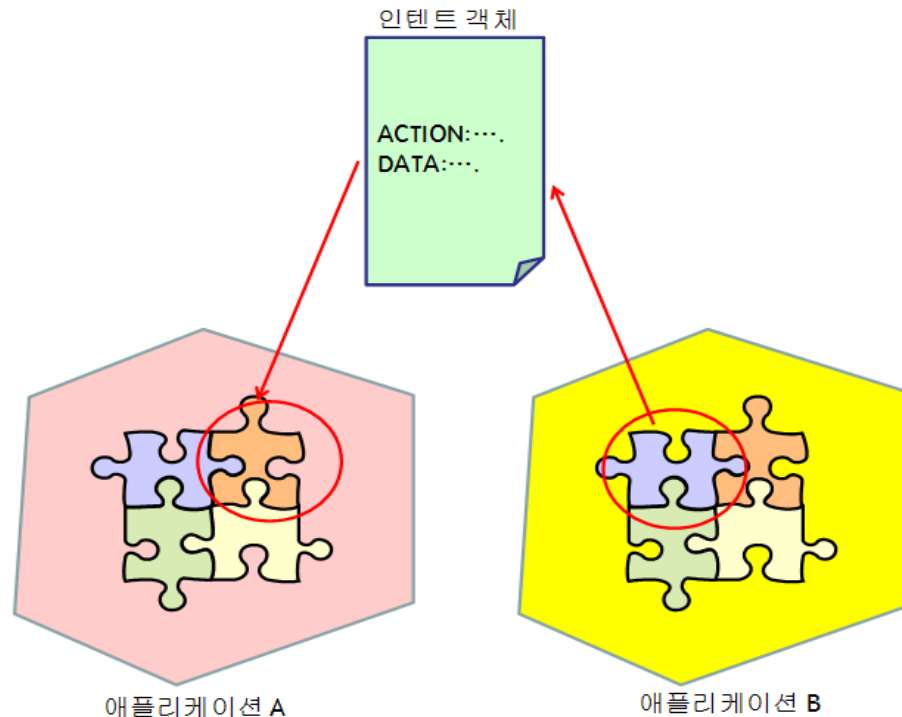


Intent

Intent

- 안드로이드의 컴포넌트끼리 통신하기 위한 메시지 시스템
- 다른 액티비티를 시작하려면 액티비티의 실행에 필요한 여러가지 정보들을 보내주는 역할을 수행함



Intent의 종류

- 명시적 인텐트(explicit intent)
 - 인텐트를 수행할 class를 명시적으로 지정

```
Intent(Context packageContext, Class<?> cls)
```

```
val intent = Intent(FirstActivity.this, SecondActivity::class.java);  
startActivity(intent)
```

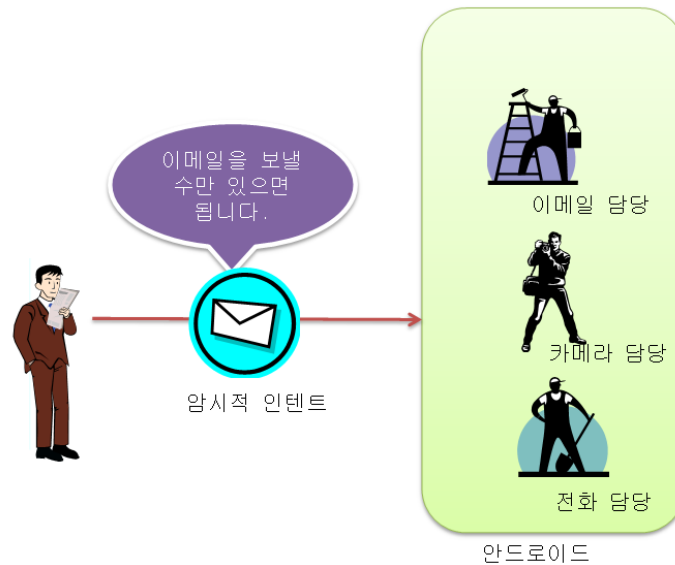
- 암시적 인텐트(implicit intent)
 - 인텐트의 Action 메시지를 통해 함축적으로 수행할 class가 결정

```
Intent(String action)  
Intent(String action, Uri uri)
```

```
val intent = Intent(Intent.ACTION_DIAL);  
val intent = Intent(Intent.ACTION_CALL);  
val intent = Intent(Intent.ACTION_CALL, Uri.parse("tel:"+"114"));  
startActivity(context, intent, null);
```

