

# Do it! 플러터 앱 프로그래밍

7장 네트워크를 이용한 통신하기

# 목차

- 07-1 카카오 API를 이용해 책 정보 받아오기
- 07-2 이미지 파일 내려받기

# 07-1 카카오 API를 이용해 책 정보 받아오기

패키지 이용하기

pub.dev 에 들어가서 http 검색하기

패키지 이름과 인기도 확인 후 클릭하기

패키지 찾은후 pubspec.yaml 파일에 입력하기

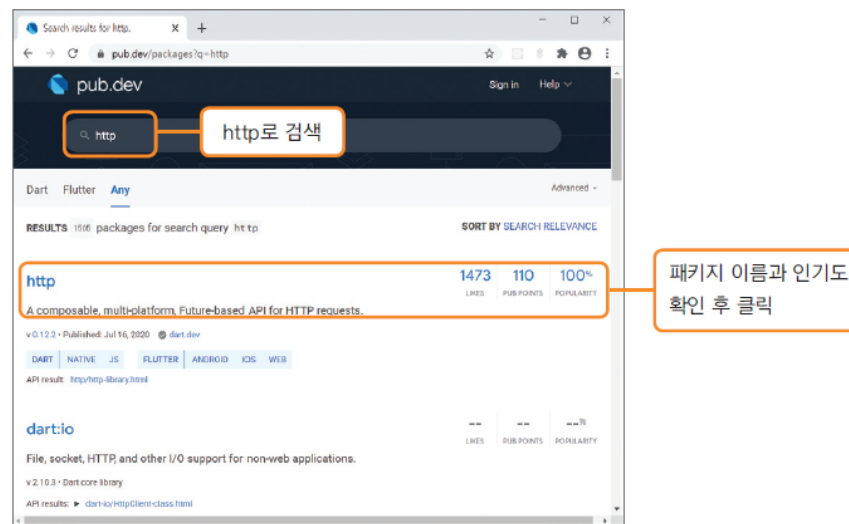


그림 7-1 http로 검색

• pubspec.yaml

```
(...생략...)
dependencies:
  flutter:
    sdk: flutter
  cupertino_icons: ^1.0.2
  http: ^0.13.3
(...생략...)
```

