

# 앱 개발 입문

성균관대학교 컬처애편테크놀로지융합전공 하계 부트캠프

7일차 - 2022. 08. 04 (목)

## 강의 슬라이드 링크

[https://github.com/kunny/skku-bootcamp-2022-summer/blob/main/\\_slides/day7.pdf](https://github.com/kunny/skku-bootcamp-2022-summer/blob/main/_slides/day7.pdf)

# 오늘 강의에서 다룰 내용

- 노트 앱에 Firebase를 적용하고, 사용자들의 앱 사용 패턴을 분석할 수 있도록 이벤트를 추가합니다.
- Firebase에서 특정 조건에 맞는 사용자를 그룹으로 관리하는 방법을 알아봅니다.
- REST API를 알아봅니다.
- 사진 검색 앱 프로젝트를 만들고, 파이어베이스 프로젝트를 설정합니다.
- 사진 검색 앱에서 사용할 앱 이벤트를 작성합니다.
- 사진 검색 API를 위한 코드를 작성합니다.

# Firebase CLI 설치




# Firebase CLI (Command Line Interface)

- Firebase 프로젝트와 관련된 다양한 작업을 할 수 있는 명령줄(command line) 도구입니다.
- Flutter 애플리케이션에 Firebase 프로젝트를 자동으로 연결할 때 사용할 도구인 **flutterfire\_cli** 를 사용하려면 Firebase CLI를 미리 설치해야 합니다.

# Firebase CLI 설치하기 (Windows)

- Firebase CLI를 구동할 때 필요한 추가 도구인 nvm-windows를 설치해야 합니다.
- [이 링크](#)를 클릭하여 nvm-windows 설치 파일을 받을 수 있는 페이지로 이동합니다.

**1.1.9** Latest Compare

 coreybutler released this Dec 11, 2021 · 17 commits to master since this release  

NOTICE: If you downloaded this version before Dec 15, 2021 and are affected by issue [#706](#), please re-download. This fix was not present in the initial binary and has since been corrected.

### What's Changed

- Code signed release.
- Fix broken link to nodejs versions by [@crhstianramirez](#) in [#679](#)
- Fix readme links. by [@veleek](#) in [#689](#)
- docs: update address of china mirror by [@hyj1991](#) in [#692](#)
- Use Windows defined env vars in all paths in README by [@gdziadkiewicz](#) in [#699](#)

### New Contributors

- [@crhstianramirez](#) made their first contribution in [#679](#)
- [@veleek](#) made their first contribution in [#689](#)
- [@hyj1991](#) made their first contribution in [#692](#)
- [@gdziadkiewicz](#) made their first contribution in [#699](#)
- [@ajyong](#) sponsored the code signing certificate

Full Changelog: [1.1.8...1.1.9](#)

### UPDATE 4/27/22

Added nvm-setup.exe after initial release to support winget efforts.

# Firebase CLI 설치하기 (Windows)

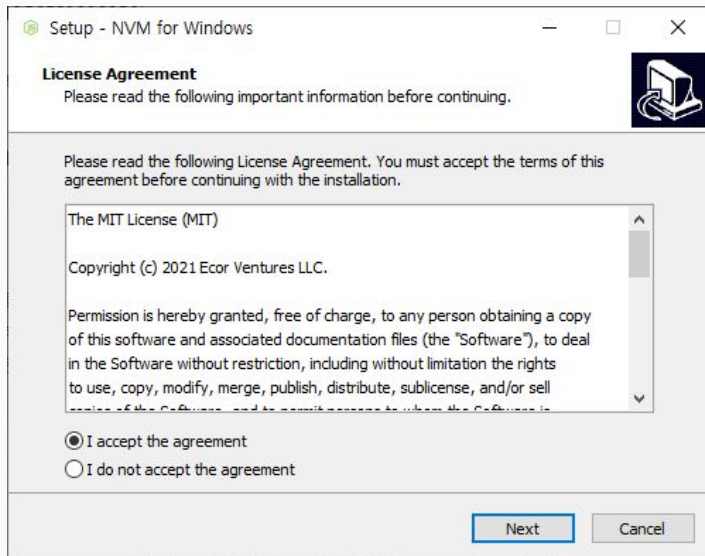
설치 파일 (nvm-setup.exe)을 다운로드 합니다.

## ▼ Assets 9

 <a href="#">nvm-noinstall.zip</a>	3.64 MB	Dec 16, 2021
 <a href="#">nvm-noinstall.zip.checksum.txt</a>	34 Bytes	Dec 16, 2021
 <a href="#">nvm-setup.exe</a>	4.64 MB	Apr 28, 2022
 <a href="#">nvm-setup.zip</a>	4.14 MB	Dec 16, 2021
 <a href="#">nvm-setup.zip.checksum.txt</a>	34 Bytes	Dec 16, 2021
 <a href="#">nvm-update.zip</a>	3.45 MB	Dec 16, 2021
 <a href="#">nvm-update.zip.checksum.txt</a>	34 Bytes	Dec 16, 2021
 <a href="#">Source code</a> (zip)		Dec 11, 2021
 <a href="#">Source code</a> (tar.gz)		Dec 11, 2021

# Firebase CLI 설치하기 (Windows)

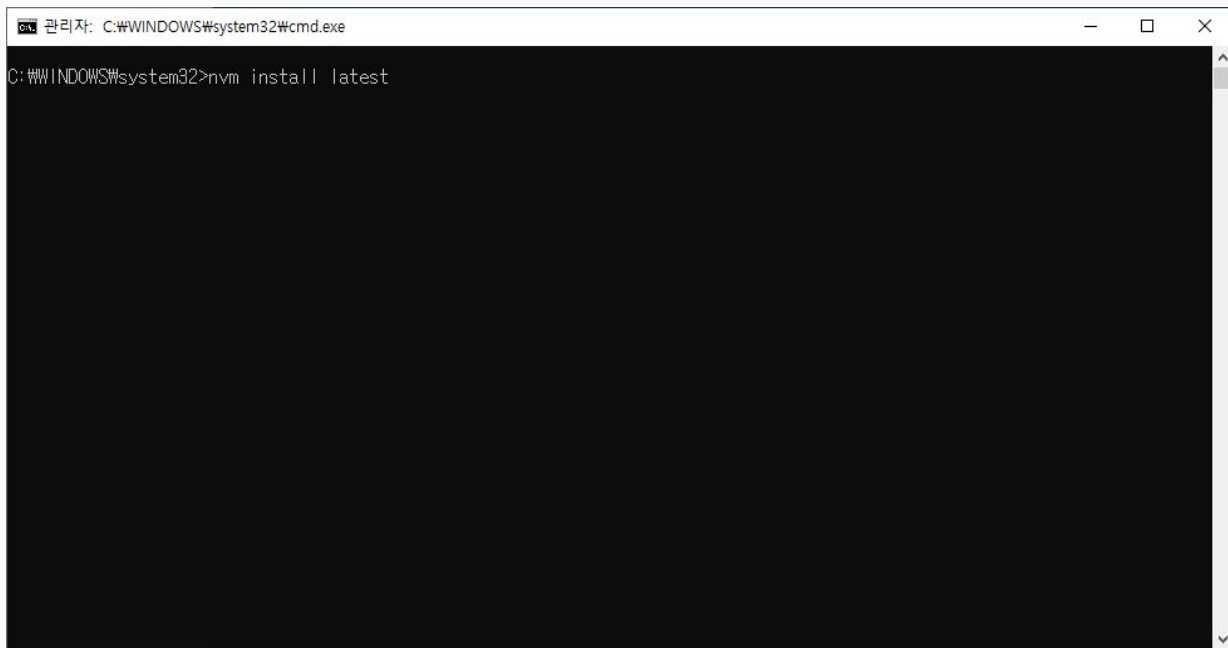
설치 마법사가 실행되면, 기본 옵션 그대로 설치를 완료합니다.





# Firebase CLI 설치하기 (Windows)

**관리자 모드**로 명령 프롬프트를 실행한 후, **`nvm install latest`** 명령어를 실행하여 Node.js 런타임을 설치합니다. (Firebase CLI 실행에 필요한 구성요소)



```
관리자: C:\WINDOWS\system32\cmd.exe
C:\WINDOWS\system32>nvm install latest
```

# Firebase CLI 설치하기 (Windows)

설치가 완료되면 설치된 버전이 화면이 표시됩니다. `nvm use [설치된 버전]` 명령을 실행하여 설치한 버전을 활성화합니다. (예: 설치한 버전이 18.7.0인 경우 `nvm use 18.7.0` 실행)