Implicit Intent

- 어떤 작업을 하기를 원하지만 그 작업을 담당하는 컴포넌트의 이름을 정확하게 모르는 경우에 사용



```
val uri = Uri.parse("mailto:"+editText.text.toString())  
val intent = Intent(Intent.ACTION_SENDTO, uri)  
startActivity(context, intent, null)
```

Implicit Intent

- 액션의 종류

상수	타겟 컴포넌트	액션
ACTIN_VIEW	액티비티	데이터를 사용자에게 표시한다.
ACTION_EDIT	액티비티	사용자가 편집할 수 있는 데이터를 표시한다.
ACTION_MAIN	액티비티	태스크의 초기 액티비티로 설정한다.
ACTION_CALL	액티비티	전화 통화를 시작한다.
ACTION_DIAL	액티비티	전화 번호를 누르는 화면을 표시한다.

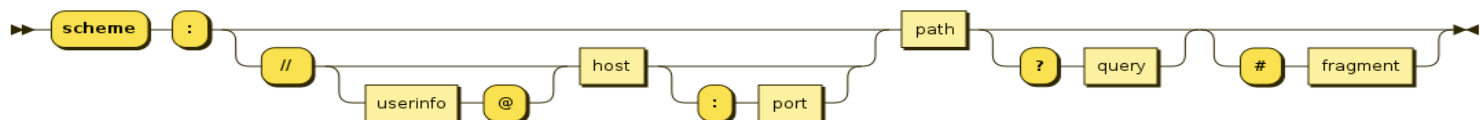
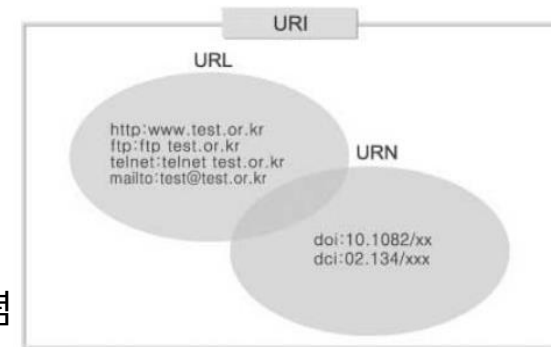
Implicit Intent

• 암시적 인텐트 예

- ▶ ACTION_VIEW *content://contacts/people/1* - 1번 연락처 정보를 표시한다.
- ▶ ACTION_DIAL *content://contacts/people/1* - 1번 연락처로 전화걸기 화면을 표시한다.
- ▶ ACTION_VIEW *tel:0101234567* - 0101234567번 전화번호로 전화걸기 화면을 표시한다.
- ▶ ACTION_DIAL *tel:0101234567* -- 0101234567번 전화번호로 전화걸기 화면을 표시한다.
- ▶ ACTION_EDIT *content://contacts/people/1* -- 1번 연락처 정보를 편집한다.
- ▶ ACTION_VIEW *content://contacts/people/* -- 연락처 리스트를 표시한다.

• URI (Uniform Resource Identifier)

- 인터넷상의 정보 자원에 대한 식별체계
 - 인터넷 주소체계(URL) + 고유이름 체계(URN)을 총칭하는 개념
- 문서, 이미지, 음악파일, 동영상 등 다양한 정보자원에 대한 유일성을 부여하고, 식별을 가능하게 하는 관리 체계