# 07-1 카카오 API를 이용해 책 정보 받아오기

카카오 API 이용하기

Kakao developers 사이트에 접속하기

로그인한후에 앱등록하기

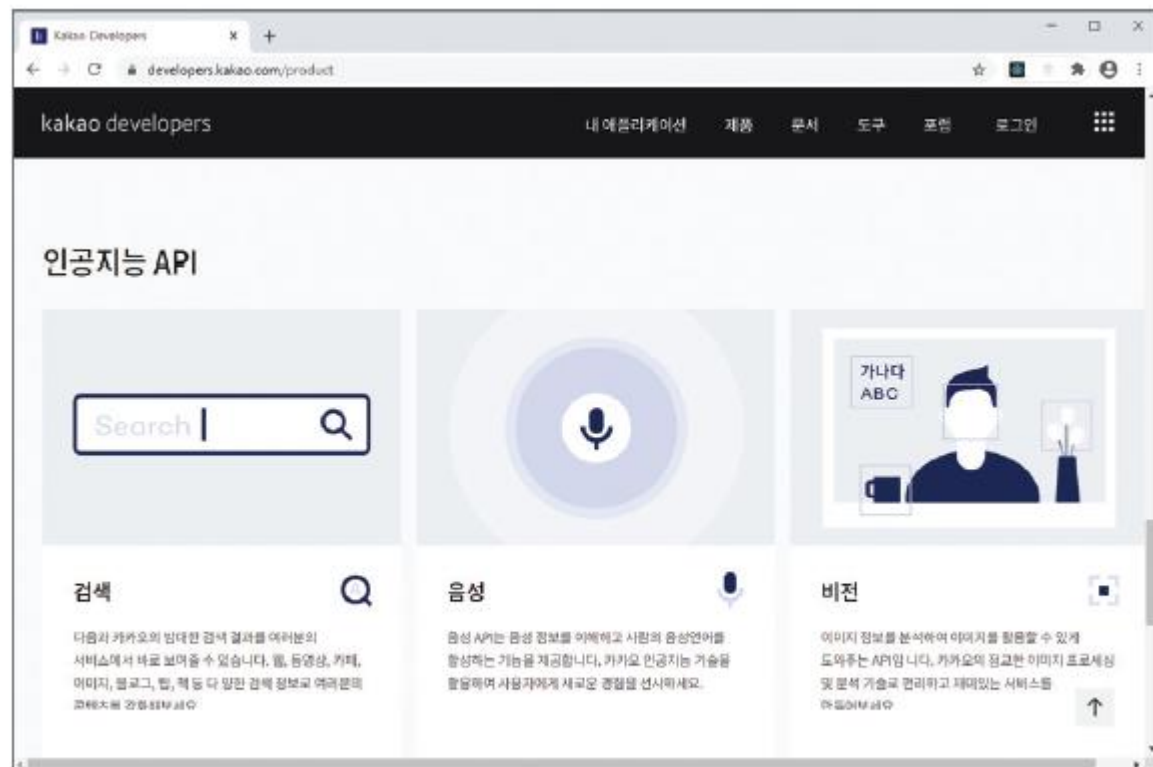


그림 7-3 카카오가 제공하는 다양한 API(출처: developers.kakao.com/)

# 07-1 카카오 API를 이용해 책 정보 받아오기

카카오 API 이용하기

전체 애플리케이션에서 애플리케이션 추가  
하기를 클릭한 후

앱 정보를 입력하기

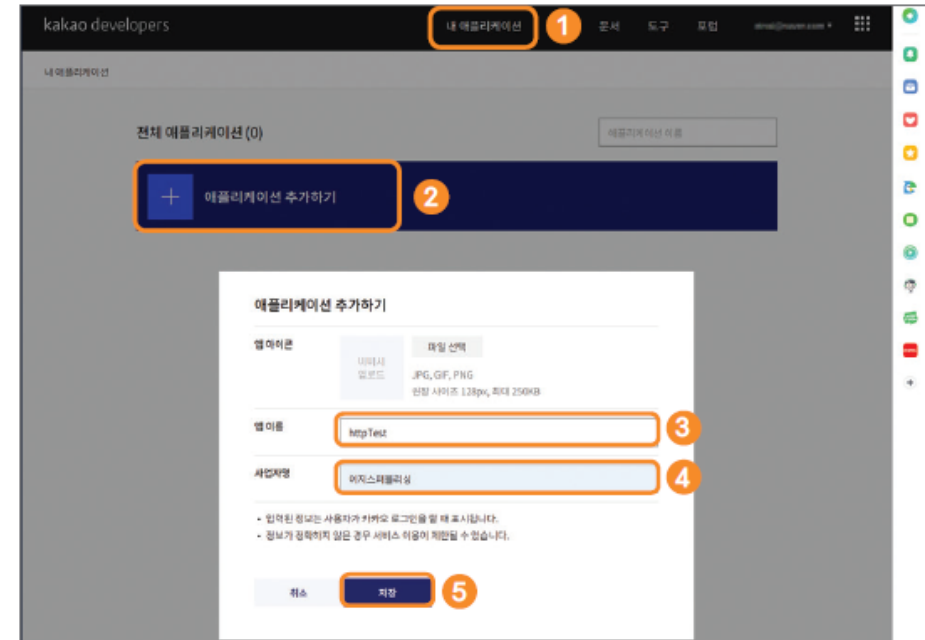


그림 7-4 애플리케이션 추가하기

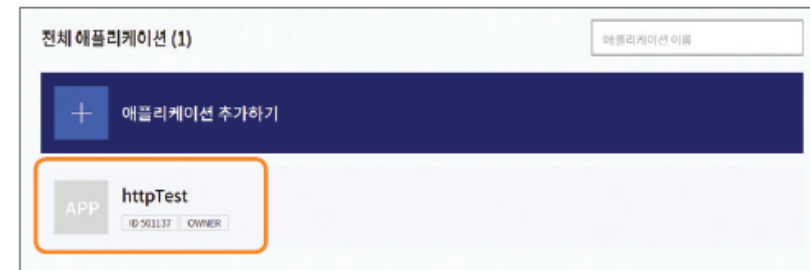


그림 7-5 추가된 애플리케이션

# 07-1 카카오 API를 이용해 책 정보 받아오기

## 카카오 API 이용하기

발급받은 API 키를 이용하여 카카오 개발 가이드 보면서 http 통신 따라하기

- 카카오 개발 가이드 책 검색:

<https://developers.kakao.com/docs/latest/ko/daum-search/dev-guide#search-book>

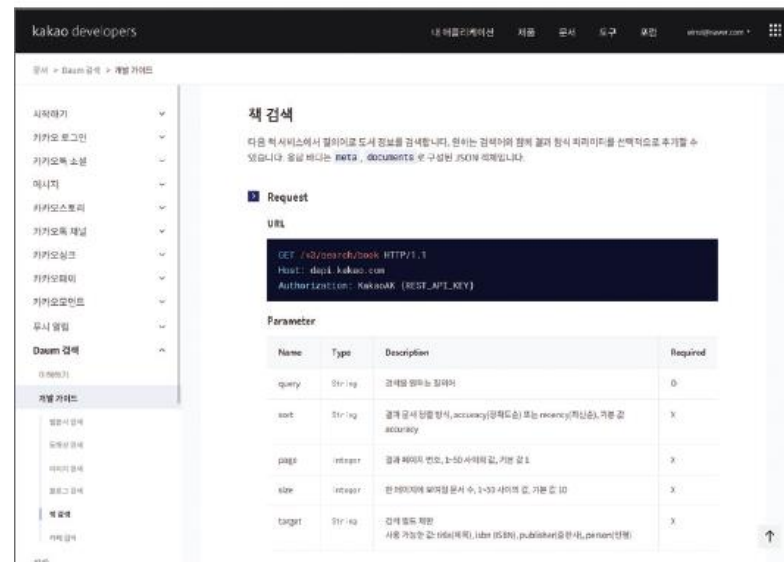


그림 7-7 카카오 책 검색 개발 가이드

# 07-1 카카오 API를 이용해 책 정보 받아오기

## 카카오 API 이용하기

Main.dart 파일에서 import를 이용하여 패키지  
지에 추가한 http 패키지 추가하기

as : http.dart 파일을 http 라는 변수를 이용  
하여 사용하겠다는 의미

• lib/main.dart

```
import 'package:flutter/material.dart';
```

```
import 'package:http/http.dart' as http;
```

http 패키지 불러오기

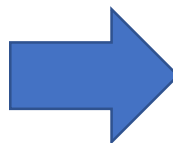
# 07-1 카카오 API를 이용해 책 정보 받아오기

카카오 API 이용하기

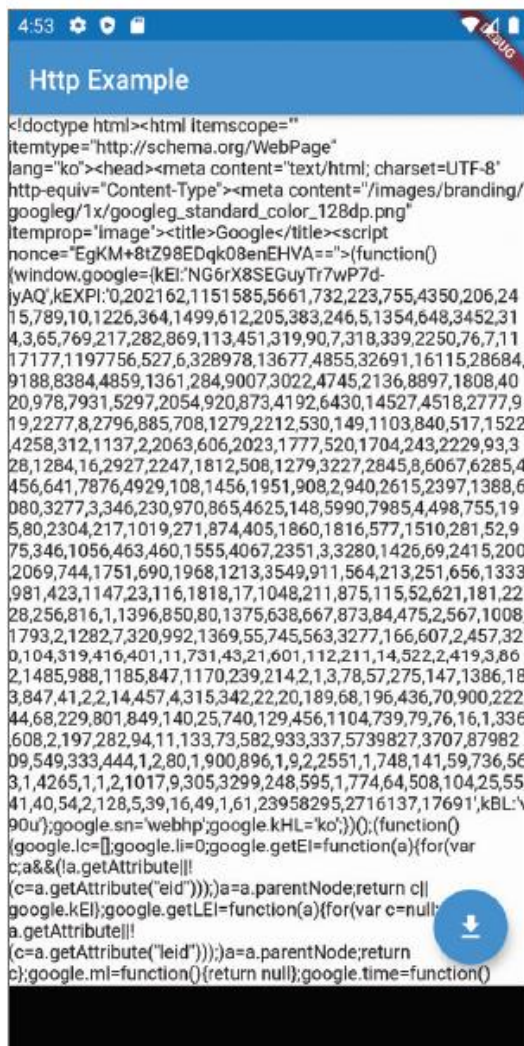
http.get 을이용하여 데이터 통신 Test 하기

```
floatingActionButton: FloatingActionButton(  
  onPressed: () async {  
    var url = 'http://www.google.com';  
    var response = await http.get(url);  
    setState(() {  
      result = response.body;  
    });  
  },  
  child: Icon(Icons.file_download),  
)
```