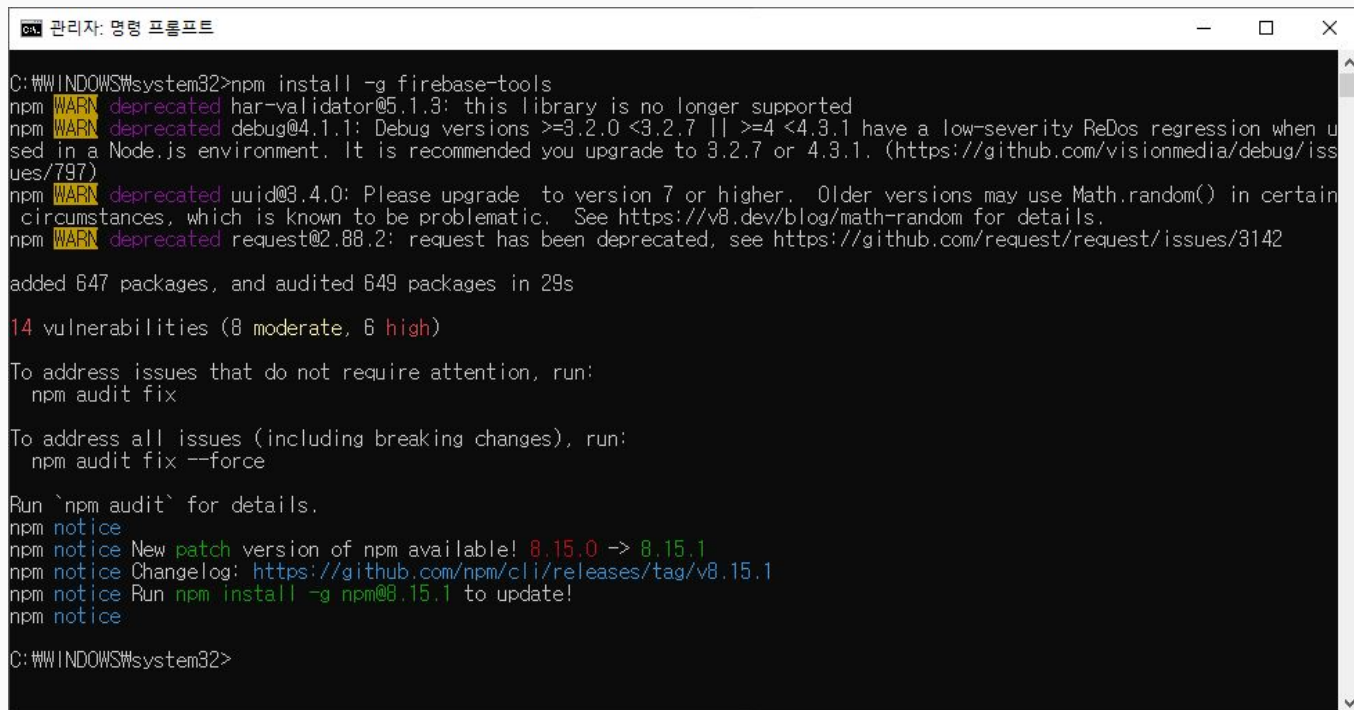
```
관리자: 명령 프롬프트
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>nvm use 18.7.0
Now using node v18.7.0 (64-bit)

C:\WINDOWS\system32>
```

# Firebase CLI 설치하기 (Windows)

`npm install -g firebase-tools` 명령을 실행하여 Firebase CLI를 설치합니다.



```
C:\WINDOWS\system32>npm install -g firebase-tools
npm WARN deprecated har-validator@5.1.3: this library is no longer supported
npm WARN deprecated debug@4.1.1: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142

added 647 packages, and audited 649 packages in 29s

14 vulnerabilities (8 moderate, 6 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
npm notice
npm notice New patch version of npm available! 8.15.0 -> 8.15.1
npm notice Changelog: https://github.com/npm/cli/releases/tag/v8.15.1
npm notice Run npm install -g npm@8.15.1 to update!
npm notice
C:\WINDOWS\system32>
```

## Firebase CLI 설치하기 (macOS)

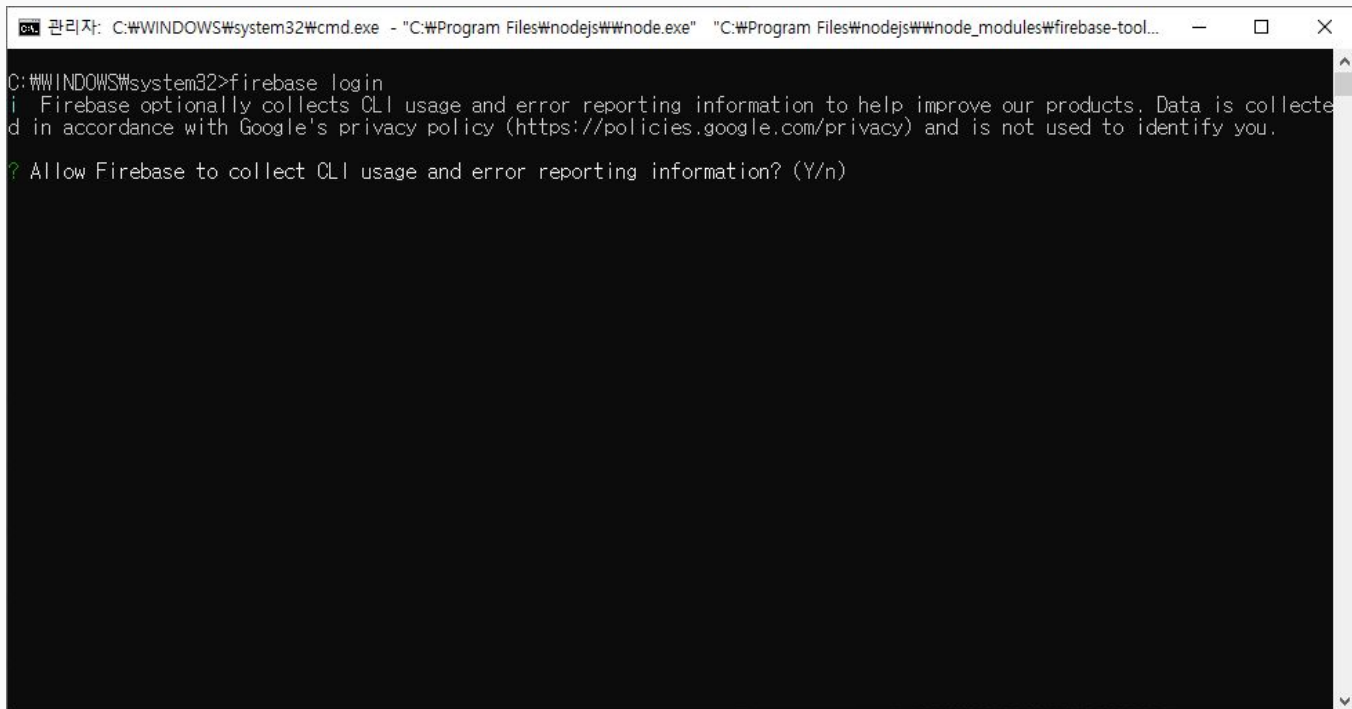
터미널을 연 후, 아래 명령어를 실행하여 Firebase CLI를 설치합니다.

```
curl -sL https://firebase.tools | bash
```

# Firestore CLI 구성 및 flutterfire\_cli 설치

# Firebase CLI 구성 및 flutterfire\_cli 설치

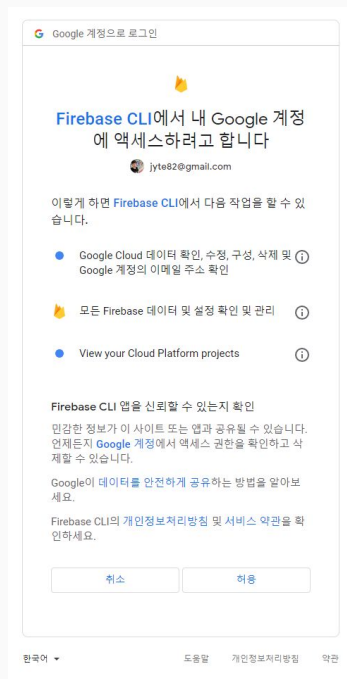
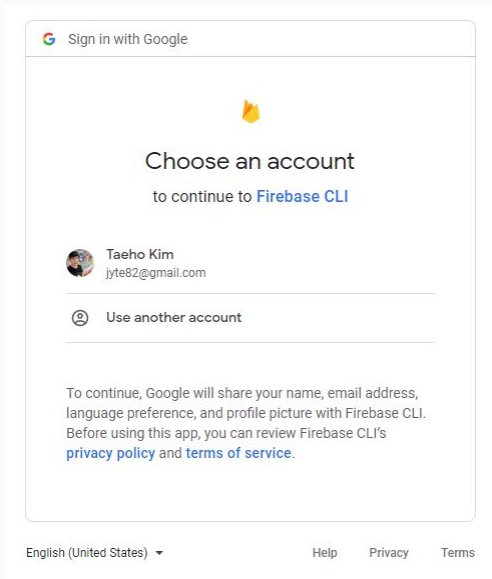
명령 프롬프트/터미널을 실행한 후, **firebase login** 명령을 실행합니다.



```
관리자: C:\WINDOWS\system32\cmd.exe - "C:\Program Files\nodejs\node.exe" "C:\Program Files\nodejs\node_modules\firebase-tool...
C:\WINDOWS\system32>firebase login
i Firebase optionally collects CLI usage and error reporting information to help improve our products. Data is collected
in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.
? Allow Firebase to collect CLI usage and error reporting information? (Y/n)
```

