

Android Runtime Permissions

Dice

Android Permission

- Android 6.0 (API 23) 이후 민감한 권한은 Runtime에 명시적으로 요청 필요
- 해당 기능 실행 시점에 권한을 요청하도록 권장

Android Permissions 분류

Android Permission은 protectionLevel에 따라 3가지로 분류됨

Normal Permission

- 시스템이 자동으로 허용
- 사용자의 개인정보와 거의 무관
- INTERNET, ACCESS_NETWORK_STATE

Android Permissions 분류

Android Permission은 protectionLevel에 따라 3가지로 분류됨

Dangerous Permission

- 민감 정보 접근 -> 런타임에 사용자 동의 필요
- 권한이 필요한 기능 실행 직전에 요청하도록 권장
- CAMERA, READ_CONTACTS, RECORD_AUDIO

Android Permissions 분류

Android Permission은 protectionLevel에 따라 3가지로 분류됨

Signature Permission

- 동일한 앱 서명 간에만 사용 가능
- 시스템 혹은 동일 개발자 앱 간 공유
- INSTALL_PACKAGES, ACCESS_CACHE_FILESYSTEM

Android Permissions 분류

Android Permission은 `protectionLevel`에 따라 3가지로 분류됨

대부분의 앱 개발자에게는

“Normal”과 “Dangerous” 권한만 신경 쓰면 충분

Dangerous Permission

- 카메라, 마이크, 연락처, 위치, 저장소 등은 위험 권한으로 분류
- AndroidManifest.xml 선언 + Runtime 권한 요청 필요

권한 요청 기본 흐름

사용자가 해당 권한이 필요한 기능과 상호 작용할 때 권한을 요청하기를 권장

```
when {  
    ContextCompat.checkSelfPermission(...) == GRANTED -> {  
        // 이미 권한 있음 → 기능 실행  
    }  
    shouldShowRequestPermissionRationale(...) -> {  
        // 이전 거부 → 설명 UI  
    }  
    else -> {  
        // 권한 요청  
        requestPermissionLauncher.launch(...)  
    }  
}
```


권한 요청

ActivityResultLauncher

- ActivityResultLauncher를 사용하여 콜백 기반으로 권한 처리를 단순화할 수 있음
- 해당 API를 사용하면 시스템은 사용자에게 권한 요청을 허용하거나 거부하도록 안내

```
private val requestPermissionLauncher = registerForActivityResult(  
    ActivityResultContracts.RequestPermission()  
) { isGranted: Boolean ->  
    if (isGranted) {  
        // ✅ 권한 부여됨  
        startCamera()  
    } else {  
        // ❌ 권한 거부됨  
        showPermissionDeniedDialog()  
    }  
}
```

권한 설명 제공

Rationale

- 경우에 따라 시스템은 `shouldRequestPermissionRationale()` 메서드를 사용하여 권한을 요청하기 전에 해당 기능을 사용하기 위해 권한이 필요한 근거를 표시할 것을 권장
- 이는 사용자 경험을 개선하고 권한 획득 가능성을 높일 수 있음

```
fun showPermissionRationale() {  
    AlertDialog.Builder(this)  
        .setTitle("권한 필요")  
        .setMessage("이 기능이 제대로 동작하기 위해 카메라 접근 권한이 필요합니다.")  
        .setPositiveButton("확인") { _, _ ->  
            requestPermissionLauncher.launch(Manifest.permission.CAMERA)  
        }  
        .setNegativeButton("취소", null)  
        .show()  
}
```

권한 거부 & 다시 묻지 않음 대응

권한을 여러 번 거부하면 안드로이드는 이를 영구 거부로 처리하여 앱이 다시 요청할 수 없음
앱은 사용자에게 기능 제한에 대해 알리고 필요한 경우 시스템 설정으로 안내

- 더 이상 시스템 권한 요청 팝업이 노출되지 않음
- 설정 화면으로 유도하는 UI 필요

```
if (
    ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED &&
    !ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.CAMERA)
) {
    // ! 영구 거부 상태
    showSettingsDialog()
}
```

설정 화면으로 유도

- 앱 설정 화면 -> 사용자가 수동으로 권한 허용 가능

```
val intent = Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS).apply {  
    data = Uri.parse("package:$packageName")  
}
```

```
AlertDialog.Builder(this)  
    .setTitle("권한 필요")  
    .setMessage("앱 설정에서 카메라 권한을 허용해주세요.")  
    .setPositiveButton("설정 열기") { _, _ -> startActivity(intent) }  
    .setNegativeButton("취소", null)  
    .show()
```

위치 권한 처리 흐름

위치 권한은 API 29 이상부터는 2단계로 나눠서 요청해야 함

- ACCESS_FINE_LOCATION 또는 COARSE 요청
- 이후 ACCESS_BACKGROUND_LOCATION 별도 요청

Background 권한은 위치 추적 목적이 명확한 경우에만 요청하도록 주의

일회성 권한

- Android 11 (API 30)부터 도입됨
- 위치, 마이크, 카메라 권한에만 적용
- 앱이 백그라운드로 가거나 종료되면 권한도 제거됨
- 사용자가 일시적으로만 허용할 수 있도록 설계

기능별 권한 처리

카메라 권한 • 기능 실행 직전 요청

- 명확한 트리거 (ex. QR 스캔, 사진 등록)

위치 권한 • 대부분 Foreground로 충분

- Background는 지속 추적 등 명확한 이유 있을 때만 사용

마이크 권한 • 시각적 UI 표시 필수 (녹음 중 아이콘 등)

- Android 12부터 자동 표시

사용자 경험(UX)를 고려한 권한 요청 설계

- 권한은 앱 시작이 아닌 기능 사용 직전 요청
- 권한이 거부된 경우
 - 대체 UI 제공
 - 안내 메시지 -> 설정 화면 유도
- 과도한 요청은 사용자에게 불신을 유발할 수 있음

정리

- 권한은 기능 사용 직전에 요청
- 거부 이력이 있다면 해당 권한이 필요한 이유를 설명하는 UI 제공
- 거부 시 Fallback 처리
- 다시 묻지 않음인 경우 설정 화면으로 유도

권한은 기술보다 UI 문제, 설득과 신뢰의 흐름을 갖춘 설계가 필요

참고 자료

[Android 공식 권한 가이드]

<https://developer.android.com/training/permissions/requesting>

[Manifest.permission 문서]

<https://developer.android.com/reference/android/Manifest.permission>

[Android Manifest Interview 도서 (KOR)]

<https://leanpub.com/manifest-android-interview-kr>