Do it! 플러터 앱 프로그래밍

12장 **운영체제별 네이티브와 통신하기**

목차

12-1 안드로이드 네이티브와 통신하기

플러터는 네이티브와 통신을 기본적으로 지원, 원하는 통신을 만들기 위해 네이버

MethodChannel 을 이용하여 데이터 통신

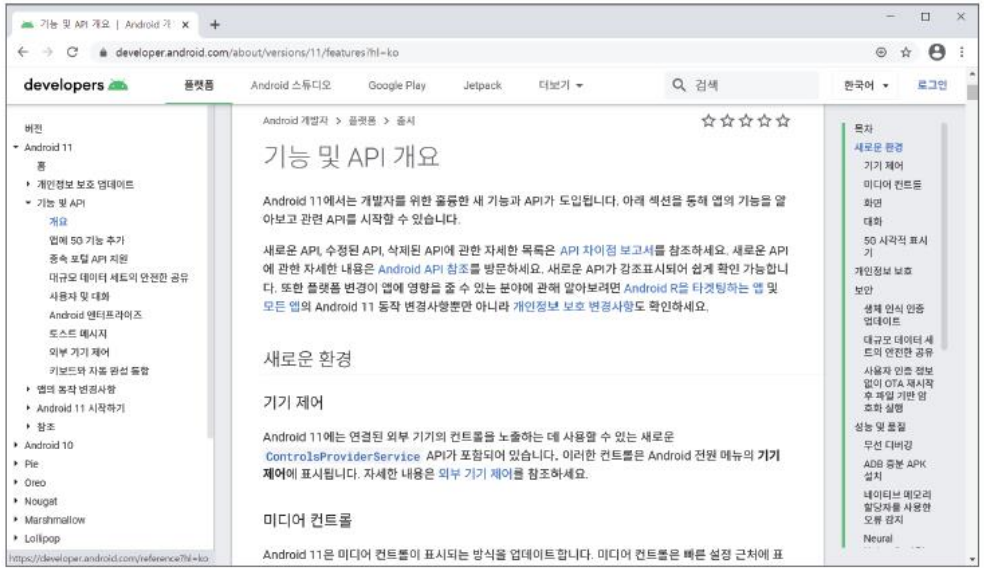
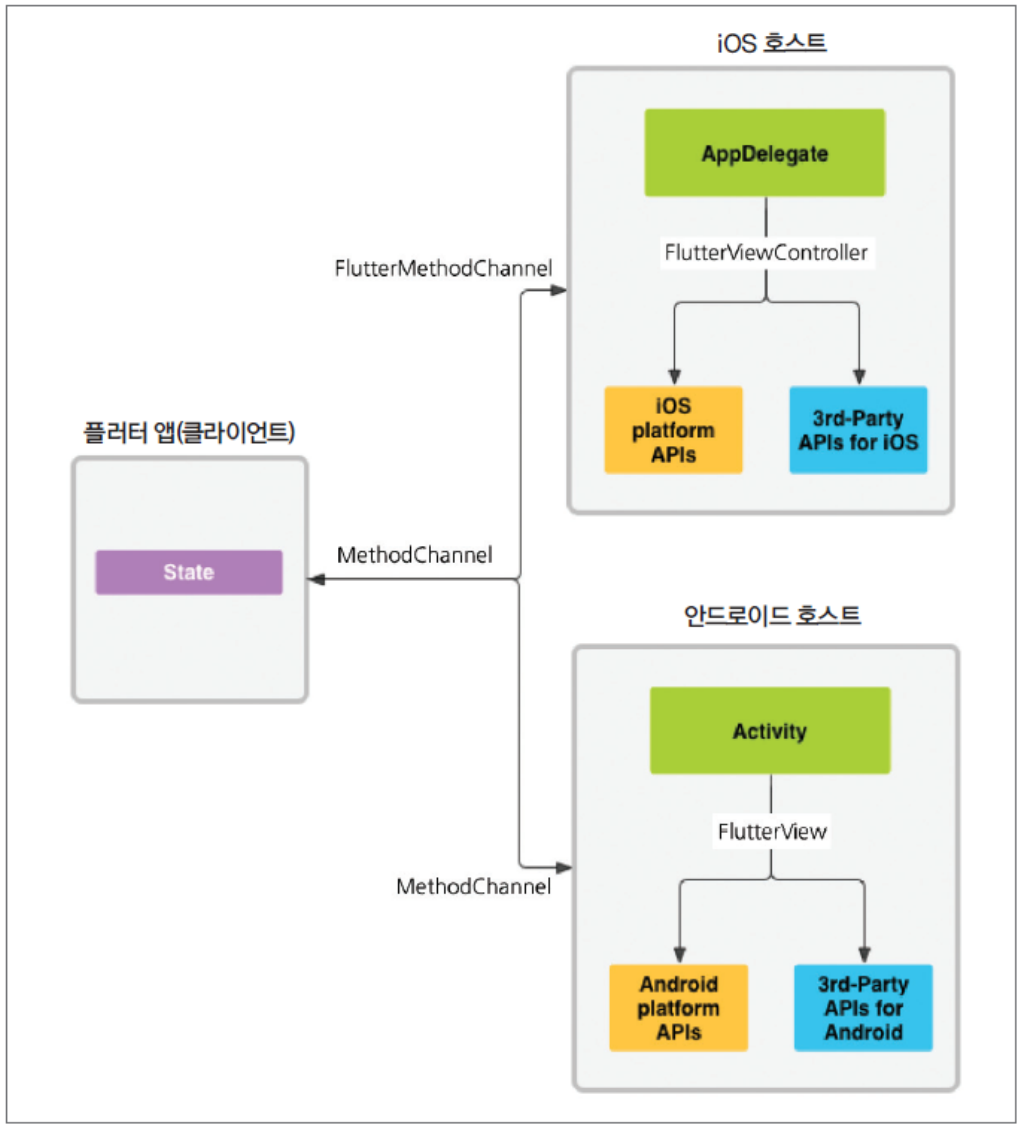