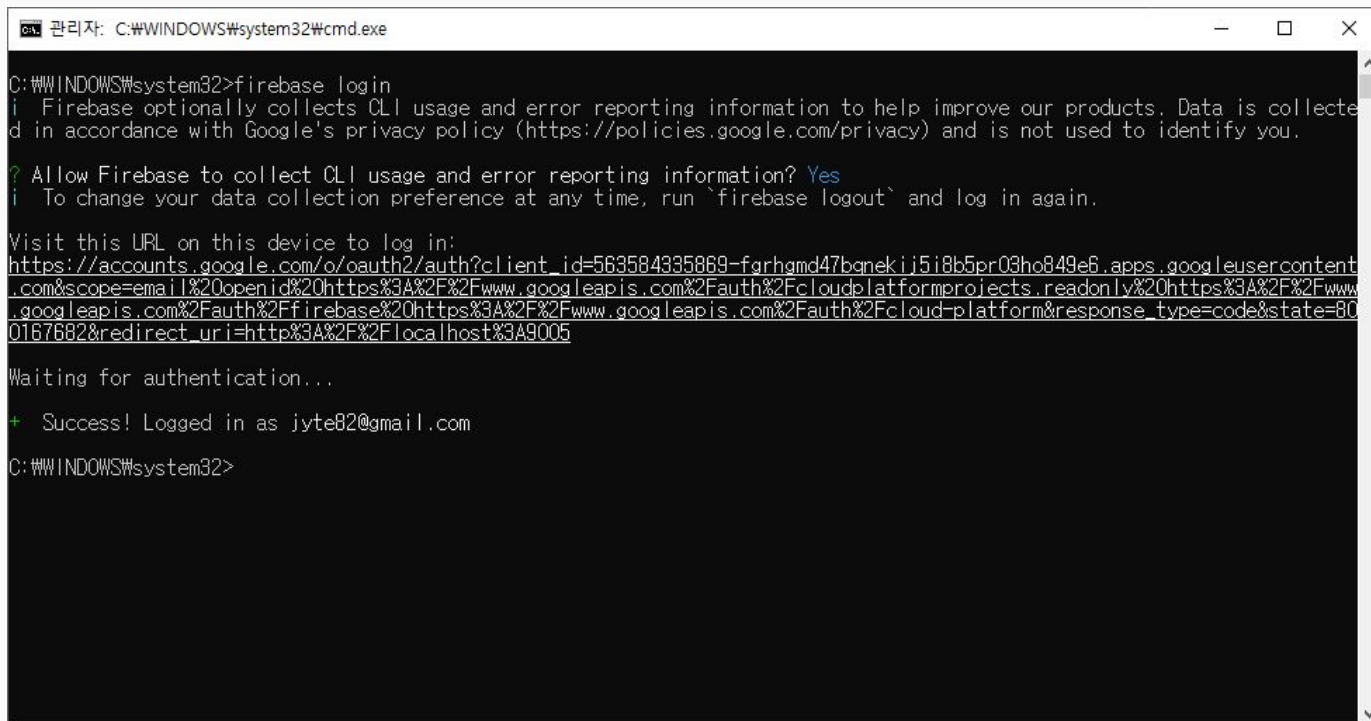
# Firebase CLI 구성 및 flutterfire\_cli 설치

웹 브라우저가 표시되며 로그인 화면이 표시됩니다. Firebase 프로젝트를 관리할 구글 계정으로 로그인한 후, Firebase CLI에서 계정과 연결된 파이어베이스 데이터에 접근할 수 있도록 권한을 부여합니다.



# Firebase CLI 구성 및 flutterfire\_cli 설치

로그인이 완료되면 아래와 같이 로그인 성공 메시지가 표시됩니다.



```
C:\WINDOWS\system32\cmd.exe

C:\WINDOWS\system32>firebase login
i  Firebase optionally collects CLI usage and error reporting information to help improve our products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI usage and error reporting information? Yes
i  To change your data collection preference at any time, run `firebase logout` and log in again.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client_id=563584335869-fgrhmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response_type=code&state=8001b7682&redirect_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...

+ Success! Logged in as jyte82@gmail.com

C:\WINDOWS\system32>
```



## Firebase CLI 구성 및 flutterfire\_cli 설치

다음 명령어를 실행하여 Flutter 앱에 파이어베이스 프로젝트를 자동으로 연결해주는 도구인 **flutterfire\_cli**를 설치합니다.

```
dart pub global activate flutterfire_cli
```

## flutterfire\_cli 실행을 위한 PATH 설정 (Windows)

1일차 강의자료 53-58 슬라이드를 참고하여 아래 경로를 시스템 변수의 Path 항목에 추가합니다.

```
C:\Users\[사용자 계정 이름]\AppData\Local\Pub\Cache\bin
```

예) 사용자 계정 이름이 jyte8 인 경우

```
C:\Users\jyte8\AppData\Local\Pub\Cache\bin
```

## flutterfire\_cli 실행을 위한 PATH 설정 (macOS)

1일차 강의자료 70-73 슬라이드를 참고하여 다음 내용을 `.bash_profile` 혹은 `.zshrc` 파일에 추가합니다.

```
export PATH="$PATH:$HOME/.pub-cache/bin"
```

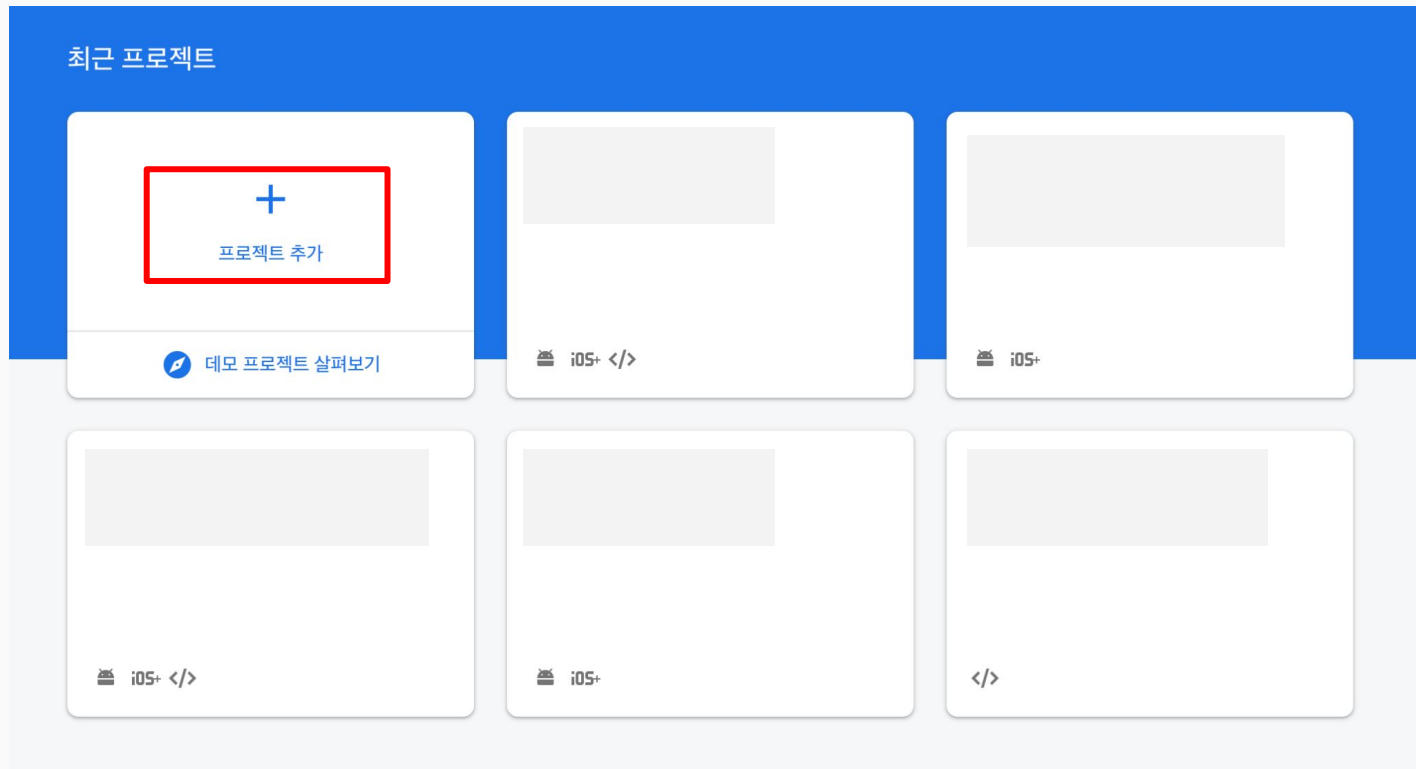
## flutterfire\_cli 실행을 위한 PATH 설정 (공통)

안드로이드 스튜디오 내 터미널에서 `flutterfire_cli` 툴을 사용할 수 있도록 안드로이드 스튜디오를 재시작합니다.

# Firebase 프로젝트 생성하기

# Firebase 프로젝트 생성하기

[console.firebase.google.com](https://console.firebase.google.com)에 접속한 후, 프로젝트 추가 버튼을 누릅니다.



# Firebase 프로젝트 생성하기

프로젝트 이름을 지정합니다. (예: androidhuman-sticky-notes)

× 프로젝트 만들기(1/3단계)

프로젝트 이름을 지정하여 시작하  
기<sup>②</sup>

프로젝트 이름 입력

my-awesome-project-id

계속



# Firebase 프로젝트 생성하기



Google 애널리틱스를 사용 설정합니다.



## Firebase 프로젝트를 위한 Google 애널리틱스



무제한 무료 분석 솔루션인 Google 애널리틱스를 사용하면 Firebase Crashlytics, 클라우드 메시징, 인앱 메시징, 원격 구성, A/B 테스트, Cloud Functions에서 타겟팅, 보고 등을 이용할 수 있습니다.



Google 애널리틱스를 통해 다음 기능을 이용할 수 있습니다.

 A/B 테스트 

 장애가 발생하지 않은 사용자 

 Firebase 제품 전반에서 사용자 세분화 및 타  
겟팅 

 이벤트 기반 Cloud Functions 트리거 

 제한 없는 무료 보고 

☒ 이 프로젝트에서 Google 애널리틱스 사용 설정  
권장

[이전](#)

[계속](#)



# Firebase 프로젝트 생성하기

Google 애널리틱스 계정을 선택하거나 새 계정을 생성한 후, 프로젝트 만들기 버튼을 누릅니다.

## Google 애널리틱스 구성

Google 애널리틱스 계정 선택 또는 만들기 ?



