안드로이드 프로그래밍 – 첫번째 앱 개발

Android SDK 구성 요소

1. SDK Platform

- 특정 버전의 Android OS 에 대한 개발 환경을 제공
- 예를 들어, Android 13 을 위한 SDK Platform 을 설치하면 해당 버전의 API 와 시스템 이미지를 사용할 수 있어 그 버전에 맞는 앱을 개발하고 테스트할 수 있음
- 각 Android OS 버전(e.g., Android 12, 13, 14)마다 별도의 SDK Platform 이 존재

2. SDK Platform-tools

- Android 기기와 상호작용하는 데 사용되는 가장 핵심적인 명령줄 도구 모음
- 안드로이드 플랫폼 버전에 종속되지 않고, 모든 Android 버전에서 작동하며, 자주 업데이트됨
- 주요 도구
    - adb (Android Debug Bridge): 컴퓨터와 안드로이드 기기 또는 에뮬레이터를 연결하여 파일을 전송하거나, 앱을 설치하고, 로그를 확인하는 등의 작업을 수행
    - fastboot: 부트로더와 직접 통신하여 기기 펌웨어를 플래시(설치)하는 데 사용

3. SDK Build-tools

- 안드로이드 앱을 빌드(컴파일 및 패키징)하는 데 필요한 도구 모음
- 안드로이드 소스 코드를 실행 가능한 앱(.apk 또는 .aab 파일)으로 만드는 데 필요한 컴파일러, 패키징 도구, 서명 도구 등이 포함
    - aapt (Android Asset Packaging Tool): 앱의 리소스(리소스, 매니페스트 파일 등)를 컴파일하고 패키징
    - dx (Dalvik Executable): Java 바이트코드를 Dalvik(또는 ART) 바이트코드로 변환

4. SDK Tools

- 개발 환경을 관리하고 보조하는 도구들의 모음
- 주로 Android Studio 의 기능들을 지원하며, 더 이상 최신 버전의 Android Studio 에서는 'SDK Tools'라는 별도의 패키지로 설치되지 않음
    - 현재는 대부분 'SDK Platform-tools'나 'SDK Build-tools'에 통합되어 제공되거나, Android Studio 자체에 내장

Android SDK 구성 요소 (요약)

- 이 네 가지는 서로 다른 역할을 수행하며, 함께 동작하여 완전한 Android 개발 환경을 구성
    - Platform 은 개발 대상 OS 버전을 제공하고,
    - Platform-tools 는 기기와의 통신을 담당하며,
    - Build-tools 는 실제 앱을 만드는 역할

The Android Manifest

- 안드로이드 앱의 핵심적인 구성 정보를 담고 있는 설정 파일
- 앱의 전체적인 구조와 동작 방식을 안드로이드 시스템에 알리는 역할
- 앱의 모든 컴포넌트(액티비티, 서비스, 브로드캐스트 리시버, 콘텐츠 프로바이더)는 이 파일에 반드시 선언되어야 함

주요 내용과 기능

1. 앱 컴포넌트 선언

- 앱을 구성하는 4 가지 주요 컴포넌트들을 선언
    - Activity: 사용자 인터페이스를 가지는 화면을 나타냄
    - Service: 백그라운드에서 실행되는 작업을 담당
    - Broadcast Receiver: 시스템 또는 다른 앱에서 발생하는 이벤트(예: 배터리 부족, 부팅 완료)를 수신
    - Content Provider: 앱의 데이터를 다른 앱과 공유

2. 앱 권한 설정 (<uses-permission>)

- 앱이 기기의 특정 기능(예: 인터넷 사용, 카메라 접근, 위치 정보)에 접근하기 위해 필요한 권한을 명시
- 사용자가 앱을 설치할 때 이 권한 목록을 확인하고 동의

3. 하드웨어 및 소프트웨어 기능 명시 (<uses-feature>)

- 앱이 동작하기 위해 특정 하드웨어(예: 카메라)나 소프트웨어 기능(예: OpenGL)이 필요함을 선언
- 이를 통해 Google Play 스토어에서 해당 기능을 지원하는 기기에서만 앱이 노출되도록 할 수 있음

4. 인텐트 필터 (<intent-filter>)

- 컴포넌트가 특정 종류의 인텐트를 수신할 수 있도록 설정
- 예를 들어, 웹 브라우저 앱은 android.intent.action.VIEW 액션을 처리하는 인텐트 필터를 선언하여
  웹 주소를 클릭했을 때 해당 앱이 실행되도록 할 수 있음

Android Launcher

- 사용자의 스마트폰 홈 화면을 관리하고, 설치된 앱들을 실행하는 핵심적인 사용자 인터페이스
    - 가장 먼저 보는 화면과 앱 목록을 보여주는 역할을 하는 앱

주요 기능

- 홈 화면: 사용자가 자주 사용하는 앱 아이콘, 위젯 등을 배치할 수 있는 메인 화면
- 앱 서랍(App Drawer): 기기에 설치된 모든 앱 아이콘을 모아 보여주는 화면
- 독 바(Dock Bar): 화면 하단에 고정되어 있어 어떤 화면에 있든 항상 접근할 수 있는 자주 사용하는 앱 아이콘 모음
- 위젯(Widgets): 앱을 실행하지 않아도 주요 정보를 한눈에 볼 수 있도록 하는 작은 앱 화면 (예: 날씨 위젯, 시계 위젯)

