CS371m - Mobile Computing

Runtime Permissions

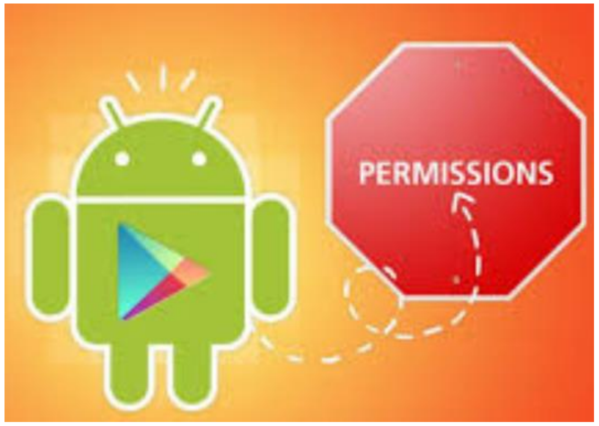
Clicker

- If your app wants to access the users contacts when do you request permission to do this?

A. Never, its not necessary
B. At compile time
C. At install time
D. At runtime
E. It depends

System Permissions

- In an attempt to maintain security (system integrity, user privacy) on Android devices …
- … run each app in a limited sandbox
- If app wants to use resources (e.g. camera, storage, network) or information (e.g. contacts info) outside of sandbox, must request permission.
- List of permissions: https://developer.android.com/reference/android/Manifest.permission.html



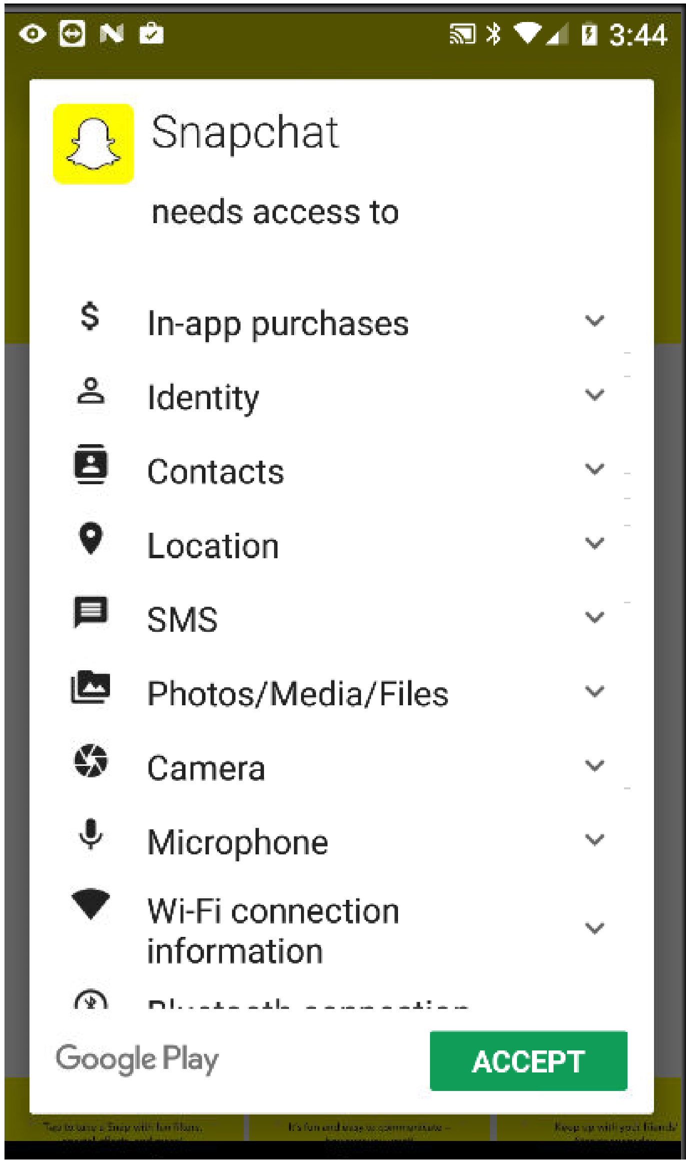
History of Permissions

- Prior to Android version 6.0, Marshmallow…
- Apps requested permission at install time
- Permissions Listed in the AndroidManifest.xml File

```xml
<uses-permission android:name=
    "android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name=
    "android.permission.READ_EXTERNAL_STORAGE" />
```

Install Time Permissions

- Pre 6.0 / Marshmallow user granted permission at *install time.*
- Google Play listed permissions for app when install selected
- Based on permissions listed in manifest
- Some apps … “we own you”



3:44

Snapchat

needs access to

- In-app purchases
- Identity
- Contacts
- Location
- SMS
- Photos/Media/Files
- Camera
- Microphone
- Wi-Fi connection information