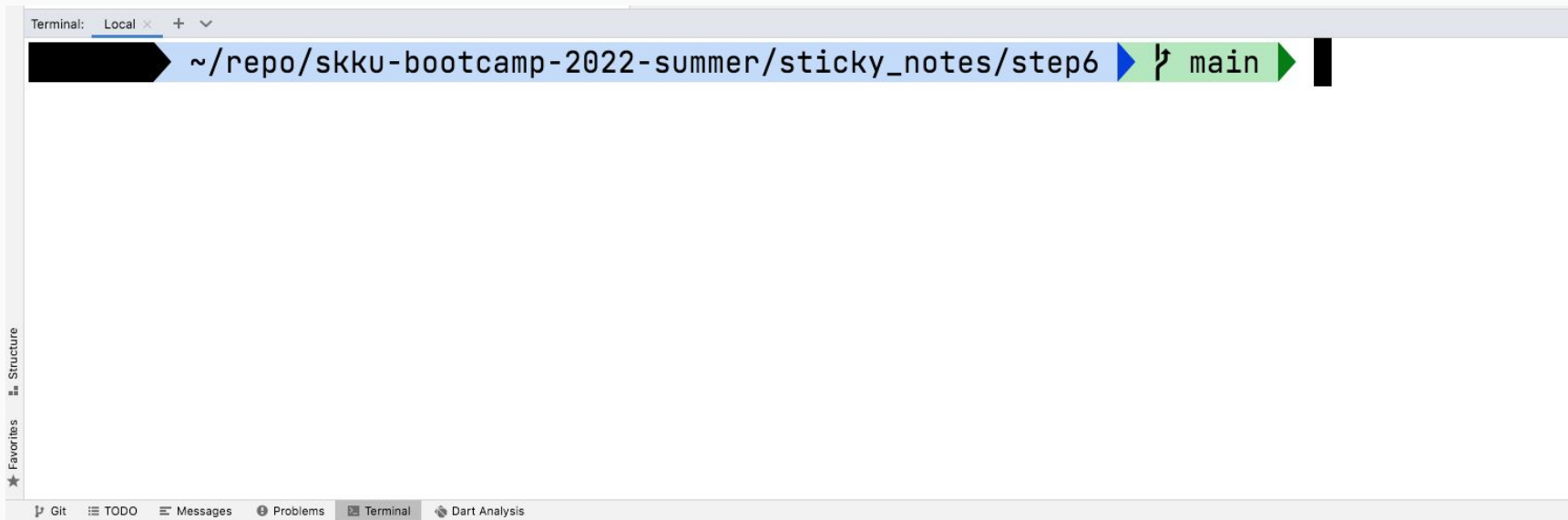
Analytics 속성이 생성되고 Firebase 프로젝트에 연결됩니다. 이 연결을 통해 제품  
base로 내보낸 데이터에는 Firebase 서비스 약관이 적용되지만 Google 애널리  
이 적용됩니다. [자세히 알아보기](#)

프로젝트 만들기

# 노트 앱 프로젝트에 Firebase 설정하기

# 노트 앱 프로젝트에 Firebase 설정하기

안드로이드 스튜디오 하단의 Terminal 탭을 클릭하여 터미널을 실행합니다.



# 노트 앱 프로젝트에 Firebase 설정하기

`flutter pub add firebase_core` 명령어를 실행하여 프로젝트에 파이어베이스 패키지를 추가합니다.

```
$ flutter pub add firebase_core
Resolving dependencies...
  async 2.8.2 (2.9.0 available)
  characters 1.2.0 (1.2.1 available)
  clock 1.1.0 (1.1.1 available)
  fake_async 1.3.0 (1.3.1 available)
  matcher 0.12.11 (0.12.12 available)
  material_color_utilities 0.1.4 (0.1.5 available)
  meta 1.7.0 (1.8.0 available)
  path 1.8.1 (1.8.2 available)
  source_span 1.8.2 (1.9.1 available)
  string_scanner 1.1.0 (1.1.1 available)
  term_glyph 1.2.0 (1.2.1 available)
  test_api 0.4.9 (0.4.12 available)
Got dependencies!
```

# 노트 앱 프로젝트에 Firebase 설정하기

[pubspec.yaml](#) 의 dependencies 섹션에 firebase\_core 패키지가 추가되었는지 확인합니다.

```
dependencies:  
  firebase_core: ^1.20.0  
  flutter:  
    sdk: flutter  
  sqflite: ^2.0.3
```

# 노트 앱 프로젝트에 Firebase 설정하기

다시 터미널 탭을 열고, **flutterfire configure** 명령어를 실행하여 파이어베이스 프로젝트를 연결을 시작합니다. 표시되는 프로젝트 중 연결할 프로젝트를 선택합니다.

```
i Found 9 Firebase projects.  
? Select a Firebase project to configure your Flutter application with ›  
  androidhuman-blog (androidhuman)  
  androidhuman-cubewhere (Cubewhere)  
  androidhuman-gc (androidhuman-gc)  
  androidhuman-my-photo (androidhuman-my-photo)  
  androidhuman-mycloud (myCloud)  
  androidhuman-myplaces (myPlaces)  
  androidhuman-newtown3 (Newtown 3)  
  androidhuman-pb (Pregnancy benefits)  
› androidhuman-sticky-notes (androidhuman-sticky-notes)  
  tesla-price-calculator (Tesla Price Calculator)  
  <create a new project>
```

# 노트 앱 프로젝트에 Firebase 설정하기

연결할 플랫폼을 선택합니다. 안드로이드와 iOS를 선택한 후 다음 단계로 진행합니다.

? Which platforms should your configuration support (use arrow keys & space to `select`)? ›

✓ android

✓ ios

macos

web

# 노트 앱 프로젝트에 Firebase 설정하기

Flutter 프로젝트에 있는 안드로이드/iOS 앱 설정을 기반으로 Firebase 프로젝트에 앱이 추가되고, 이 설정 정보가 담긴 파일 (**firebase\_options.dart**)이 프로젝트에 추가됩니다.

```
i Firebase android app com.androidhuman.skku.stickynotes registered.  
i Firebase ios app com.androidhuman.skku.stickynotes registered.
```

Firebase configuration file lib/firebase\_options.dart generated successfully with the following Firebase apps:

Platform	Firebase App Id
android	1:356563551621:android:049331c3dfea914d51b0ec
ios	1:356563551621:ios:9757ebd3cd9e5d7e51b0ec

Learn more about using this file and next steps from the documentation:  
> <https://firebase.google.com/docs/flutter/setup>



# 노트 앱 프로젝트에 Firebase 설정하기

Firebase 서비스를 사용하려면 앱 시작 시점에 초기화가 필요합니다.

**Firestore.initializeApp()**을 호출하여 초기화를 해 줍니다.

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firestore.initializeApp(options: DefaultFirestoreOptions.currentPlatform);  
  runApp(MyApp());  
}
```

main.dart

# 안드로이드 프로젝트 수정하기

파이어베이스 SDK를 사용하려면 앱이 지원하는 최소 안드로이드 버전을 조정해야 합니다.

**android/app/build.gradle** 파일을 연 후, **minSdkVersion**을 21로 변경합니다.

```
android {  
    ...  
    defaultConfig {  
        ...  
        minSdkVersion 21  
        targetSdkVersion flutter.targetSdkVersion  
        versionCode flutterVersionCode.toInteger()  
        versionName flutterVersionName  
    }  
}
```

## 완성된 코드

[https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/sticky\\_notes/step5](https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/sticky_notes/step5)

노트 앱에 앱 이벤트 추가하기

## 노트 앱에 앱 이벤트 추가하기

1. 프로젝트에 `firebase_analytics` 패키지를 추가합니다.
2. `service` 폴더 아래에 `analytics_service.dart` 파일을 추가합니다.
3. 앱 이벤트를 담당하는 `AnalyticsService`를 구현합니다.
4. `AnalyticsService`에 정의한 함수들을 활용하여 노트 화면 페이지에 앱 이벤트를 추가합니다.

## 노트 앱에 앱 이벤트 추가하기

`pubspec.yaml` 의 `dependencies` 섹션에 `firebase_analytics` 패키지를 추가합니다.

```
dependencies:  
  firebase_analytics: ^9.3.0  
  flrebae_core: ^1.20.0  
  flutter:  
    sdk: flutter  
  sqflite: ^2.0.3
```

# 노트 앱에 앱 이벤트 추가하기

- 앱 이벤트는 **이벤트 이름**과 **매개변수**로 구성됩니다.
- 이벤트 이름 (필수)
  - 유의미한 사용자의 행동을 대상으로 하며, 최대한 직관적인 이름을 사용합니다.
- 매개변수 (선택)
  - 이벤트에 추가로 전달할 정보가 있을 때 사용합니다.
  - 예)
    - 설정을 변경하는 이벤트인 경우, 사용자가 설정값 정보

# 노트 앱에 앱 이벤트 추가하기

- 추가할 앱 이벤트
  - `add_note`: 새 노트를 추가했을 때 기록합니다.
  - `change_view_mode`: 노트 보기 옵션을 변경할 때마다 기록합니다.
    - `mode`: 선택한 보기 옵션을 기록합니다. (grid/list)
  - `delete_note`: 노트를 삭제했을 때 기록합니다.
  - `update_note`: 노트를 수정했을 때 기록합니다.
  - `view_note`: 노트를 조회했을 때 기록합니다.



# 노트 앱에 앱 이벤트 추가하기

```
class AnalyticsService {  
  void addNote() {  
    FirebaseAnalytics.instance.logEvent(name: 'add_note');  
  }  
  
  void changeViewMode(bool isGridMode) {  
    FirebaseAnalytics.instance.logEvent(  
      name: 'change_view_mode',  
      parameters: {  
        'mode': isGridMode ? 'grid' : 'list',  
      },  
    );  
  }  
  
  void deleteNote() {  
    FirebaseAnalytics.instance.logEvent(name: 'delete_note');  
  }  
  
  void updateNote() {  
    FirebaseAnalytics.instance.logEvent(name: 'update_note');  
  }  
  
  void viewNote() {  
    FirebaseAnalytics.instance.logEvent(name: 'view_note');  
  }  
}
```

## 노트 앱에 앱 이벤트 추가하기