기본 런처 외 사용자 런처

- 사용자가 원하는 다른 런처 앱을 설치하여 사용 가능
  (예: Nova Launcher, Microsoft Launcher, Niagara Launcher)

The Android Permissions (1)

Permission 의 종류와 동작 방식

- 일반 권한(Normal Permissions)과 위험한 권한(Dangerous Permissions)으로 나뉨

일반 권한 (Normal Permissions)

- 앱의 프라이버시 침해 위험이 적은 권한
- 앱 설치 시 사용자의 동의 없이 자동으로 부여
- 예시:
    - 인터넷 연결(android.permission.INTERNET)
    - 진동 제어(android.permission.VIBRATE)

위험한 권한 (Dangerous Permissions)

- 사용자의 개인 정보나 기기 기능에 민감하게 접근하는 권한
- 앱 사용 중 사용자에게 동의를 구하는 런타임 권한 요청이 필요
    - 사용자가 동의를 거부하면 앱은 해당 기능을 사용할 수 없음
- 예시:
    - 카메라(android.permission.CAMERA)
    - 위치 정보(android.permission.ACCESS_FINE_LOCATION)
    - 연락처(android.permission.READ_CONTACTS)

The Android Permissions (2)

signature 권한

- 앱이 같은 개발자에 의해 서명된 다른 앱과 독점적으로 통신할 수 있도록 허용하는 특별한 종류의 권한
- 앱을 배포할 때 사용하는 디지털 서명(Signing Key)에 의해 결정
- 일반적으로 signature 권한을 요청하는 앱과 그 권한을 제공하는 앱이 동일한 서명 키로 서명되어 있어야 함

signature 권한의 주요 용도

- 앱 간의 안전한 통신: 한 개발사의 앱들이 서로 안전하게 데이터를 주고받거나 기능을 공유해야 할 때
- 보안 강화: 악의적인 앱이 민감한 내부 API 나 기능에 접근하는 것을 막아줌
- 기기 제조사 앱: 스마트폰 제조사 자체 설치 시스템 앱(예: 삼성페이, LG Q 메모)들이 서로 통신할 때 사용

The Android Permissions (3)

안드로이드 앱의 권한 설정 방식

- 두 가지 주요 방법 존재
    - AndroidManifest.xml 파일에 권한을 선언
    - 사용자에게 직접 권한을 요청
- 이 두 가지 방법은 함께 사용되어야 함

The Android Permissions

1. 매니페스트 파일에 권한 선언하기

- AndroidManifest.xml 파일에 <uses-permission> 태그를 사용하여 미리 선언해야 함
- 앱이 실행되기 전에 안드로이드 시스템에 해당 기능이 필요함을 알리는 역할
- 예시: 앱이 인터넷을 사용하거나 카메라에 접근해야 한다면 다음과 같이 선언

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.myapp">
```

```
    <uses-permission android:name="android.permission.INTERNET" />

    <uses-permission android:name="android.permission.CAMERA" />

    <application ...>

        ...

    </application>

</manifest>
```

2. 런타임 권한 요청하기 (API 23 이상)

- “위험한 권한”(예: 위치, 카메라, 저장소 접근 등)은 앱 설치 시 자동 부여되지 않음
- 대신, 앱이 해당 기능을 처음 사용할 때 사용자에게 직접 권한을 요청하고 승인받아야 함

작동 순서

1. 권한 확인:
   ContextCompat.checkSelfPermission() 메서드로 앱이 이미 권한을 가지고 있는지 확인
2. 권한 요청:
   권한이 없을 경우 ActivityCompat.requestPermissions() 호출 → 사용자에게 권한 요청 대화상자 표시
3. 결과 처리:
   사용자가 응답하면 onRequestPermissionsResult() 콜백 메서드가 호출되어 권한 승인 여부에 따라 앱 동작을 처리

Packages, Modules, and Libraries

패키지 (Package)

- 자바나 코틀린 언어에서 사용되는 논리적 그룹 단위
- 관련된 클래스나 파일들을 하나의 이름 공간(namespace)으로 묶어 관리
- 코드의 가독성과 유지보수성을 높이기 위해 사용되며, 파일 시스템의 폴더 구조와 유사
- 예시:
  com.example.myapp.ui 패키지에는 사용자 인터페이스 관련 클래스들이 포함될 수 있음

모듈 (Module)

- Android Studio 프로젝트의 독립적인 구성 단위
- 각 모듈은 자체적으로 컴파일, 테스트, 디버깅 가능하며, 자체 빌드 설정(build.gradle)을 가짐
- 하나의 프로젝트는 여러 개의 모듈로 구성될 수 있음

모듈 종류

- 앱 모듈: 사용자가 설치할 수 있는 .apk 파일을 생성
- 라이브러리 모듈: 다른 모듈이나 앱에서 재사용할 수 있는 코드를 포함

라이브러리 (Library)

- 재사용 가능한 코드의 집합을 의미하는 일반 용어
- 특정 기능(예: 이미지 로딩, 네트워킹 등)을 미리 구현해 둔 것으로
  개발자가 처음부터 작성할 필요 없이 프로젝트에 추가하여 사용 가능

- 안드로이드에서는 라이브러리를 모듈 형태로 만들거나 외부에서 가져와 의존성(dependency) 으로 추가 가능