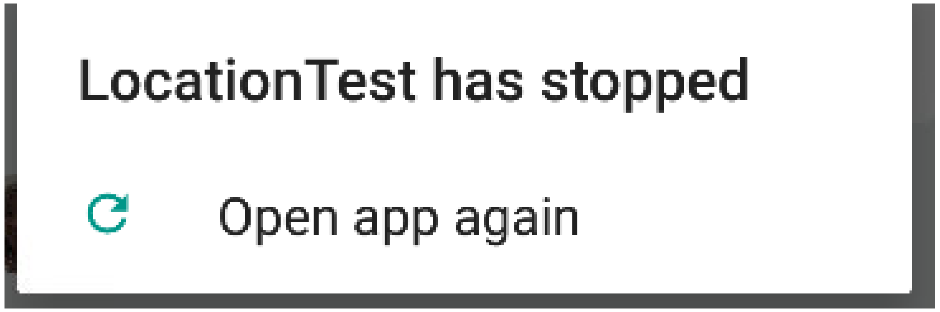
Clicker

- What happens if an app tries to use a resource or access information it doesn’t have permission to use?

A. Compile error
B. Runtime error
C. Nothing, app allowed to access info or resource
D. Nothing, BUT app not allowed to access info or resource, a NO-OP

Lacks Permission

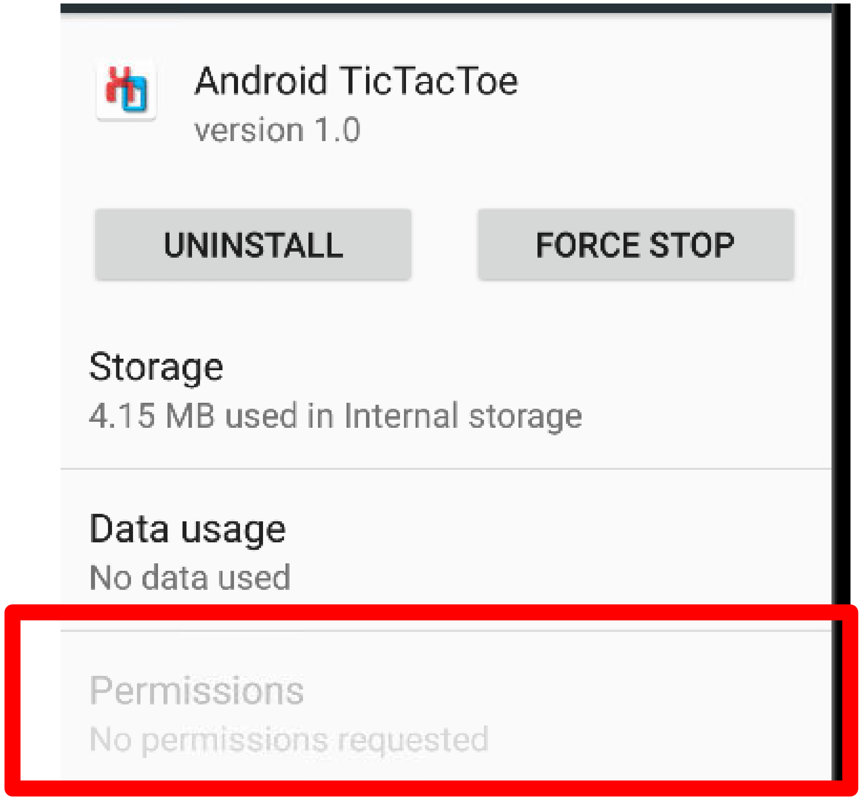
- App stopped by system as soon as it tries to perform operation it doesn't have permission for


LocationTest has stopped

```
02-09 16:03:39.857 26846-26846/org.example.locationtest
        E/AndroidRuntime: FATAL EXCEPTION: main
Process: org.example.locationtest, PID: 26846
java.lang.RuntimeException: Unable to start activity
        ComponentInfo
        {org.example.locationtest/org.example.locationtest.LocationTest}:
        java.lang.SecurityException: "passive" location provider requires
        ACCESS_FINE_LOCATION permission.
at android.location.LocationManager.getProvider(LocationManager.java:383)
at org.example.locationtest.LocationTest.dumpProvider(LocationTest.java:372)
at org.example.locationtest.LocationTest.dumpProviders(LocationTest.java:364)
at org.example.locationtest.LocationTest.onCreate(LocationTest.java:80)
```