```
AnalyticsService? _analyticsService;  
  
AnalyticsService analyticsService() {  
  if (_analyticsService == null) {  
    _analyticsService = AnalyticsService();  
  }  
  return _analyticsService!;  
}
```

# 노트 앱에 앱 이벤트 추가하기

```
class _NotelistPageState extends State<NotelistPage> {  
  ...  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Sticky Notes'),  
        actions: [  
          IconButton(  
            icon: Icon(_showAsGrid ? Icons.list : Icons.grid_view),  
            tooltip: _showAsGrid ? '목록으로 보기' : '격자로 보기',  
            onPressed: () {  
              setState(() {  
                _showAsGrid = !_showAsGrid;  
                analyticsService().changeViewMode(_showAsGrid);  
              });  
            },  
          ),  
        ],  
      ),  
      ...  
    )  
  }  
}
```

note\_list\_page.dart

# 노트 앱에 앱 이벤트 추가하기

```
class _NoteViewState extends State<NoteViewPage> {  
  @override  
  Widget build(BuildContext context) {  
    return FutureBuilder<Note>(  
      future: noteService().getNote(widget.id),  
      builder: (context, snap) {  
        if (snap.connectionState == ConnectionState.waiting) {  
          return const Center(  
            child: CircularProgressIndicator(),  
          );  
        }  
  
        if (snap.hasError) {  
          return Scaffold(  
            appBar: AppBar(),  
            body: const Center(  
              child: Text('오류가 발생했습니다'),  
            ),  
          );  
        }  
  
        final note = snap.requireData;  
        analyticsService().viewNote();  
        return Scaffold( ... );  
      }  
    );  
  }  
}
```

note\_view\_page.dart

# 노트 앱에 앱 이벤트 추가하기

```
class _NoteViewPageState extends State<NoteViewPage> {  
  ...  
  void _confirmDelete(int id) {  
    showDialog(  
      context: context,  
      builder: (context) {  
        return AlertDialog(  
          title: const Text('노트 삭제'),  
          content: const Text('노트를 삭제할까요?'),  
          actions: [  
            TextButton(  
              child: const Text('아니오'),  
              onPressed: () {  
                Navigator.pop(context);  
              },  
            ),  
            TextButton(  
              child: const Text('예'),  
              onPressed: () {  
                analyticsService().deleteNote();  
                noteService().deleteNote(id);  
                Navigator.popUntil(context, (route) => route.isFirst);  
              },  
            ),  
          ],  
        );  
      },  
    );  
  }  
}
```

note\_view\_page.dart

# 노트 앱에 앱 이벤트 추가하기

```
class _NoteEditPageState extends State<NoteEditPage> {  
  ...  
  void _saveNote() {  
    if (bodyController.text.isNotEmpty) {  
      final note = Note(  
        bodyController.text,  
        title: titleController.text,  
        color: color,  
      );  
  
      final noteId = widget.id;  
      if (noteId != null) {  
        analyticsService().updateNote();  
        noteService().updateNote(noteId, note);  
      } else {  
        analyticsService().addNote();  
        noteService().addNote(note);  
      }  
  
      Navigator.pop(context);  
    } else {  
      ScaffoldMessenger.of(context).showSnackBar(const SnackBar(  
        content: Text('노트를 입력하세요.'),  
        behavior: SnackBarBehavior.floating,  
      )));  
    }  
  }  
}
```

note\_edit\_page.dart



지난 28일 7월 6일~2022년 8월 2일  
Compare: 6월 8일~2022년 7월 5일



이벤트 수정

이벤트 만들기

추천 이벤트



sign\_up 추천 이벤트(일반)

login 추천 이벤트(일반)

share 추천 이벤트(일반)

기존 이벤트



이벤트 이름 ↑	수	변동률(%)	사용자	변동률(%)	전환으로 표시
add_note	1	-	1	-	<input checked="" type="checkbox"/>
app_clear_data	1	-	1	-	<input type="checkbox"/>
change_view_mode	3	-	1	-	<input checked="" type="checkbox"/>
delete_note	4	-	1	-	<input checked="" type="checkbox"/>
first_open	2	-	2	-	<input type="checkbox"/>
screen_view	7	-	2	-	<input checked="" type="checkbox"/>
session_start	4	-	2	-	<input checked="" type="checkbox"/>
update_note	4	-	1	-	<input checked="" type="checkbox"/>

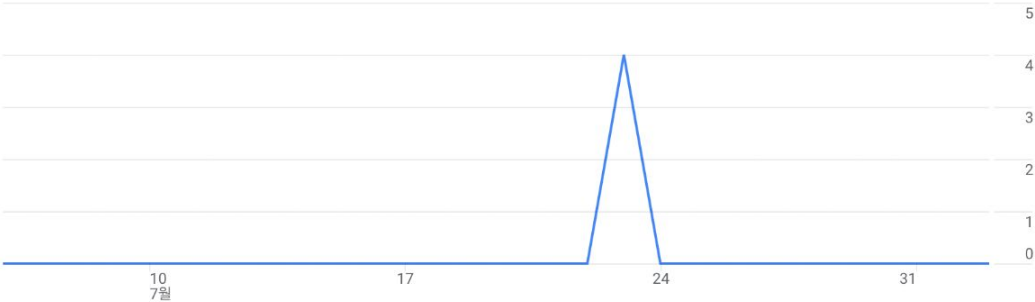
update\_note ▾ ⚠ ▾

지난 28일 7월 6일~2022년 8월 2일 ▾



모든 사용자 + 비교 추가 +

이벤트 수    총 사용자    사용자당 이벤트 수    이벤트 값  
4    1    4    0



지난 30분 동안의 이벤트

0

매개변수 이름

- ▾

매개변수 값    횟수    %

실시간 보기 →

국가 별 이벤트 수



국가	이벤트 수
South Korea	4

성별 별 이벤트 수



사용 가능한 데이터 없음



맞춤 정의

할당량 정보

맞춤 측정기준 맞춤 측정항목

맞춤 측정기준 만들기

측정기준 이름 ↑	설명	범위	사용자 속성/매개변수	마지막 변경일	
mode		이벤트	mode	2022년 8월 3일	⋮

Items per page: 25 1 - 1 of 1 < > >>

MODE

맞춤 매개변수	이벤트 수	총 사용자
(총) 1개 항목	1	1
grid	1	1

× 새 맞춤 측정기준

저장

⚠️ 고유 값이 많은 측정기준을 등록하면 보고서에 부정적인 영향을 미칠 수 있습니다. 맞춤 측정기준 설정에 대한 권장사항을 따라주세요.

권장사항에 대해 자세히 알아보기

측정기준 이름 ②

mode

범위 ②

이벤트

설명 ②

이벤트 매개변수 ②

mode

특정 조건에 맞는 사용자를  
그룹으로 관리하기

# 특정 조건에 맞는 사용자를 그룹으로 관리하기

애널리틱스 > Audiences를 열고, **새 잠재고객** 버튼을 클릭합니다.

 Analytics | Audiences

[Google 애널리틱스에서 더보기](#)



지난 28일 7월 6일~2022년 8월 2일 ▾

Compare: 6월 8일~2022년 7월 5일



새 잠재고객

잠재고객 이름

설명

사용자 ⓘ

변동률(%)

생성일 ↓

All Users

All users

2

-

2022. 7. 16.

Purchasers

Users who have made a purchase

사용자 10명 미만

-


2022. 7. 16.

# 특정 조건에 맞는 사용자를 그룹으로 관리하기

맞춤 잠재고객 만들기 버튼을 클릭합니다.

× 새 잠재고객 만들기

처음부터 새로 시작


 맞춤 잠재고객 만들기


추천 잠재고객


고려할 만한 추가 잠재고객 추천

일반


템플릿


 추천 잠재고객

 최근에 활동한 사용자  
최근에 활동한 사용자

 비구매자  
구매하지 않은 사용자

 구매자  
구매한 적이 있는 사용자

 최근 7일 동안 사라진 사용자  
한때 활동이 있었지만 최근 7일 동안 활동이 없는 사용자입니다.

 최근 7일 동안 사라진 구매자  
한때 활동이 있었지만 최근 7일 동안 활동이 없는 구매자입니다.

# 특정 조건에 맞는 사용자를 그룹으로 관리하기

최근 30일 이내에 5회 이상 노트를 생성한 사용자를 포함하는 조건을 입력합니다.

← Active users

취소

저장

설명 추가

다음 기준을 충족하는 사용자 를 포함합니다,

add\_note

이벤트 수 > 5

OR

AND

+ 조건 그룹 추가

+ 시퀀스 추가

+ 제외할 그룹 추가

포함 기간

☒ 30 days

☐ 최대 한도로 설정

잠재고객 트리거

+ 새로 만들기

요약

이 잠재고객과 일치하는 사용자 수

0

총 사용자 중 0%

포함 0

제외 -

# REST API 소개

# REST란?

- Representational State Transfer API의 약자로, 인터넷을 통해 서버와 클라이언트와 정보를 교환하기 위해 사용하는 인터페이스입니다.
- 구성
  - 자원 (리소스): 접근할 데이터의 위치는 URI(Uniform Resource Identifier)로 표현합니다.
  - Verb: 요청하는 작업의 종류를 설정할 수 있습니다.
    - POST: 데이터를 생성합니다.
    - GET: 데이터를 요청합니다.
    - PUT: 데이터를 수정합니다.
    - DELETE: 데이터를 삭제합니다.

# API란?

- Application Programming Interface의 약자로, 다른 소프트웨어 시스템과 통신하기 위해 따라야 하는 규칙을 정의합니다.
- 클라이언트(데이터를 요청하는 쪽)와 리소스를 제공하는 서버 사이를 이어주는 역할을 합니다.





**BACKEND**



**FRONTEND**



**API's**

# REST API 예시

## Method

GET

## URL (URI)

`https://api.unsplash.com/search/photos?query=cat&page=1`

## Request Header

Accept-Version

v1

Authorization

Client-ID abcdef12345

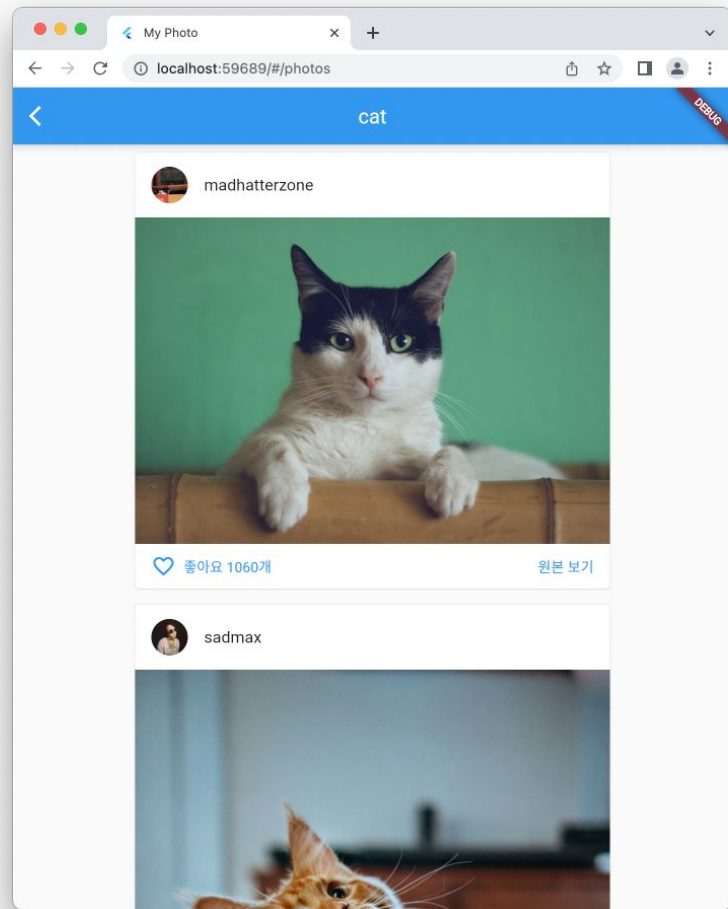
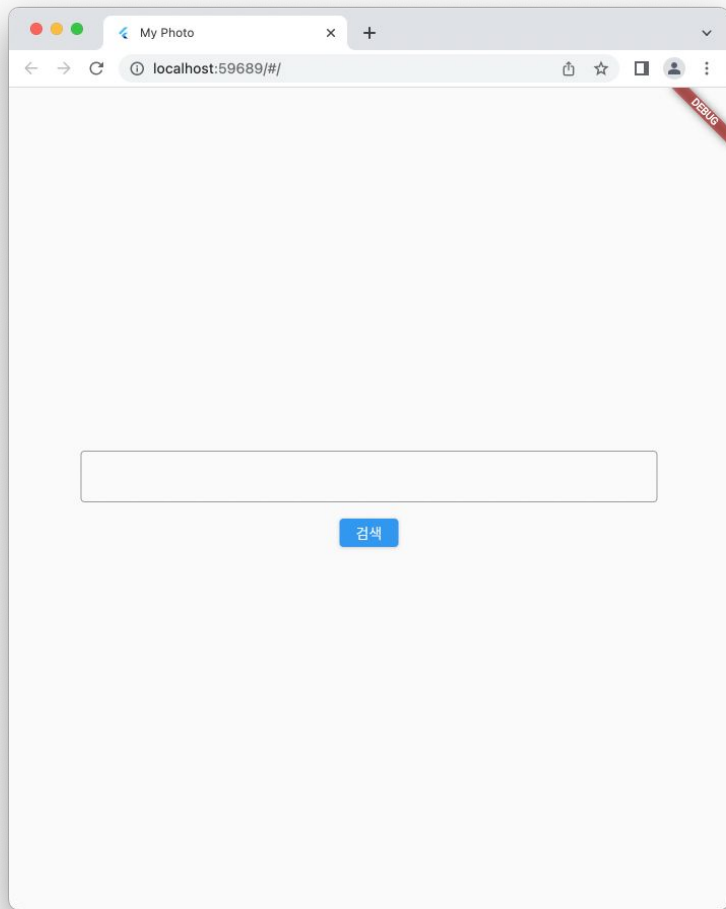
# REST API 응답 예시

Code: 200 OK

Body:

