

# Android Dev Kotlin



CS 402: Mobile Development



# Permissions Overview

To access certain device features, you have to ask the user for permission.

Some are granted on install.

Others need to be asked at run time.



# Steps

Add permission to AndroidManifest.xml

Check permission

If granted, launch camera activity

If not yet granted ask

If already asked and denied, show rationale



# Steps

Launch camera with `startActivityResult`

Take the picture

Grab the bitmap data from the intent data

Display bitmap data in `ImageView`



# Permissions

```
val permissionCheck = ContextCompat.checkSelfPermission(this,  
    Manifest.permission.[PERMISSION])
```

CAMERA

ACCESS\_COARSE\_LOCATION

INTERNET

READ\_CONTACTS

SEND\_SMS

WRITE\_EXTERNAL\_STORAGE

<https://developer.android.com/guide/topics/permissions/requesting.html#perm-groups>



# Permission Statuses

`PackageManager.PERMISSION_GRANTED`

`PackageManager.PERMISSION_DENIED`

```
if (permissionCheck != PackageManager.PERMISSION_GRANTED) {  
    Log.i("BSU", "Permission to record denied")  
    requestPermissions()  
}
```



# Request Permission

```
private fun requestPermissions(){  
  
    ActivityCompat.requestPermissions(this,  
  
    arrayOf(android.Manifest.permission.CAMERA),  
  
    888)  
}
```



# Permission Request Result

```
override fun onRequestPermissionsResult(requestCode: Int, permissions:  
Array<out String>, grantResults: IntArray) {
```

```
    super.onRequestPermissionsResult(requestCode, permissions, grantResults)
```

```
}
```





# Permission Request Result

```
if (permission != PackageManager.PERMISSION_GRANTED) {  
    if (ActivityCompat.shouldShowRequestPermissionRationale(this, Manifest.permission.CAMERA)) {  
  
        val builder = AlertDialog.Builder(this)  
        builder.setMessage("Permission to access the camera is required.")  
        .setTitle("Permission required")  
        builder.setPositiveButton("OK") { dialog, id ->  
            requestPermissions()  
        }  
        val dialog = builder.create()  
        dialog.show()  
    } else {  
        requestPermissions()  
    }  
}
```



# Camera

```
val intent = Intent(MediaStore.ACTION_IMAGE_CAPTURE)  
  
startActivityForResult(intent, 999)
```