Viewing Permissions

- A user can see what permissions an app has in Settings -> Apps



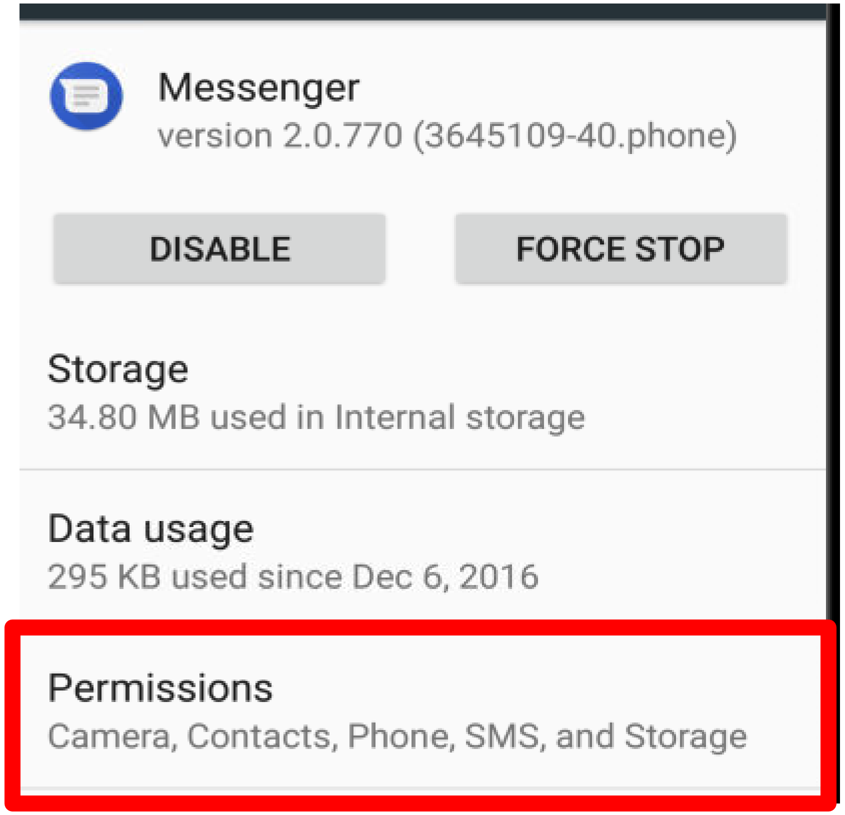
Android TicTacToe
version 1.0

UNINSTALL FORCE STOP

Storage
4.15 MB used in Internal storage

Data usage
No data used

Permissions
No permissions requested



Messenger
version 2.0.770 (3645109-40.phone)

DISABLE FORCE STOP

Storage
34.80 MB used in Internal storage

Data usage
295 KB used since Dec 6, 2016

Permissions
Camera, Contacts, Phone, SMS, and Storage

- Developers can see device permissions via adb: $ adb shell pm list permissions -s

Changes to Permissions in M

- Android 6.0 / Marshmallow introduced changes to permissions
- Introduced Permission Groups
- Introduced Normal and Dangerous Permissions
  - “Dangerous Permissions cover areas where the app wants data or resources that involve the user's private information, or could potentially affect the user's stored data or the operation of other apps.”

Normal Permissions – API 23

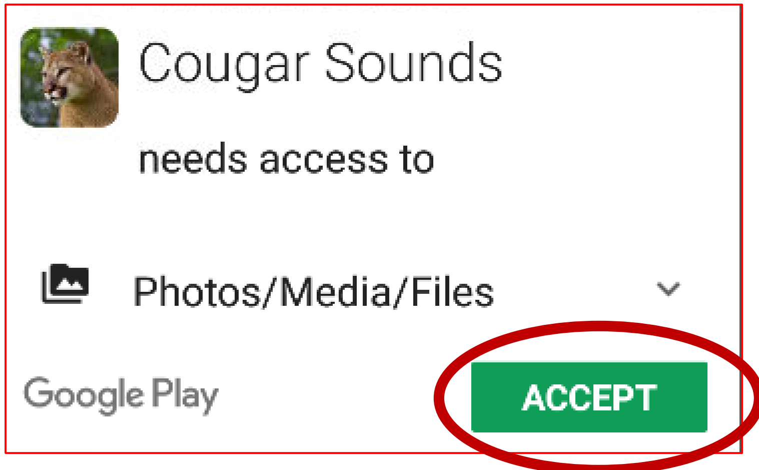
- ACCESS_LOCATION_EXTRA_COMMANDS
- ACCESS_NETWORK_STATE
- ACCESS_NOTIFICATION_POLICY
- ACCESS_WIFI_STATE
- **BLUETOOTH**
- BLUETOOTH_ADMIN
- BROADCAST_STICKY
- CHANGE_NETWORK_STATE
- CHANGE_WIFI_MULTICAST_STATE
- CHANGE_WIFI_STATE
- DISABLE_KEYGUARD
- EXPAND_STATUS_BAR
- GET_PACKAGE_SIZE
- INSTALL_SHORTCUT
- **INTERNET**
- KILL_BACKGROUND_PROCESSES
- MODIFY_AUDIO_SETTINGS
- NFC
- READ_SYNC_SETTINGS
- READ_SYNC_STATS
- RECEIVE_BOOT_COMPLETED
- REORDER_TASKS
- REQUEST_IGNORE_BATTERY_OPTIMIZATIONS
- REQUEST_INSTALL_PACKAGES
- **SET_ALARM**
- SET_TIME_ZONE
- SET_WALLPAPER
- SET_WALLPAPER_HINTS
- TRANSMIT_IR
- UNINSTALL_SHORTCUT
- USE_FINGERPRINT
- **VIBRATE**
- **WAKE_LOCK**
- WRITE_SYNC_SETTINGS