```
{
  "total": 133,
  "total_pages": 7,
  "results": [
    {
      "id": "e0LpJytrbsQ",
      "created_at": "2014-11-18T14:35:36-05:00",
      "width": 4000,
      "height": 3000,
      "color": "#A7A2A1",
      "blur_hash": "LaLXMa9Fx[D%~q%MtQM|kDRjtRIU",
      "likes": 286,
      "liked_by_user": false,
      "description": "A man drinking a coffee.",
      "user": { ... },
      "current_user_collections": [],
      "urls": { ... },
      "links": { ... }
    },
    // more photos ...
  ]
}
```

# 사진 검색 앱 프로젝트 만들기



- Project name: [my\\_photo](#)
- Organization: [edu.skku.sco](#).[\[본인 닉네임 혹은 ID\]](#)
  - 예) 닉네임이 androidhuman인 경우: [edu.skku.sco.androidhuman](#)
- Platforms: [Android](#), [iOS](#), [Web](#)

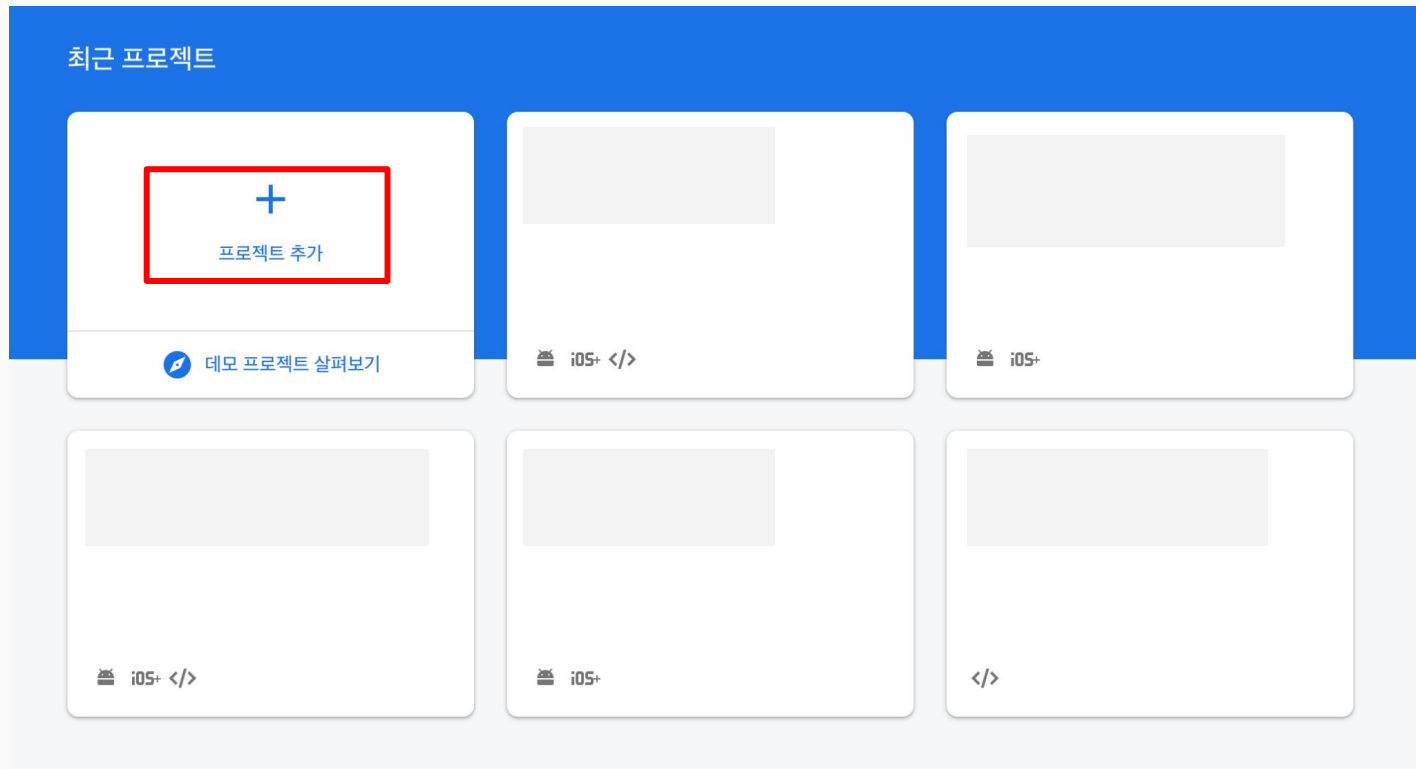
- pubspec.yaml 내 주석 삭제
- main.dart 파일 내 불필요한 코드 삭제

# 사진 검색 앱에 Firebase 연결하기



# Firebase 프로젝트 생성하기

[console.firebase.google.com](https://console.firebase.google.com)에 접속한 후, 프로젝트 추가 버튼을 누릅니다.



# Firebase 프로젝트 생성하기

프로젝트 이름을 지정합니다. (예: androidhuman-my-photo)

× 프로젝트 만들기(1/3단계)

프로젝트 이름을 지정하여 시작하  
기<sup>②</sup>

프로젝트 이름 입력

my-awesome-project-id

계속



# Firebase 프로젝트 생성하기



Google 애널리틱스를 사용 설정합니다.



## Firebase 프로젝트를 위한 Google 애널리틱스



무제한 무료 분석 솔루션인 Google 애널리틱스를 사용하면 Firebase Crashlytics, 클라우드 메시징, 인앱 메시징, 원격 구성, A/B 테스트, Cloud Functions에서 타겟팅, 보고 등을 이용할 수 있습니다.



Google 애널리틱스를 통해 다음 기능을 이용할 수 있습니다.

 A/B 테스트 

 장애가 발생하지 않은 사용자 

 Firebase 제품 전반에서 사용자 세분화 및 타  
겟팅 

 이벤트 기반 Cloud Functions 트리거 

 제한 없는 무료 보고 

☒ 이 프로젝트에서 Google 애널리틱스 사용 설정  
권장

[이전](#)


[계속](#)

# Firebase 프로젝트 생성하기

Google 애널리틱스 계정을 선택하거나 새 계정을 생성한 후, 프로젝트 만들기 버튼을 누릅니다.

## Google 애널리틱스 구성

Google 애널리틱스 계정 선택 또는 만들기 ?



새 계정 만들기

Google 애널리틱스 속성이 생성되고 Firebase 프로젝트에 연결됩니다. 이 연결을 통해 제품  
base로 내보낸 데이터에는 Firebase 서비스 약관이 적용되지만 Google 애널리  
이 적용됩니다. [자세히 알아보기](#)

프로젝트 만들기

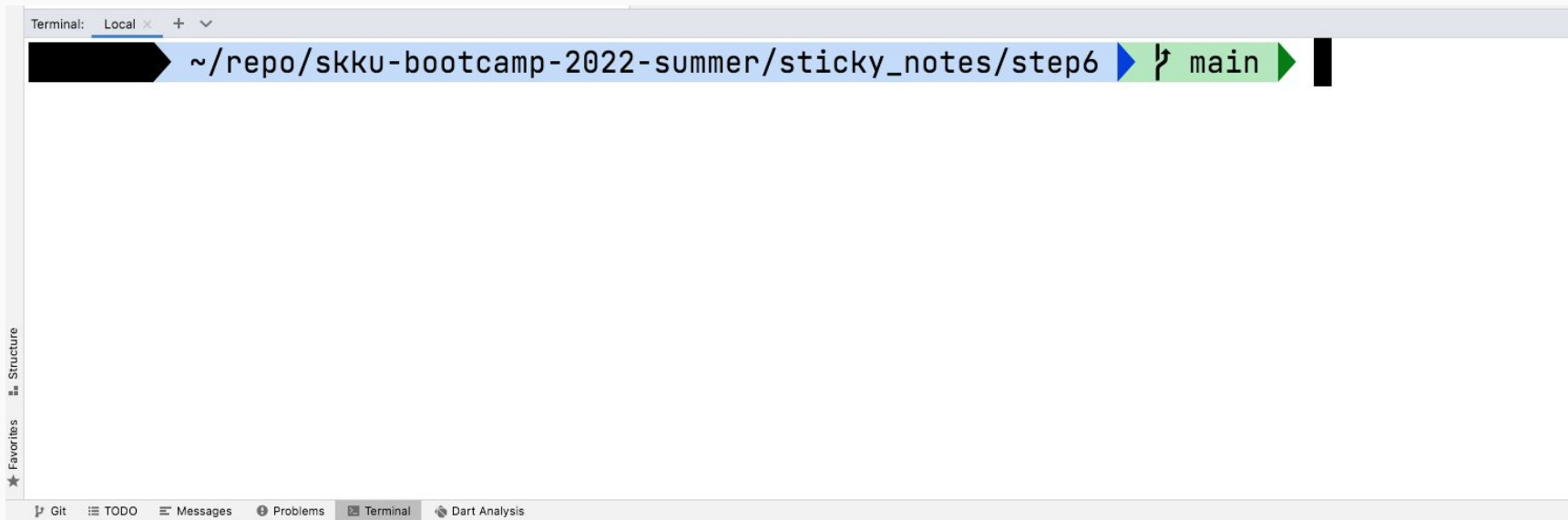
# 사진 검색 앱 프로젝트에 Firebase 설정하기

`pubspec.yaml` 의 `dependencies` 섹션에 `firebase_core`, `firebase_analytics` 패키지를 추가합니다.