그림 12-1 안드로이드 API 소개(developer.android.com/about/versions/11/features?hl=ko)



12-1 안드로이드 네이티브와 통신하기

플러터는 보내는 데이터를

Android , iOS 에서 받아올때 데이터 타입으로하는게 중요

표 12-1 다트 자료형에 대응하는 플랫폼별 자료형

타입	다트	안드로이드	iOS
공백	null	null	nil
참, 거짓	bool	java.lang.Boolean	NSNumber numberWithBool
정수형	int	java.lang.Integer	NSNumber numberWithInt
		java.lnag.Long	NSNumber numberWithLong
부동소수점	double	java.lang.Double	NSNumber numberWithDouble
문자열	String	java.lang.String	NSString
바이트 배열	Uint8List	byte[]	FlutterStandardTypedData typedDataWithBytes
정수형 배열	Int32List	int[]	FlutterStandardTypedData typedDataWithInt32
	Int64List	long[]	FlutterStandardTypedData typedDataWithInt64
부동소수점 배열	Float64List	double[]	FlutterStandardTypedData typedDataWithFloat64
List형 자료구조	List	java.util.ArrayList	NSArray
Map형 자료구조	Map	java.util.HashMap	NSDictionary

12-1 안드로이드 네이티브와 통신하기

실습하기

_getDeviceInfo() 함수를 이용하여 네이티브의 데이터를 가져오는 함수를 만들어 주도록 합니다.

Platform.invokeMethod 함수를 이용하여 어떤 함수를 네이티브로 호출할지 정의합니다

• lib/main.dart

```
(...생략...)
Future<void> _getDeviceInfo() async {
  String deviceInfo;
  try {
    final String result = await platform.invokeMethod('getDeviceInfo');
    deviceInfo = 'Device info : $result';
  } on PlatformException catch (e) {
    deviceInfo = 'Failed to get Device info: '${e.message}'.';
  }
  setState(() {
    _deviceInfo = deviceInfo;
  });
}
(...생략...)
```

12-1 안드로이드 네이티브와 통신하기

실습하기

안드로이드 스튜디오 프로젝트를 열어서 MainActivity.kt 에 import 해주도록 합니다.

추가로 플러터와 통신한 코드를 이용하여 함수를 만들어 주도록 합니다

• MainActivity.kt(안드로이드 프로젝트)

```
package com.rollcake.native_example

import android.os.Build
import androidx.annotation.NonNull
import io.flutter.embedding.android.FlutterActivity
import io.flutter.embedding.engine.FlutterEngine
import io.flutter.plugin.common.MethodChannel

class MainActivity : FlutterActivity() {
  private val CHANNEL = "com.flutter.dev/info"

  override fun configureFlutterEngine(@NonNull flutterEngine: FlutterEngine) {
    super.configureFlutterEngine(flutterEngine)
    MethodChannel(flutterEngine.dartExecutor.binaryMessenger, CHANNEL)
        .setMethodCallHandler { call, result ->
          if (call.method == "getDeviceInfo") {
            val deviceInfo = getDeviceInfo()
            result.success(deviceInfo)
          }
        }
  }
}
```

12-1 안드로이드 네이티브와 통신하기

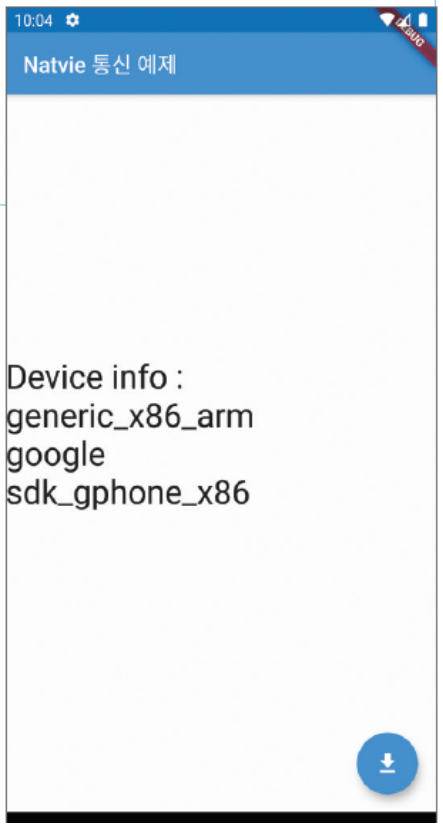
실습하기

getDeviceInfo() 함수를 이용하여
어떤 데이터를 전달할지 정의한 후 return을 하도록 합니다

• MainActivity.kt(안드로이드 프로젝트)

```
(...생략...)
private fun getDeviceInfo(): String {
  val sb = StringBuffer()
  sb.append(Build.DEVICE + "\n")
  sb.append(Build.BRAND + "\n")
  sb.append(Build.MODEL + "\n")
  return sb.toString()
}
```