Dangerous Permissions

- Dangerous Permissions Organized into Permission Groups:
- **CALENDAR**: READ_CALENDAR, WRITE_CALENDAR
- **CAMERA**: CAMERA
- **CONTACTS**: READ_CONTACTS, WRITE_CONTACTS, GET_ACCOUNTS
- **LOCATION**: ACCESS_FINE_LOCATION, ACCESS_COARSE_LOCATION
- **MICROPHONE**: RECORD_AUDIO
- **PHONE**: READ_PHONE_STATE, CALL_PHONE, READ_CALL_LOG, WRITE_CALL_LOG, ADD_VOICEMAIL, USE_SIP, PROCESS_OUTGOING_CALLS
- **SENSORS**: BODY_SENSORS
- **SMS**: SEND_SMS, RECEIVE_SMS, READ_SMS, RECEIVE_WAP_PUSH, RECEIVE_MMS
- **STORAGE**: READ_EXTERNAL_STORAGE, WRITE_EXTERNAL_STORAGE

Runtime Permissions

- Post Android 6.0 / Marhsmallow
- All Normal and Dangerous Permissions **<u>still</u>** placed in app Manifest
- If Device is Android 5.1 or lower OR app targets API level 22 or lower …
- … then user must grant Dangerous Permissions at install time
- Or app doesn't install


Cougar Sounds

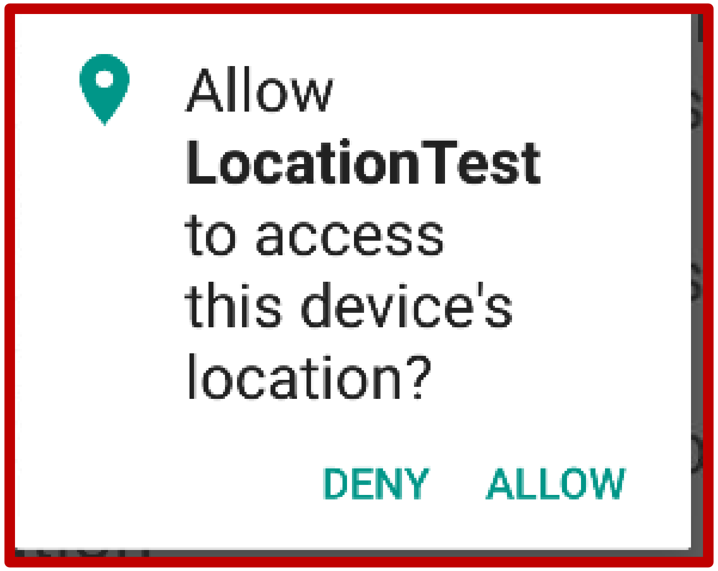
needs access to

Photos/Media/Files

Google Play

Runtime Permissions

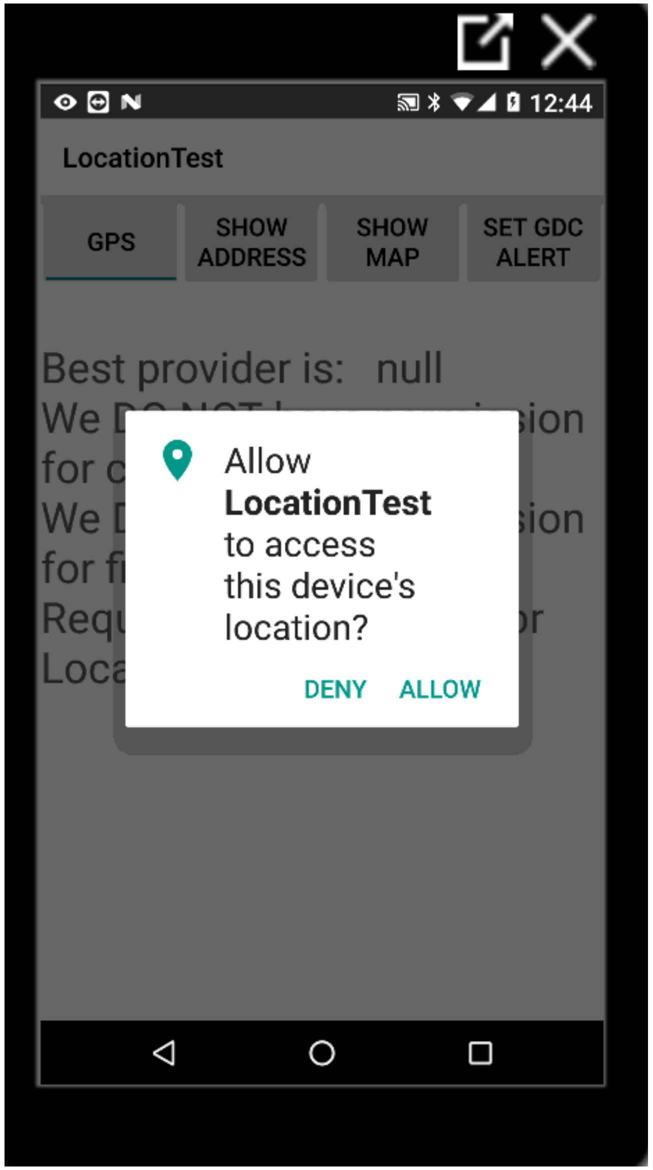
- If device is Android 6.0 or higher AND app targets API level 23 or higher
- Must still list all permissions in manifest
- App must request each dangerous permission in needs **while the app is running**



Allow **LocationTest** to access this device's location?

