ENCS5150 · EXP 3 COMPANION

Intents and Notifications

How an app asks other apps for help, what happens when none can, and how to tell the user something from outside your app.

Five questions

01 What does an Intent carry?

02 How does Android choose which app answers?

03 What if no app can?

04 Toast, Snackbar or notification?

05 Why does notification code from 2019 fail today?

01 · WHAT AN INTENT CARRIES

An Intent is a message asking for an action

Field	Means	Example from the lab
Action	What to do	Intent.ACTION_DIAL, ACTION_VIEW, ACTION_SENDTO
Data	What to do it on, as a URI	tel:+97022982000, geo:31.959,35.182, mailto:
Category	Which kind of component may answer	CATEGORY_LAUNCHER
Component	The exact class, when you know it	AddCustomerActivity.class (Exp 2)
Extras	Extra key–value data	EXTRA_SUBJECT → “My Subject”

```
new Intent(Intent.ACTION_DIAL, Uri.parse("tel:+97022982000"));
```

01 · WHAT AN INTENT CARRIES

Explicit names the class; implicit names the job

EXPLICIT

“Open AddCustomerActivity”

```
new Intent(this, AddCustomerActivity.class)
```

Inside your own app, where you know the class. Always goes to that one screen.

IMPLICIT

“Show this place on a map”

```
new Intent(Intent.ACTION_VIEW, geoUri)
```

Any installed app that can do the job may answer: Google Maps, or another map app.

Implicit Intents are how apps cooperate without knowing each other: your app never needs the map app’s class name.

02 · CHOOSING AN APP

Intent resolution

`startActivity(new Intent(ACTION_VIEW, geo:31.959,35.182))`

↓

Android compares its action, data and category with the intent filters every installed app declares

↓

NO MATCH

ActivityNotFoundException

The app crashes, unless you catch it.

ONE MATCH

That app opens

Straight away, no question asked.

SEVERAL

The user picks

A chooser lists every app that matches.

03 · WHEN NO APP CAN

Start other apps inside a try

```
private void safeStart(Intent intent) {
    try {
        startActivity(intent);
    } catch (ActivityNotFoundException e) {
        Toast.makeText(this,
            "No app installed to handle this",
            Toast.LENGTH_SHORT).show();
    }
}
```

Why not check first?

resolveActivity() used to answer “is there an app?”. Since Android 11 it returns null unless the manifest lists the apps you may see, in a <queries> element.

So just try it

Catching the exception works on every version and needs no extra manifest entries.

04 · TELLING THE USER

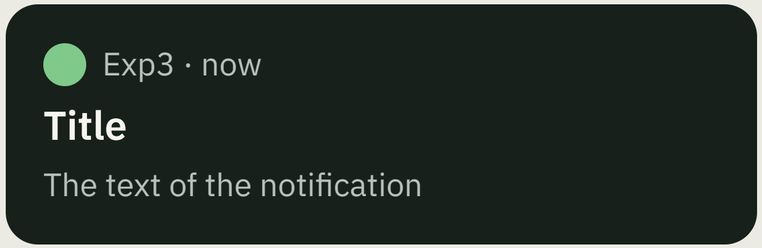
Toast, Snackbar or notification?

	Toast	Snackbar	Notification
Appears	Over any screen, briefly	At the bottom of your screen	In the status bar and drawer
The user can	Only read it	Tap one action, such as Undo	Tap it to open the app, or dismiss it
Stays	2 or 3.5 seconds	A few seconds, or until dismissed	Until the user clears it
Works when the app is closed	No	No	Yes
Use for	“Saved”	“Customer deleted. Undo”	“New message from Sara”

Snackbar comes from the Material Components library that the template already includes: **Snackbar.make(view, "Deleted", Snackbar.LENGTH_LONG)**.

04 · TELLING THE USER

What a notification is made of



Tapping it fires the PendingIntent, which reopens MainActivity, and auto-cancel removes it.

```
new NotificationCompat.Builder(this, MY_CHANNEL_ID)
    .setSmallIcon(R.drawable.ic_launcher_foreground)
    .setContentTitle(title)
    .setContentText(body)
    .setContentIntent(pendingIntent)
    .setAutoCancel(true);

NotificationManagerCompat.from(this)
    .notify(NOTIFICATION_ID, builder.build());
```

05 · WHAT CHANGED

Three releases, three new rules

ANDROID 8 · API 26

Channels

Every notification belongs to a channel the user can mute. No channel: nothing shows.

ANDROID 12 · API 31

PendingIntent flags

You must say FLAG_IMMUTABLE or FLAG_MUTABLE. Passing 0 throws an exception.

ANDROID 13 · API 33

Runtime permission

POST_NOTIFICATIONS must be granted by the user, or notify() is silently ignored.

The …Compat classes from AndroidX handle the version checks, so one piece of code works from API 24 up.

05 · WHAT CHANGED

Asking for a runtime permission

1 · Declare it in the manifest → 2 · Check: already granted? → 3 · If not, launch the request → 4 · The callback gets the answer

```
ActivityResultLauncher<String> requestPermissionLauncher =
    registerForActivityResult(new ActivityResultContracts.RequestPermission(),
        granted -> { if (granted) createNotification(); });

// when the button is pressed, on Android 13+ without the permission:
requestPermissionLauncher.launch(Manifest.permission.POST_NOTIFICATIONS);
```

The same four steps work for the camera, location and microphone. Ask when the user does something that needs it, not at launch.

TAKEAWAYS

Four things to remember

01 Explicit Intents name a class; implicit Intents describe a job any app may do.

02 No matching app means an exception: start other apps inside a try.

03 A notification needs a channel, an immutable PendingIntent and, on Android 13+, permission.

04 Use the …Compat classes, and old Android versions take care of themselves.