```
dependencies:  
  firebase_analytics: ^9.3.0  
  firebase_core: ^1.20.0  
  flutter:  
    sdk: flutter
```

# 노트 앱 프로젝트에 Firebase 설정하기

안드로이드 스튜디오 하단의 Terminal 탭을 클릭하여 터미널을 실행합니다.



# 노트 앱 프로젝트에 Firebase 설정하기

다시 터미널 탭을 열고, **flutterfire configure** 명령어를 실행하여 파이어베이스 프로젝트를 연결을 시작합니다. 표시되는 프로젝트 중 연결할 프로젝트를 선택합니다.

```
i Found 9 Firebase projects.  
? Select a Firebase project to configure your Flutter application with ›  
  androidhuman-blog (androidhuman)  
  androidhuman-cubewhere (Cubewhere)  
  androidhuman-gc (androidhuman-gc)  
› androidhuman-my-photo (androidhuman-my-photo)  
  androidhuman-mycloud (myCloud)  
  androidhuman-myplaces (myPlaces)  
  androidhuman-newtown3 (Newtown 3)  
  androidhuman-pb (Pregnancy benefits)  
  androidhuman-sticky-notes (androidhuman-sticky-notes)  
  tesla-price-calculator (Tesla Price Calculator)  
  <create a new project>
```

# 노트 앱 프로젝트에 Firebase 설정하기

연결할 플랫폼을 선택합니다. 안드로이드와 iOS를 선택한 후 다음 단계로 진행합니다.

? Which platforms should your configuration support (use arrow keys & space to `select`)? ›

✓ android

✓ ios

macos

✓ web



# 노트 앱 프로젝트에 Firebase 설정하기

Flutter 프로젝트에 있는 안드로이드/iOS 앱 설정을 기반으로 Firebase 프로젝트에 앱이 추가되고, 이 설정 정보가 담긴 파일 (**firebase\_options.dart**)이 프로젝트에 추가됩니다.

```
i Firebase android app com.androidhuman.my_photo registered.  
i Firebase ios app com.androidhuman.myPhoto registered.  
i Firebase web app my_photo (web) registered.
```

Firebase configuration file lib/firebase\_options.dart generated successfully with the following Firebase apps:

Platform	Firebase App Id
web	1:619610477876:web:b3fab5d2c8cacc5f0b0161
android	1:619610477876:android:bd380c77d71253350b0161
ios	1:619610477876:ios:c199ffa6aef79b770b0161

Learn more about using this file and next steps from the documentation:

> <https://firebase.google.com/docs/flutter/setup>

# 노트 앱 프로젝트에 Firebase 설정하기

Firebase 서비스를 사용하려면 앱 시작 시점에 초기화가 필요합니다.

**Firestore.initializeApp()**을 호출하여 초기화를 해 줍니다.

```
void main() async {  
  WidgetsFlutterBinding.ensureInitialized();  
  await Firestore.initializeApp(options: DefaultFirestoreOptions.currentPlatform);  
  runApp(MyApp());  
}
```

main.dart

# 안드로이드 프로젝트 수정하기

파이어베이스 SDK를 사용하려면 앱이 지원하는 최소 안드로이드 버전을 조정해야 합니다.

**android/app/build.gradle** 파일을 연 후, **minSdkVersion**을 21로 변경합니다.

```
android {  
    ...  
    defaultConfig {  
        ...  
        minSdkVersion 21  
        targetSdkVersion flutter.targetSdkVersion  
        versionCode flutterVersionCode.toInteger()  
        versionName flutterVersionName  
    }  
}
```

# 사진 검색 앱에서 사용할 앱 이벤트 작성하기

## 사진 검색 앱에서 사용할 앱 이벤트 작성하기

1. `service` 폴더 아래에 `analytics_service.dart` 파일을 추가합니다.
2. 앱 이벤트를 담당하는 `AnalyticsService`를 구현합니다.

# 사진 검색 앱에서 사용할 앱 이벤트 작성하기

- 추가할 앱 이벤트
  - **favorite**: 사용자가 사진에 좋아요를 눌렀을 때 기록합니다.
    - id: 사진의 ID를 기록합니다.
  - **search**: 사용자가 사진을 검색했을 때 기록합니다.
    - search\_term: 검색어를 기록합니다.
  - **view\_original\_photo**: 노트를 삭제했을 때 기록합니다.
    - id: 사진의 ID를 기록합니다.

# 사진 검색 앱에서 사용할 앱 이벤트 작성하기

