀번호 인증은 사용자의 회원가입 정보인 이메일/비밀번호를 파이어베이스에 저장하고 이를 바탕으로 로그인을 처리하는 방식

- 로그인할 때는 등록된 이메일 서버와 연동하지 않고 파이어베이스에서 처리

20-3 인증 기능 이용하기

- 이메일/비밀번호 인증 사용 설정하기



• 파이어베이스 인증 라이브러리 등록

```
implementation 'com.google.firebase:firebase-auth-ktx:21.0.1'
```

20-3 인증 기능 이용하기

- 파이어베이스 인증 객체 얻기
  - FirebaseAuth 객체는 Firebase.auth로 얻습니다.

• 파이어베이스 인증 객체 얻기

```
lateinit var auth: FirebaseAuth
(... 생략 ...)
auth = Firebase.auth
```

- 회원가입하기
  - FirebaseAuth 객체의 createUserWithEmailAndPassword() 함수로 파이어베이스에 이메일/비밀번호를 등록

• 회원가입하기

```
auth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
    (... 생략 ...)
}
```

20-3 인증 기능 이용하기

- 콜백 함수의 매개변수인 Task<AuthResult> 객체로 회원가입 성공·실패를 판단

• 회원가입 성공·실패 판단

```
auth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            (... 생략 ...)
        } else {
            Log.w("kkang", "createUserWithEmail:failure", task.exception)
        }
    }
```

20-3 인증 기능 이용하기

- 이메일/비밀번호 등록에 성공했을 때 FirebaseUser 클래스의 sendEmailVerification() 함수로 인증 메일 전송

• 인증 메일 발송하기(등록 성공했을 때)

```
auth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            auth.currentUser?.sendEmailVerification()
                ?.addOnCompleteListener { sendTask ->
                    if (sendTask.isSuccessful) {
                        // ...
                    } else {
                        // ...
                    }
                }
        } else {
            Log.w("kkang", "createUserWithEmail:failure", task.exception)
        }
    }
```

20-3 인증 기능 이용하기

- 로그인하기
  - 로그인을 처리할 때는 signInWithEmailAnd Password() 함수를 이용

• 로그인 처리

```
auth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(this) { task ->
        if (task.isSuccessful) {
            // 로그인 성공했을 때 처리
        } else {
            // 로그인 실패했을 때 처리
        }
    }
```

- 사용자 정보 가져오기

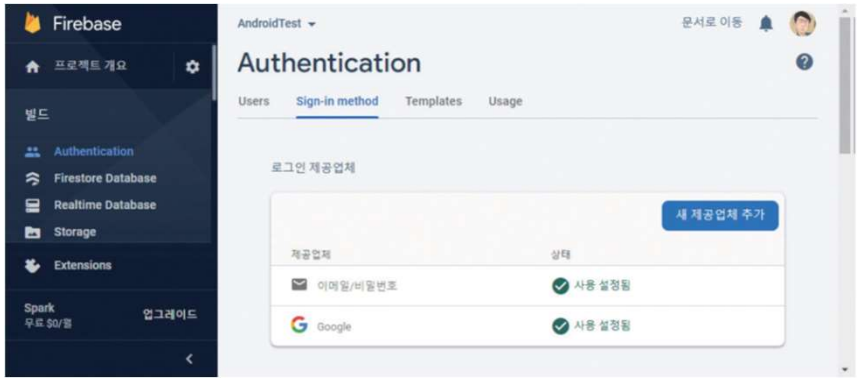
• 사용자 정보 가져오기

```
currentUser?.let {
    val isEmailVerified = currentUser.isEmailVerified
    val email = currentUser.email
    val uid = currentUser.uid
}
```

20-3 인증 기능 이용하기

구글 인증

- 구글 인증 사용 설정하기



• 플레이 서비스 인증 라이브러리 등록

```
implementation 'com.google.android.gms:play-services-auth:19.2.0'
```

20-3 인증 기능 이용하기

- 구글 인증 처리하기
  - 안드로이드 시스템에 설치된 구글의 인증 앱과 연동하여 처리

• 구글 인증 앱 실행

```
val gso = GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken(getString(R.string.default_web_client_id))
    .requestEmail()
    .build()
val signInIntent = GoogleSignIn.getClient(this, gso).signInIntent
requestLauncher.launch(signInIntent)
```

인텐트 객체 생성

인텐트 시작

20-3 인증 기능 이용하기

• 구글 인증 처리

```
val requestLauncher = registerForActivityResult(
    ActivityResultContracts.StartActivityForResult())
{
    val task = GoogleSignIn.getSignedInAccountFromIntent(it.data)
    try {
        val account = task.getResult(ApiException::class.java)!!
        val credential = GoogleAuthProvider.getCredential(account.idToken, null)
        MyApplication.auth.signInWithCredential(credential)
            .addOnCompleteListener(this) { task ->
                if (task.isSuccessful) {
                    // 구글 로그인 성공
                } else
                    // 구글 로그인 실패
                }
            }
    } catch (e: ApiException) {
        // 예외 처리
    }
}
```

Do it! 실습

회원가입과 로그인 기능 만들기

1단계. 이메일/비밀번호 인증 설정하기

- 파이어베이스 콘솔에서 이메일/비밀번호 인증을 사용하겠다고 설정

2단계. 구글 인증 설정하기

- 구글 인증을 활성화

3단계. 빌드 그래들 설정하기

- 파이어베이스 이메일/비밀번호와 구글 인증을 사용하는 라이브러리를 등록

Do it! 실습 시계 앱의 스톱워치 기능 만들기

4단계. MyApplication 클래스 작성과 등록

- MyApplication 클래스를 작성하고 매니페스트에 등록

5단계. 인증 처리 액티비티 추가하기

- AuthActivity라는 이름으로 새로운 액티비티를 만듭니다.

6단계. 실습 파일 복사하기

- layout과 menu 두 디렉터리를 현재 모듈의 res 디렉터리 아래에 복사
- 코틀린 소스가 있는 디렉터리의 AuthActivity.kt와 MainActivity.kt 파일을 앱의 소스 영역에 복사

Do it! 실습

시계 앱의 스톱워치 기능 만들기

7단계. 인증 액티비티 작성하기

- AuthActivity.kt 파일 작성

8단계. 앱 실행하기

감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.

윌리엄 셰익스피어
William Shakespeare