예제) Intent를 통한 화면 이동

```
val webpage = Uri.parse("http://www.naver.com")  
val webIntent = Intent(Intent.ACTION_VIEW, webpage)  
startActivity(context, webIntent, null)
```

```
val location = Uri.parse("geo:37.543684,127.077130?z=16")  
val mapIntent = Intent(Intent.ACTION_VIEW, location)  
context.startActivity(context, mapIntent, null)
```

```
val message = Uri.parse("sms:010-1234-1234")  
val messageIntent = Intent(Intent.ACTION_SENDTO, message)  
messageIntent.putExtra("sms_body", "밥 먹자....")  
context.startActivity(context, messageIntent, null)
```

```
val number = Uri.parse("tel:010-1234-1234")  
val callIntent = Intent(Intent.ACTION_DIAL, number)  
context.startActivity(context, callIntent, null)
```

네이버

맵

문자보내기

전화걸기

팀실습) youtube 화면 이동 추가

네이버

맵

문자보내기

youtube-아이유

전화걸기

예제) Intent를 통한 화면 이동

```
val webpage = Uri.parse("http://www.naver.com")  
val webIntent = Intent(Intent.ACTION_VIEW, webpage)  
startActivity(context, webIntent, null)
```

```
val location = Uri.parse("geo:37.543684,127.077130?z=16")  
val mapIntent = Intent(Intent.ACTION_VIEW, location)  
context.startActivity(context, mapIntent, null)
```

```
val message = Uri.parse("sms:010-1234-1234")  
val messageIntent = Intent(Intent.ACTION_SENDTO, message)  
messageIntent.putExtra("sms_body", "밥 먹자...")  
context.startActivity(context, messageIntent, null)
```

```
val number = Uri.parse("tel:010-1234-1234")  
val callIntent = Intent(Intent.ACTION_CALL, number)  
context.startActivity(context, callIntent, null)
```

네이버

맵

문자보내기

전화걸기

퍼미션

시스템 퍼미션

- 시스템에서 보호하고 있는 특정 기능을 위한 퍼미션
 - 이 기능을 사용하는 앱은 <uses-permission>을 반드시 선언해 주어야 함.

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

- 퍼미션 구조

<https://developer.android.com/guide/topics/manifest/permission-element?hl=ko>

[illegible]

시스템 퍼미션

- 퍼미션의 Protection Level <android:protectionLevel>
 - Normal
 - 위험성이 낮은 권한으로, 사용자의 명시적인 승인을 요구하지 않고 설치시 권한을 요청하는 애플리케이션에 권한을 자동 부여
 - Dangerous
 - 위험성이 높은 권한으로, 사용자에게 명시적으로 표시하고, 확인을 요청함
 - Signature
 - 동일한 키로 서명된 앱만 실행

시스템 퍼미션

- 대표적인 예

CALL_PHONE

Added in API level 1

```
public static final String CALL_PHONE
```

Allows an application to initiate a phone call without going through the Dialer user interface for the user to confirm the call.

Protection level: dangerous

.....> 사용자에게 권한 부여할 지 물어봄

Constant Value: "android.permission.CALL_PHONE"

- AndroidManifest.xml 파일에 퍼미션 요청

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

권한 요청

- AndroidManifest.xml 파일에 요청권한 등록

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

```
<uses-feature
```

```
    android:name="android.hardware.telephony"
```

```
    android:required="false" />
```

권한에 따라 추가

예) 전화 기능 없어도 앱 설치 가능

- 권한 요청

- 단일 권한 요청

- rememberPermissionState

- 다수의 권한 요청

- rememberMultiPermissionState

* 라이브러리 추가 : [com.google.accompanist:accompanist-permissions](#)

[@ExperimentalPermissionsApi](#)

Permission Check하기

- 안드로이드 6.0(Marshmallow)이후의 권한 설정 방법
 - 기기 제어(카메라, 블루투스, GPS, 파일읽기 등)할 수 있는 권한은 사용자의 동의를 받아야만 사용할 수 있고, **사용자는 언제든지 권한 사용을 취소할 수 있음**
 - 사용 절차
 - Manifest 파일에 권한 정보 등록하기
`<uses-permission android:name="android.permission.CALL_PHONE"/>`
 - 권한 체크
 - 만족하는 경우
 - » 정상 기능 수행
 - 만족하지 않는 경우
 - » 권한 요청
 - 권한 요청 결과에 따른 제어