```
class AnalyticsService {  
  void favorite(String photoId) {  
    FirebaseAnalytics.instance.logEvent(  
      name: 'favorite_photo',  
      parameters: {  
        'id': photoId,  
      },  
    );  
  }  
  
  void search(String searchTerm) {  
    FirebaseAnalytics.instance.logSearch(searchTerm: searchTerm);  
  }  
  
  void viewOriginal(String photoId) {  
    FirebaseAnalytics.instance.logEvent(  
      name: 'view_original_photo',  
      parameters: {  
        'id': photoId,  
      },  
    );  
  }  
}
```

# 사진 검색 서비스 개발자 계정 신청하기



## 사진 검색 서비스 개발자 계정 신청하기

- 외부 API를 사용하려면 해당 서비스에 개발자 계정을 생성하고, 연동하려는 애플리케이션에서 사용할 수 있는 키를 발급받아야 합니다.
- 발급받은 키를 기반으로 한도 제한이 적용되므로, 외부에 노출되지 않도록 관리해야 합니다.

## 사진 검색 서비스 개발자 계정 신청하기

[Unsplash 개발자 사이트](#)에 접속한 후, Register as developer 버튼을 눌러 계정을 생성합니다.



[Documentation](#)
[Dataset](#)
[API Status](#)
[API Changelog](#)
[Login](#)

**Register as a developer**

# The most powerful photo engine in the world.

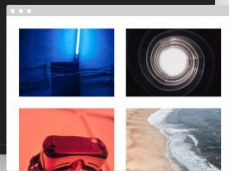
Welcome to the Official Unsplash API. Create with the largest open collection of high-quality photos.

[Register as a developer](#)

[View the documentation](#)

Interested in using the Unsplash API in a high-volume application? Get in touch with our Partnerships team at [partnerships@unsplash.com](mailto:partnerships@unsplash.com)

```
curl https://api.unsplash.com/users/samuelzeller/photos
{
  "id": "fwJ9E9amob6K",
  "created_at": "2018-04-11T16:02:15-04:00",
  "updated_at": "2018-04-11T16:02:17-04:00",
  "width": 4811,
  "height": 3207,
  "color": "#A3A3AD",
  "likes": 726,
  "user": {
    "id": "gnwb2chnB8Q",
    "updated_at": "2018-04-11T16:02:15-04:00",
    "username": "SamuelZeller",
    "name": "Samuel Zeller",
    "first_name": "Samuel",
    "last_name": "Zeller",
    "portfolio_url": "https://samuelzeller.ch",
    "location": "Geneva, Switzerland",
    "profile_image": {
      "small": "https://images.unsplash.com/profile
```



## 1.5B

requests/month

## 4.1M

photos

**303k**

photographers

# 사진 검색 서비스 개발자 계정 신청하기

Your Apps 페이지로 이동한 다음, New Application을 눌러 새 앱을 등록합니다.

## Welcome, Taeho

We're excited to see what you'll build. Don't forget to apply for production once your application is ready.

Your applications

### Cat of the day

Status: Demo



New Application

# 사진 검색 서비스 개발자 계정 신청하기

API 사용 약관을 수락합니다.

## API Use and Guidelines

Last updated March 2, 2020

- ☒ The API is to be used for non-automated, high-quality, and authentic experiences.  
[More info & examples ↗](#)
- ☒ You cannot replicate the core user experience of Unsplash (unofficial clients, wallpaper applications, etc.).  
[More info & examples ↗](#)
- ☒ Your Access Key and Secret Key must remain confidential.
- ☒ Do not abuse the API. Too many requests too quickly will get your access turned off.

These Unsplash API Terms of Use (these "**API Terms**") comprise a legal agreement between Unsplash, Inc. ("**Unsplash**," "**us**," "**we**," or "**our**") and you, and describe the rules developers must follow when accessing or using the Unsplash application programming interfaces (the "**APIs**") and related documentation made available by Unsplash through the Unsplash developer website currently located at <https://unsplash.com/developers> ("**Developer Site**"). Any application you integrate with the APIs is a "**Developer App**."

BY CLICKING "I AGREE" OR IMPLEMENTING OR OTHERWISE USING ANY UNSPLASH API, YOU ACKNOWLEDGE THAT YOU HAVE READ, UNDERSTAND AND AGREE TO BE BOUND BY THESE API TERMS.

### 1. Your Relationship with Unsplash.


- **Legal Agreement.** Your use of the APIs is subject to these API Terms. These API Terms, together with any other documents or terms incorporated into these API Terms or provided to you by

Decline

Accept terms

# 사진 검색 서비스 개발자 계정 신청하기

애플리케이션 이름과 간단한 설명을 입력합니다.

 **Application information** ×

Application name

Description

Describe your application

Cancel

Create application

# 사진 검색 서비스 개발자 계정 신청하기

생성한 애플리케이션을 목록에서 확인할 수 있습니다.

## Welcome, Taeho

We're excited to see what you'll build. Don't forget to apply for production once your application is ready.

Your applications

**Cat of the day**

Status: Demo



New Application

# 사진 검색 서비스 개발자 계정 신청하기

애플리케이션 페이지 내 Keys 섹션에서 Access Key를 확인할 수 있습니다.

## Keys

Access Key



Secret key



Note: both your `Access Key` and `Secret Key` must remain confidential.

# 사진 검색 API를 위한 코드 작성하기



## 사진 검색 API를 위한 코드 작성하기

1. pubspec.yaml 내 dependencies 섹션에 **dio** 패키지를 추가합니다.
2. 사진 검색 서비스 API의 응답을 표현할 클래스를 담아줄 **model** 폴더를 생성합니다.
3. **model** 폴더 아래에 **photo.dart** 와 **search\_result.dart**파일을 추가합니다.
4. 사진 검색 서비스 API를 편리하게 사용할 수 있게 해주는 **PhotoService** 클래스를 구현합니다.

# 사진 검색 API를 위한 코드 작성하기

`pubspec.yaml` 의 `dependencies` 섹션에 **dio** 패키지를 추가합니다.

```
dependencies:  
  dio: ^4.0.6  
  firebase_analytics: ^9.3.0  
  flrebae_core: ^1.20.0  
  flutter:  
    sdk: flutter
```

# 사진 검색 API를 위한 코드 작성하기

[사진 검색 API](#)에서 반환하는 결과 포맷을 확인합니다.

```
{
  "total": 133,
  "total_pages": 7,
  "results": [
    {
      "id": "e0LpJytrbsQ",
      "created_at": "2014-11-18T14:35:36-05:00",
      "width": 4000,
      "height": 3000,
      "color": "#A7A2A1",
      "blur_hash": "LaLXMa9Fx[D%~q%MtQM|kDRjtRIU",
      "likes": 286,
      "liked_by_user": false,
      "description": "A man drinking a coffee.",
      "user": { ... },
      "current_user_collections": [],
      "urls": { ... },
      "links": { ... }
    },
    // more photos ...
  ]
}
```

# 사진 검색 API를 위한 코드 작성하기

결과 포맷을 담아줄 클래스를 정의합니다.

```
{
  "total": 133,
  "total_pages": 7,
  "results": [
    {
      "id": "e0LpJytrbsQ",
      "created_at": "2014-11-18T14:35:36-05:00",
      "width": 4000,
      "height": 3000,
      "color": "#A7A2A1",
      "blur_hash": "LaLXMa9Fx[D%~q%MtQM|kDRjtRIU",
      "likes": 286,
      "liked_by_user": false,
      "description": "A man drinking a coffee.",
      "user": { ... },
      "current_user_collections": [],
      "urls": { ... },
      "links": { ... }
    },
    // more photos ...
  ]
}
```

Photo

SearchResult

# 사진 검색 API를 위한 코드 작성하기

```
class Photo {  
  final String id;  
  
  final int width;  
  
  final int height;  
  
  final String previewUrl;  
  
  final String downloadUrl;  
  
  final int likes;  
  
  final String username;  
  
  final String userProfileImageUrl;  
  
  Photo.from(dynamic source)  
    : id = source['id'],  
      width = source['width'],  
      height = source['height'],  
      previewUrl = source['urls']['regular'],  
      downloadUrl = source['urls']['raw'],  
      likes = source['likes'],  
      username = source['user']['username'],  
      userProfileImageUrl = source['user']['profile_image']['medium'];  
  
  const Photo(  
    this.id,  
    this.width,  
    this.height,  
    this.previewUrl,  
    this.downloadUrl,  
    this.likes,  
    this.username,  
    this.userProfileImageUrl,  
  );  
}
```

# 사진 검색 API를 위한 코드 작성하기

```
class SearchResult {  
    final int total;  
  
    final int totalPages;  
  
    final List<Photo> results;  
  
    SearchResult.from(dynamic source)  
        : total = source['total'],  
          totalPages = source['total_pages'],  
          results =  
              List.from(source['results']).map((e) => Photo.from(e)).toList();  
  
    const SearchResult(this.total, this.totalPages, this.results);  
}
```

search\_result.dart

# 사진 검색 API를 위한 코드 작성하기

사진 검색 API를 호출하기 위해 [기본으로 필요한 정보](#)를 확인합니다.

- Host: <https://api.unsplash.com>
- Headers:
  - Accept-Version: v1
  - Authorization: Client-ID [ACCESS\_KEY]

# 사진 검색 API를 위한 코드 작성하기

[사진 검색 결과를 반환하는 API](#)에 필요한 정보를 확인합니다.

- Method: GET
- Path: /search/photos
- Query parameters:
  - query: [검색어]
  - page: [Page number to retrieve; default: 1]



# 사진 검색 API를 위한 코드 작성하기

```
const _accessKey = '8ml-5NmWCUwf0IHYwgF0Si88rSydiJ7jaboFbhjtkV4';

class PhotoService {
  Future<SearchResult> search(String searchTerm, int page) async {
    try {
      final response = await Dio().get(
        'https://api.unsplash.com/search/photos',
        queryParameters: {
          'query': searchTerm,
          'page': page,
        },
        options: Options(
          headers: {
            'Accept-Version': 'v1',
            'Authorization': 'Client-ID $_accessKey',
          },
        ),
      );
      return SearchResult.from(response.data);
    } catch (e) {
      return Future.error(e);
    }
  }
}
```

photo\_service.dart

완성된 코드

[https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/my\\_photo/step1](https://github.com/kunny/skku-bootcamp-2022-summer/tree/main/my_photo/step1)