Requesting Permission at Runtime

- When app needs dangerous permission:
- Check to see if you already have permission
- If not call one of the requestPermission methods with desired permissions and request code
- requestPermission shows a standard, non modifiable dialog box to the user



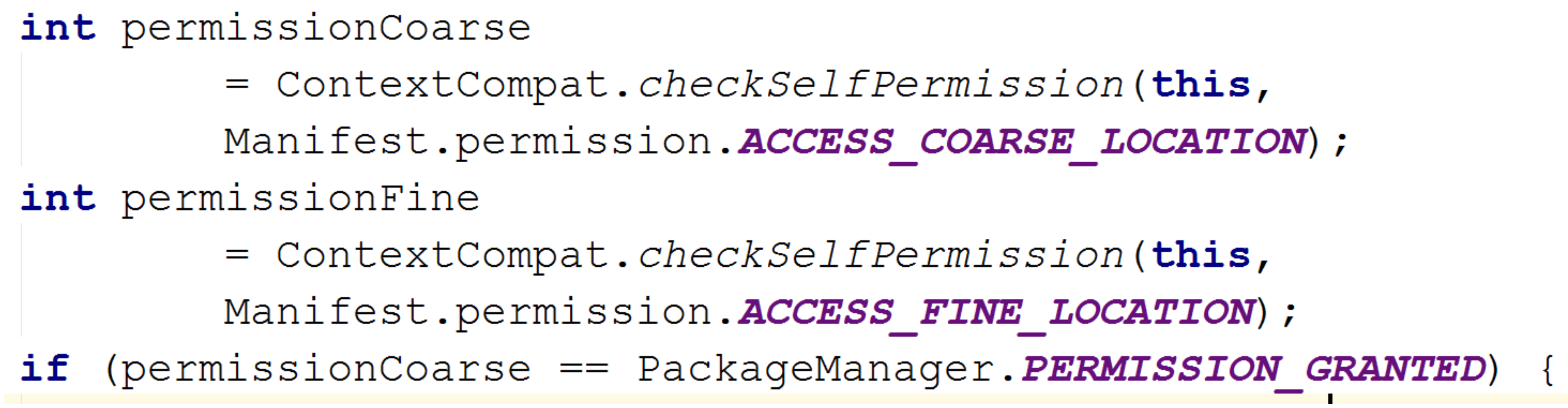
Best provider is: null

Allow **LocationTest** to access this device's location?

DENY ALLOW

Checking Permissions

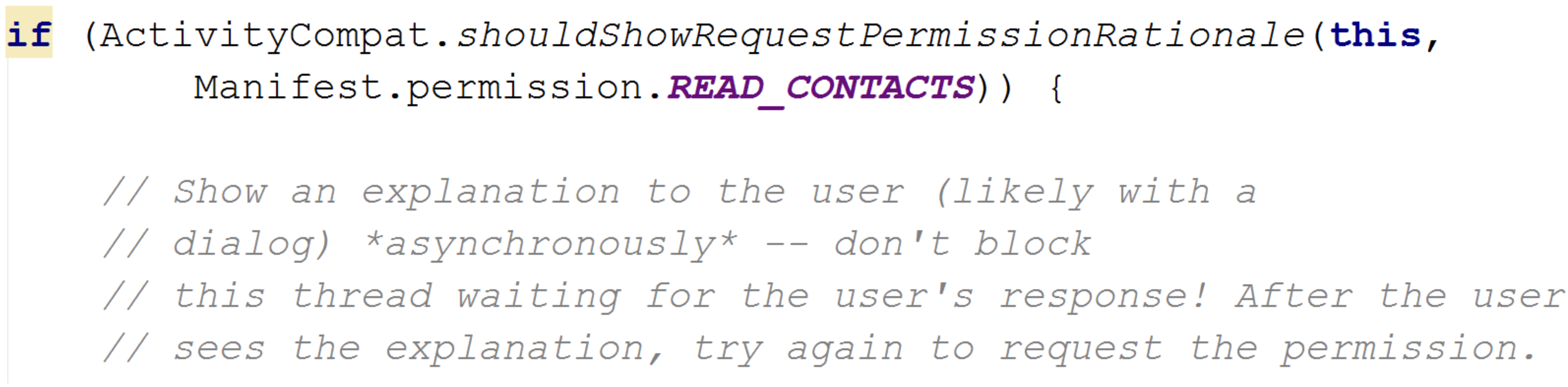
- Before Requesting a Permission Check to see if you already have it.



```
int permissionCoarse
        = ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION);
int permissionFine
        = ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_FINE_LOCATION);
if (permissionCoarse == PackageManager.PERMISSION_GRANTED) {
```