Permission Check하기

- checkSelfPermission 함수

- 권한을 만족하는지 체크하는 함수
- 반환값 : PERMISSION_GRANTED / PERMISSION_DENIED

```
if( ActivityCompat.checkSelfPermission(context, Manifest.permission.CALL_PHONE)
    !=PackageManager.PERMISSION_GRANTED){
    // 권한이 승인 되지 않았을 때 수행할 내역
}
```


Permission Check하기

- rememberPermissionState 함수

- 퍼미션 상태 관리

```
var permissionState =
```

```
rememberPermissionState(permission = Manifest.permission.CALL_PHONE)
```

```
* permissionState.status.isGranted // 승인
```

```
* permissionState.status.shouldShowRationale // 거부
```

권한승인 ➔ 기능 수행

권한거부 ➔ 권한이 필요한 이유 설명

초기상태 ➔ 권한 요청

Permission Check하기

- RequestPermissions 함수(단일 권한)

- 권한 요청하는 함수
- 권한 설정 다이얼로그창이 띄워짐

- 요청 퍼미션 `permissionState.launchPermissionRequest()`

*권한 요청 다이얼로그 실행

- 승인 거부시 권한이 필요한 이유 설명

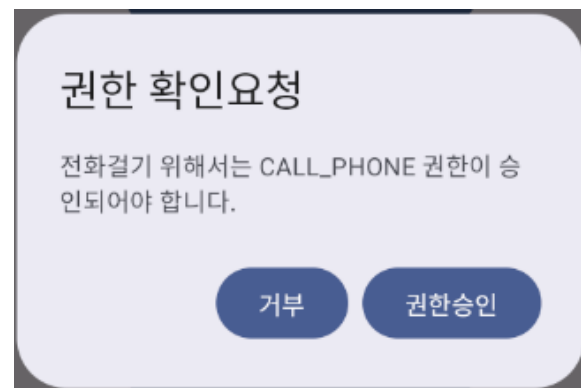
```
val requestPermissionLauncher = rememberLauncherForActivityResult(  
    ActivityResultContracts.RequestPermission()  
) { isGranted: Boolean ->  
    if (isGranted) {  
        // 권한 승인이 되었을 때  
    } else {  
        // 권한 승인이 거부되었을 때  
    }  
}
```

실행시킬 때

`requestPermissionLauncher.launch(Manifest.permission.CALL_PHONE)`

AlertDialog 창 만들기

```
@Composable
fun ShowCALLPermissionRationale(
    onDismiss: () -> Unit,
    onConfirm: () -> Unit,
) {
    AlertDialog(
        onDismissRequest = onDismiss,
        title = {
            Text(text: "권한 확인요청")
        },
        text = {
            Text(text: "전화걸기 위해서는 CALL_PHONE 권한이 승인되어야 합니다.")
        },
        confirmButton = {
            Button(onClick = onConfirm) { this: RowScope
                Text(text: "권한승인")
            }
        },
        dismissButton = {
            Button(onClick = onDismiss) { this: RowScope
                Text(text: "거부")
            }
        }
    )
}
```



예제. CALL_PHONE 권한 처리

- 전화걸기 기능 수행

네이버

맵

문자보내기

전화걸기

Permission Check하기

- 복수 권한 요청
 - RequestMultiplePermissions()

```
val permissions = arrayOf(  
    Manifest.permission.CALL_PHONE,  
    Manifest.permission.CAMERA)
```

```
val requestMultiplePermissionsLauncher =  
    rememberLauncherForActivityResult(ActivityResultContracts.RequestMultiplePermissions())  
{ map -> // map (permission, Boolean)  
  
}
```

```
requestMultiplePermissionsLauncher.launch(permissions)
```

예제. CAMERA 권한 추가하기

- 카메라 기능 수행하기

```
val cameraIntent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)  
startActivity(context, cameraIntent, null)
```

네이버

문자보내기

전화걸기

카메라

수고하셨습니다.