

# Android Development

---



# Launch modes - Flags

When starting an activity, you can modify the default association of an activity to its task by including flags in the intent that you deliver to `startActivity()`. The flags you can use to modify the default behavior are

## `FLAG_ACTIVITY_NEW_TASK`

Start the activity in a new task. If a task is already running for the activity you are now starting, that task is brought to the foreground with its last state restored and the activity receives the new intent in `onNewIntent()`. This produces the same behavior as the "singleTask" launchMode value, discussed in the previous section.

## `FLAG_ACTIVITY_SINGLE_TOP`

If the activity being started is the current activity (at the top of the back stack), then the existing instance receives a call to `onNewIntent()`, instead of creating a new instance of the activity. This produces the same behavior as the "singleTop" launchMode value, discussed in the previous section.

## `FLAG_ACTIVITY_CLEAR_TOP`

If the activity being started is already running in the current task, then instead of launching a new instance of that activity, all of the other activities on top of it are destroyed and this intent is delivered to the resumed instance of the activity (now on top), through `onNewIntent()`. There is no value for the launchMode attribute that produces this behavior.

# StartActivityForResult

```
val LAUNCH_SECOND_ACTIVITY = 1
val intentResult = Intent(this, SecondActivity::class.java)
startActivityForResult(intentResult, LAUNCH_SECOND_ACTIVITY)
```

—>FirstActivity

```
val returnIntent = Intent()
returnIntent.putExtra("result", result)
setResult(RESULT_OK, returnIntent)
finish()
```

—>SecondActivity

```
override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    super.onActivityResult(requestCode, resultCode, data)
    if (requestCode === LAUNCH_SECOND_ACTIVITY) {
        if (resultCode === RESULT_OK) {
            val result: String = attr.data.getStringExtra("result")
        }
        if (resultCode === RESULT_CANCELED) {
            // Write your code if there's no result
        }
    }
}
```

—>FirstActivity

# StartActivityForResult

```
var resultLauncher = registerForActivityResult(StartActivityForResult()) { result ->
    if (result.resultCode == Activity.RESULT_OK) {
        // There are no request codes
        val data: Intent? = result.data
        doSomeOperations()
    }
}
```

```
fun openSomeActivityResult() {
    val intent = Intent(this, SomeActivity::class.java)
    resultLauncher.launch(intent)
}
```