Requesting Permission

- If you do not have a permission you need you must request it



```
if (ActivityCompat.shouldShowRequestPermissionRationale(this,
        Manifest.permission.READ_CONTACTS)) {

    // Show an explanation to the user (likely with a
    // dialog) *asynchronously* -- don't block
    // this thread waiting for the user's response! After the user
    // sees the explanation, try again to request the permission.
```

- shouldShowRequestPermissionRationale returns true if the app previously requested the permission and the user denied it

Requesting Permission

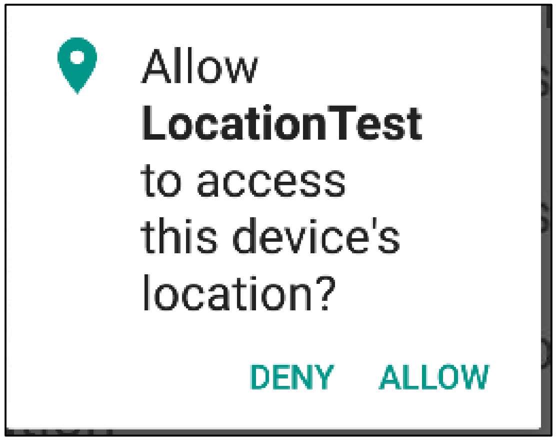
- If user has not previously denied permission

```
} else {
    // No explanation needed, we can request the permission.
    ActivityCompat.requestPermissions(this,
            new String[]{
                    Manifest.permission.ACCESS_COARSE_LOCATION,
                    Manifest.permission.ACCESS_FINE_LOCATION},
            MY_PERMISSIONS_REQUEST_GET_LOCATION);

    // MY_PERMISSIONS_REQUEST_GET_LOCATION is an
    // app-defined int constant. The callback method gets the
    // result of the request.
}
```

Dialog Result

- Dialog displayed to user. Lists permission group, not individual permissions
- Will lead to call back



Allow **LocationTest** to access this device's location?

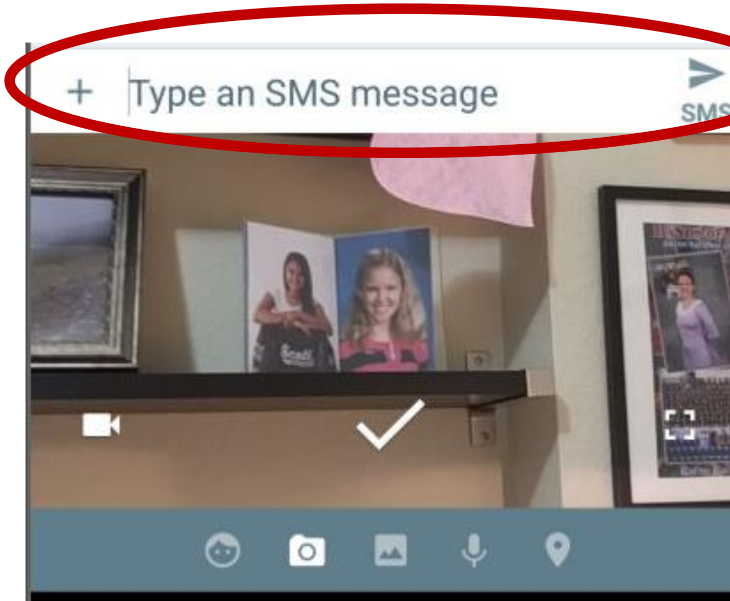
```java
public void onRequestPermissionsResult(int requestCode,
                                       String permissions[],
                                       int[] grantResults) {
    log("onRequestPermissionsResult");
    Log.d(TAG, "onRequestPermissionsResult");
    Log.d(TAG, "permissions: " + Arrays.toString(permissions));
    Log.d(TAG, "results: " + Arrays.toString(grantResults));
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_GET_LOCATION: {
            // If request is cancelled, the result arrays are empty.
            locationPermissionsGranted =
                    (grantResults.length > 0)
                            && (grantResults[0]
                            == PackageManager.PERMISSION_GRANTED);
```

Permission Groups

- Dialog shows Permission Group based on requested permission
- If use grants permission for one permission in group …
- … app now has all permission in that group.
- If you check on different permission in group after one granted, it will be granted as well
- Permissions retained when app rerun

