



Activity Result APIs



Here in my office, we
have used the best
practices to develop
Android apps.

How can we
ask for
audio
permission?



Show a popup saying:
*"Please go to the
settings screen and grant
all permissions the app
needs."*

Michael is
crazy!
Let's use the
Activity Result
API.



First, import the `ActivityResultCallback`.

```
import androidx.activity.result.ActivityResultCallback
```

```
class MainActivity : ComponentActivity(), ActivityResultCallback<Boolean> {
```

Check if user has permission
or not.
If not, launch requesting
the permission.

```
class MainActivity : ComponentActivity(), ActivityResultCallback<Boolean> {  
  
    private lateinit var launcher: ActivityResultLauncher<String>  
    private val viewModel: MainViewModel by viewModels { MainViewModel.Factory }  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView {...}  
  
        if (hasRecordAudioPermission().not()) {  
            launcher = registerForActivityResult(  
                ActivityResultContracts.RequestPermission(),  
                callback: this  
            )  
            launcher.launch(Manifest.permission.RECORD_AUDIO)  
        }  
    }  
}
```

The result will
call the
following
method:

```
class MainActivity : ComponentActivity(), ActivityResultCallback<Boolean> {  
  
    private lateinit var launcher: ActivityResultLauncher<String>  
    private val viewModel: MainViewModel by viewModels { MainViewModel.Factory }  
  
    Gabriel Bronzatti Moro  
    override fun onCreate(savedInstanceState: Bundle?) {...}  
  
    Gabriel Bronzatti Moro  
    override fun onActivityResult(result: Boolean?) {  
        if (result != true) {  
            Toast.makeText(context: this, text: "Permissions are required", Toast.LENGTH_SHORT).show()  
            finish()  
        }  
    }  
}
```



In case we have multiple permissions, how can we handle them?

To multiple permissions, you need to update the code:

1. Update the interface you are implementing:

```
ActivityResultCallback<Map<String, Boolean>>
```

2. Launcher will ask for **multiple permissions**, so we have an array:

```
private lateinit var launcher: ActivityResultLauncher<Array<String>>
```

3. Update the launcher initialization:

```
launcher = registerForActivityResult(  
    ActivityResultContracts.RequestMultiplePermissions(),  
    callback: this  
)  
launcher.launch(arrayOf(Manifest.permission.RECORD_AUDIO, Manifest.permission.CAMERA))
```

4. You will have a boolean for each permission saying if it is granted or not.

```
override fun onActivityResult(result: Map<String, Boolean>)
```





I'm a model for these people. I must guide them as a jedi master.