▶ 실행 결과



12-1 안드로이드 네이티브와 통신하기

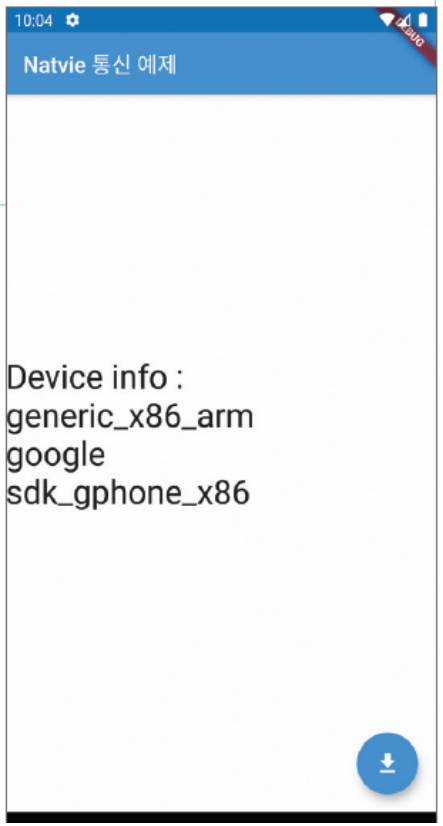
실습하기

getDeviceInfo() 함수를 이용하여
어떤 데이터를 전달할지 정의한 후 return을 하도록 합니다

• MainActivity.kt(안드로이드 프로젝트)

```
(...생략...)
private fun getDeviceInfo(): String {
  val sb = StringBuffer()
  sb.append(Build.DEVICE + "\n")
  sb.append(Build.BRAND + "\n")
  sb.append(Build.MODEL + "\n")
  return sb.toString()
}
```

▶ 실행 결과



12-2 안드로이드 네이티브와 데이터 주고받기

실습하기 – **네이티브 소스로 인코딩/디코딩 구현하기**

네이티브에 보낼 데이터를 platform.invokeMethod를 통해 처리하도록 합니다.

```
Future<void> _sendData(String text) async {
  final String result = await platform.invokeMethod('getEncrypto', text);
  print(result);
  setState(() {
    _changeText = result;
  });
}
}
```

네이티브 소스에 전달할 데이터

12-2 안드로이드 네이티브와 데이터 주고받기

실습하기 – **네이티브 소스로 인코딩/디코딩 구현하기**

네이티브에 보낼 데이터를 platform.invokeMethod를 통해 처리하도록 합니다.

안드로이드 스튜디오를 실행한 후

MainActivity.kt 에 새로운 channel을 정의하도록 합니다.

```
Future<void> _sendData(String text) async {
  final String result = await platform.invokeMethod('getEncrypto', text);
  print(result);
  setState(() {
    _changeText = result;
  });
 }
}
```

네이티브 소스에 전달할 데이터

• MainActivity.kt(안드로이드 프로젝트)

```
import android.util.Base64
(...생략...)
class MainActivity : FlutterActivity() {
  private val CHANNEL = "com.flutter.dev/info"
  private val CHANNEL2 = "com.flutter.dev/encrypto"
(...생략...)
```

Base64 인코딩을 사용하기 위한 라이브러리

플러터 프로젝트에 작성한 메서드 채널의 키값과 일치

12-2 안드로이드 네이티브와 데이터 주고받기

실습하기 – **네이티브 소스로 인코딩/디코딩 구현하기**

네이티브에서 Base64로 인코딩하는 글자를 넣도록 합니다

Base64는 다양한 통신환경에서 인코딩에 구애받지 않도록하기위하여 64개의 글자로 표현한 방식입니다.

• MainActivity.kt(안드로이드 프로젝트)

```
(...생략...)
MethodChannel(flutterEngine.dartExecutor.binaryMessenger,
    CHANNEL2).setMethodCallHandler { call, result ->
  if (call.method == "getEncrypto") {
    val data = call.arguments.toString().toByteArray();
    val changeText = Base64.encodeToString(data, Base64.DEFAULT)
    result.success(changeText)
  }
}
(...생략...)
```

플러터에서 보낸 데이터

데이터를 Base64로 인코딩하기

12-2 안드로이드 네이티브와 데이터 주고받기

실습하기 – **네이티브 소스로 인코딩/디코딩 구현하기**

이제 플러터에서 UI를 처리한 후 빌드하면

다음과 같이 변환된 데이터가 화면에 출력되는 것을 확인할 수 있습니다.

▶ 실행 결과