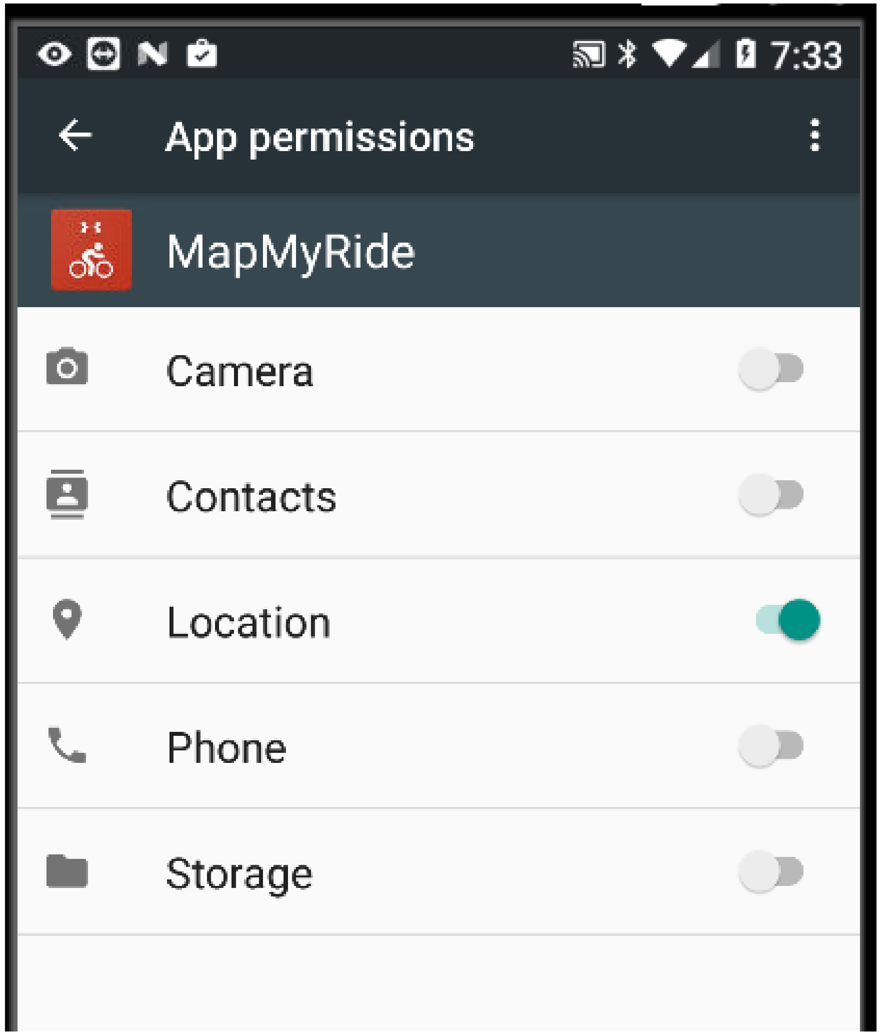
Permission Notes

- If you use an intent to accomplish something, then your app does not usually have to request permission.
- For example, if you use an Intent to take a picture, you DO NOT need CAMERA permission.
- Only if you want to access camera directly in your app.
- Example, messenger ->



Permission Notes

- A user can also alter permissions in the Settings app
- Settings -> app
- Toggle Switches to grant and deny permissions


7:33

App permissions

MapMyRide

Camera: off
Contacts: off
Location: on
Phone: off
Storage: off

The Support Library

- You may have noticed:

```
int permissionCoarse
        = ContextCompat.checkSelfPermission(this,
        Manifest.permission.ACCESS_COARSE_LOCATION);
```

```
ActivityCompat.requestPermissions(this,
```

- What are ContextCompat and ActivityCompat?