GET 방식으로 URL에 접속하는 코드



▶ 실행 결과





# 07-1 카카오 API를 이용해 책 정보 받아오기

카카오 API 이용하기

발급받은 API 키를 이용하여 통신 구현하기

```
Future<String> getJSONData() async {  
    var url = 'https://dapi.kakao.com/v3/search/book?target=title&query=doit';  
    var response = await http.get(Uri.parse(url),  
    headers: {"Authorization": "KakaoAK ##REST API 키 넣기##"});  
    print(response.body);  
    return "Successfull";  
}
```

<https://dapi.kakao.com/v3/search/book?target=title&query=doit>

①

②

③

- ① 요청할 도메인입니다. 카카오에서 책을 검색하는 API를 나타냅니다.
- ② 도메인에 요청할 파라미터입니다. target 파라미터에 title을 전달합니다.
- ③ query 파라미터에 검색어 doit을 전달합니다.

# 07-1 카카오 API를 이용해 책 정보 받아오기

카카오 API 이용하기

받은 데이터 처리하기

Response.body를 json 으로 변환후 List로 출력하기

```
Future<String> getJSONData() async {  
    (...생략...)  
    print(response.body);  
    setState(() {  
        var dataConvertedToJSON = json.decode(response.body);  
        List result = dataConvertedToJSON['documents'];  
        data.addAll(result);  
    });  
    return response.body;  
}
```

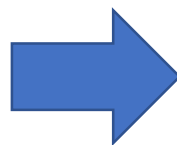
# 07-1 카카오 API를 이용해 책 정보 받아오기

카카오 API 이용하기

받은 데이터 처리하기

Response.body를 json 으로 변환후 List로 출력하기

```
Future<String> getJSONData() async {  
    (...생략...)  
    print(response.body);  
    setState(() {  
        var dataConvertedToJSON = json.decode(response.body);  
        List result = dataConvertedToJSON['documents'];  
        data.addAll(result);  
    });  
    return response.body;  
}
```



▶ 실행 결과



## 07-2 이미지파일 내려받기

실습하기

Pub.dev 에서 dio 와 path\_provider 검색하기

Pubspec.yaml 에 패키지 추가하기

추가할 dart 파일에 import 를 이용한 패키지 추가하기

```
• pubspec.yaml

(...생략...)
dependencies:
  flutter:
    sdk: flutter

  cupertino_icons: ^1.0.2
  dio: ^4.0.0
  path_provider: ^2.0.2
(...생략...)
```

```
import 'package:dio/dio.dart';
import 'package:path_provider/path_provider.dart';
import 'dart:io';
```

## 07-2 이미지파일 내려받기

실습하기

downloadFile() 함수를 생성하고 dio 패키지를 이용하여 파일을 다운로드 받을수있도록 함수 구현하기

```
Future<void> downloadFile() async {
  Dio dio = Dio();
  try {
    var dir = await getApplicationDocumentsDirectory();
    await dio.download(imgUrl, '${dir.path}/myimage.jpg',
      onReceiveProgress: (rec, total) {
        print('Rec: $rec , Total: $total');
        file = '${dir.path}/myimage.jpg';
        setState(() {
          downloading = true;
          progressString = ((rec / total) * 100).toStringAsFixed(0) + '%';
        });
      });
  } catch (e) {
    print(e);
  }
  setState(() {
    downloading = false;
    progressString = 'Completed';
  });
  print('Download completed');
}
```

## 07-2 이미지파일 내려받기

실습하기

FutureBuilder 를 이용하여 함수를 호출하기  
전까지 구현할 위젯 구현하기

None : 데이터 없음

Waiting : 데이터 전달 받는 중

Active : stream 모드에서 동작하는 코드

Done : 모든 데이터 전송이 완료된 상태

```
FutureBuilder(  
  builder: (context, snapshot) {  
    switch  
    (snapshot.connectionState) {  
      case ConnectionState.none:  
        print('none');  
        return Text('데이터 없음');  
      case ConnectionState.waiting:  
        print('waiting');  
        return  
        CircularProgressIndicator();  
      case ConnectionState.active:  
        print('active');  
        return  
        CircularProgressIndicator();  
      case ConnectionState.done:  
        print('done');  
        if (snapshot.hasData) {  
          return snapshot.data;  
        }  
        print('end process');  
        return Text('데이터 없음');  
    }  
  },  
)
```

## 07-2 이미지파일 내려받기

실습하기

다운로드 버튼을 누르면

download 가 실행되어지고 완료되면 이미지가 출력되는 실행결과

▶ 실행 결과