The Support Library

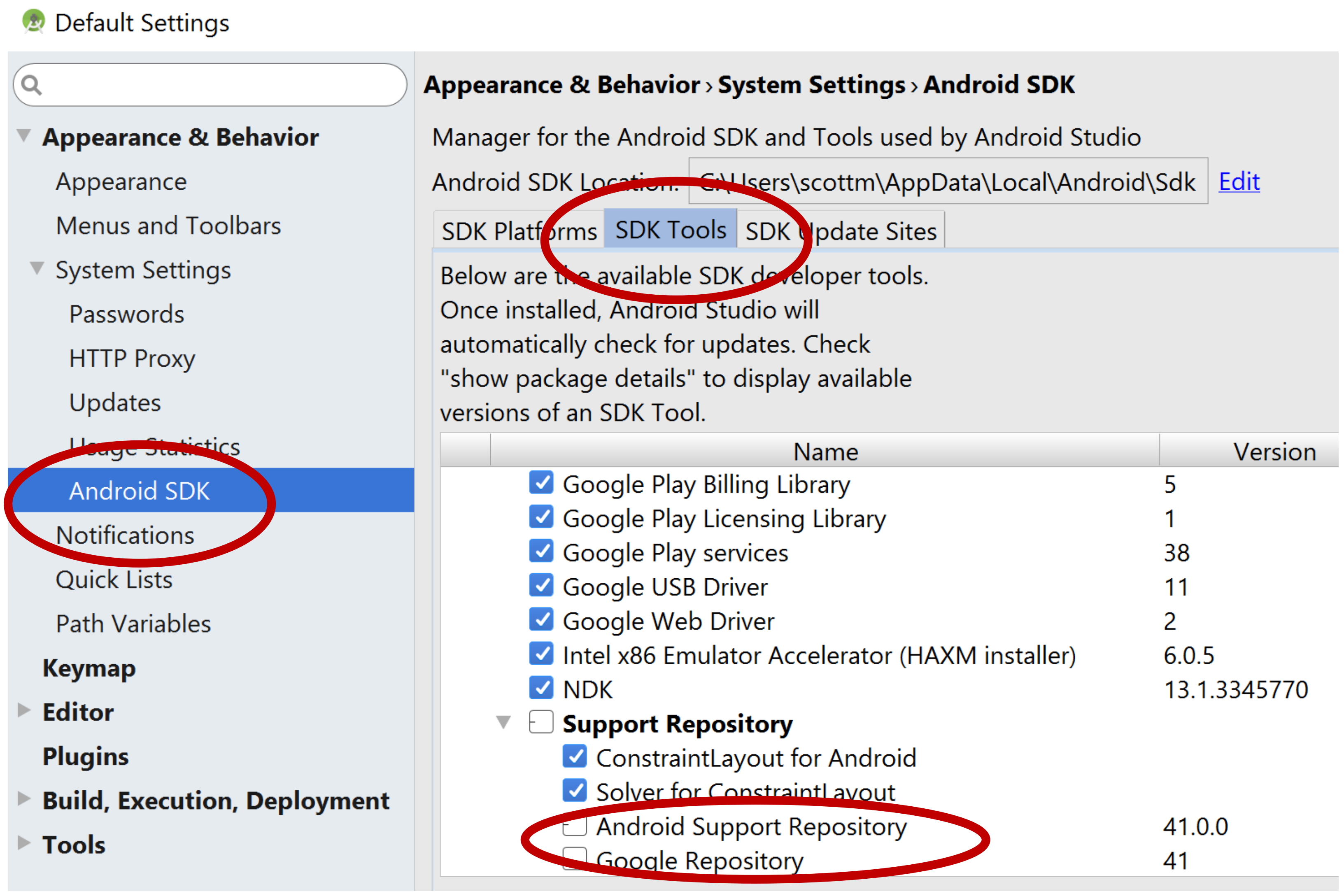
- We want to use newest features, but also allow app to run on older versions of Android
- Android provides libraries to provide newer functionality on older devices
- Mimics newer features with code built on older versions
  - Example: app bar using older widgets

The Support Library

- Previously a single library
- Now multiple libraries
- Min API levels for most of the official libraries (as fall 2016) is API level 9, Gingerbread

Support Library Setup

- Must download via SDK Manager



Default Settings

Appearance & Behavior
- Appearance
- Menus and Toolbars
- System Settings
  - Passwords
  - HTTP Proxy
  - Updates
  - Usage Statistics
  - Android SDK
- Notifications
- Quick Lists
- Path Variables

Keymap
Editor
Plugins
Build, Execution, Deployment
Tools

Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\scottm\AppData\Local\Android\Sdk Edit

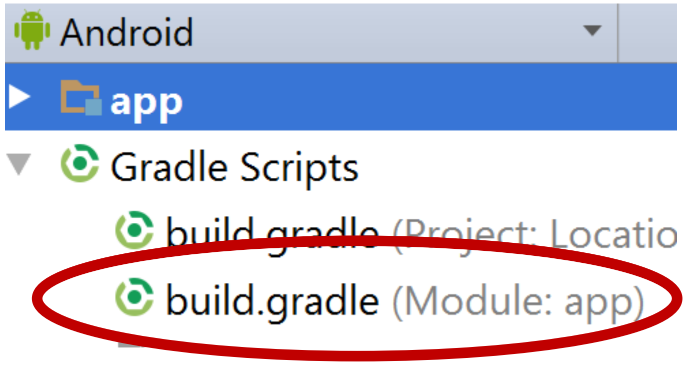
SDK Platforms | SDK Tools | SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

	Name	Version
✓	Google Play Billing Library	5
✓	Google Play Licensing Library	1
✓	Google Play services	38
✓	Google USB Driver	11
✓	Google Web Driver	2
✓	Intel x86 Emulator Accelerator (HAXM installer)	6.0.5
✓	NDK	13.1.3345770
☐	**Support Repository**	
✓	ConstraintLayout for Android	
✓	Solver for ConstraintLayout	
☐	Android Support Repository	41.0.0
☐	Google Repository	41

Support Library Setup

- Must add dependency to gradle build file
- build.gradle file for **<u>app</u>** NOT project



- Sample dependency:

```
dependencies {
    compile 'com.android.support:appcompat-v7:23.4.0'
```

- Sample import:

```
import android.support.v4.content.ContextCompat;
import android.support.v7.app.AppCompatActivity;
```

Support Library References

- Features:
- https://developer.android.com/topic/libraries/support-library/features.html
- Versions:
- https://developer.android.com/topic/libraries/support-library/packages.html
- Sample Documentation:
- https://developer.android.com/reference/android/support/v4/app/package-summary.html
- Release notes:
- https://developer.android.com/topic/libraries/support-library/revisions.html
- Design Patterns
- https://material.io/guidelines/patterns/permissions.html#