Android 6.0 Marshmallow

GDG DevFest
2015 Season

Who am I?!

Deuphil Kaufmann

@deubaka

SDK Developer @ HyperGrowth

PADC Co-organizer

Java/Android/Mobile

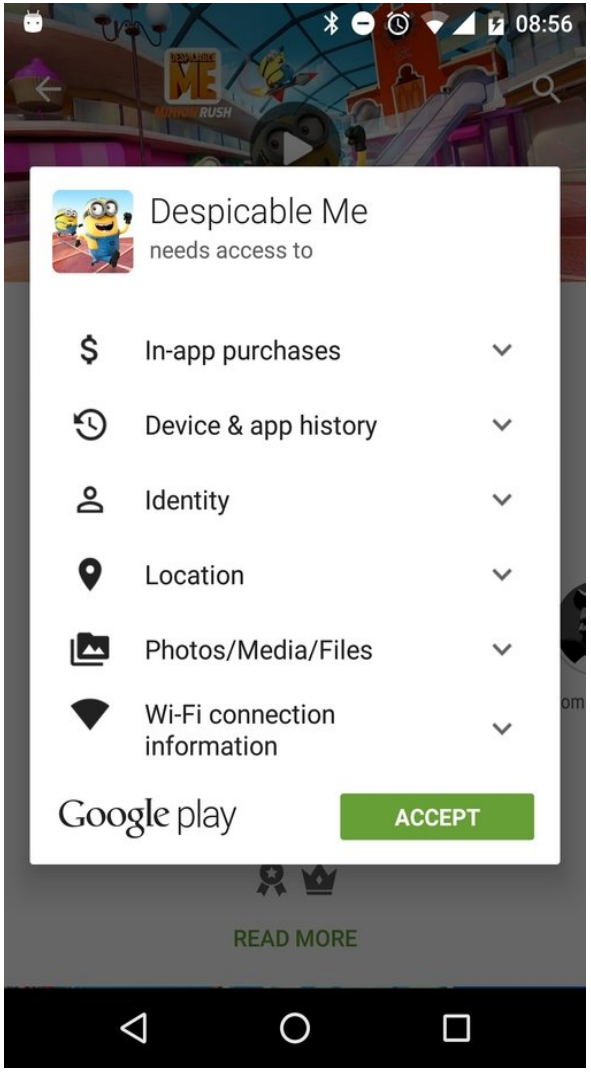
What we'll cover

- What's new on Android 6.0 Marshmallow (~15mins)
- Code Lab (~60mins)
  - New App Permission Model
  - Android Auto Backup*
- Q&A (~15mins)
- ???

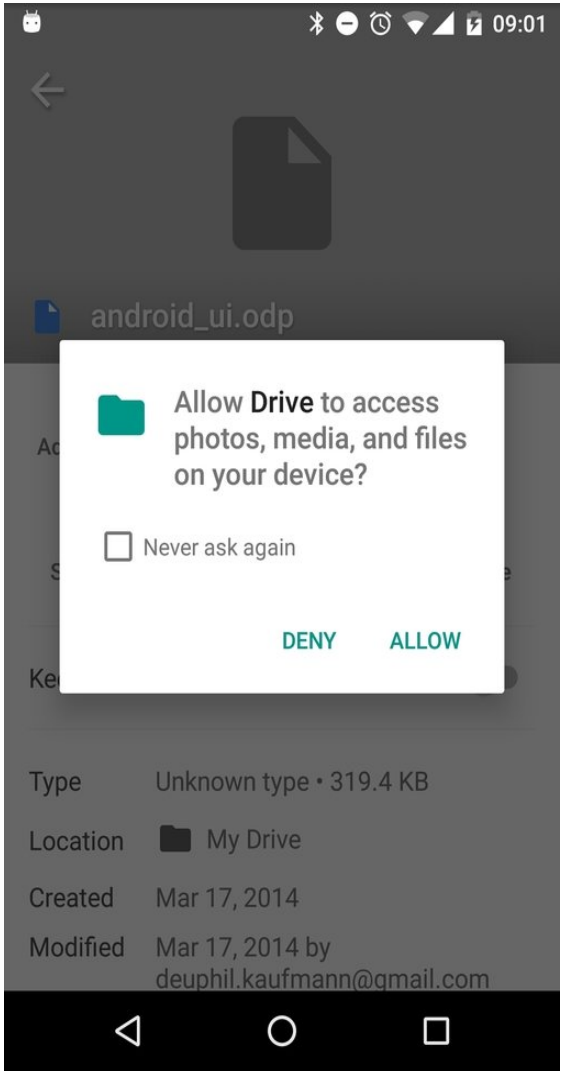
So what's new on Android 6.0?

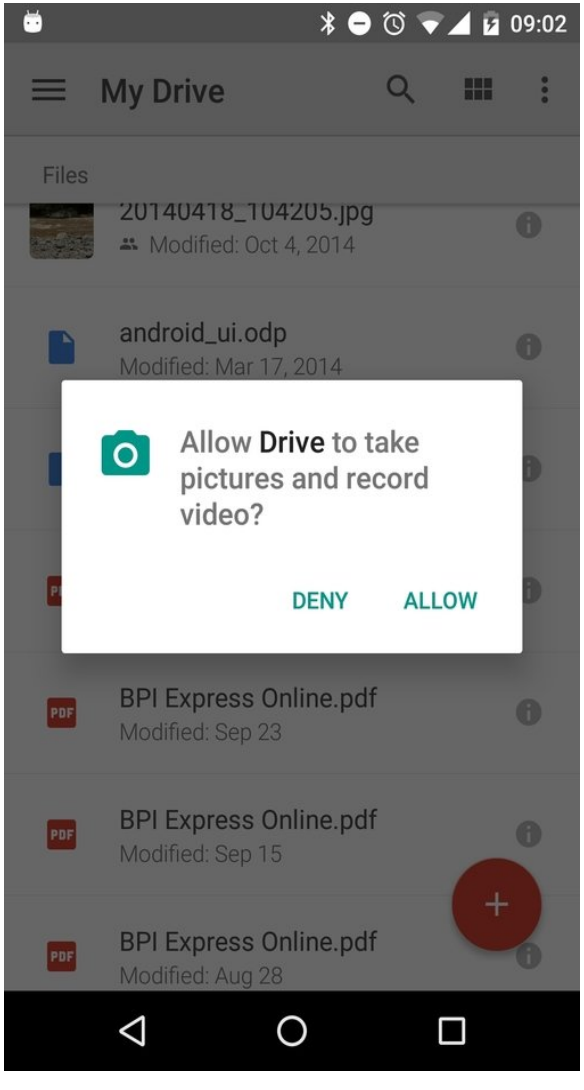
App Permissions

Then: Install-time permissions

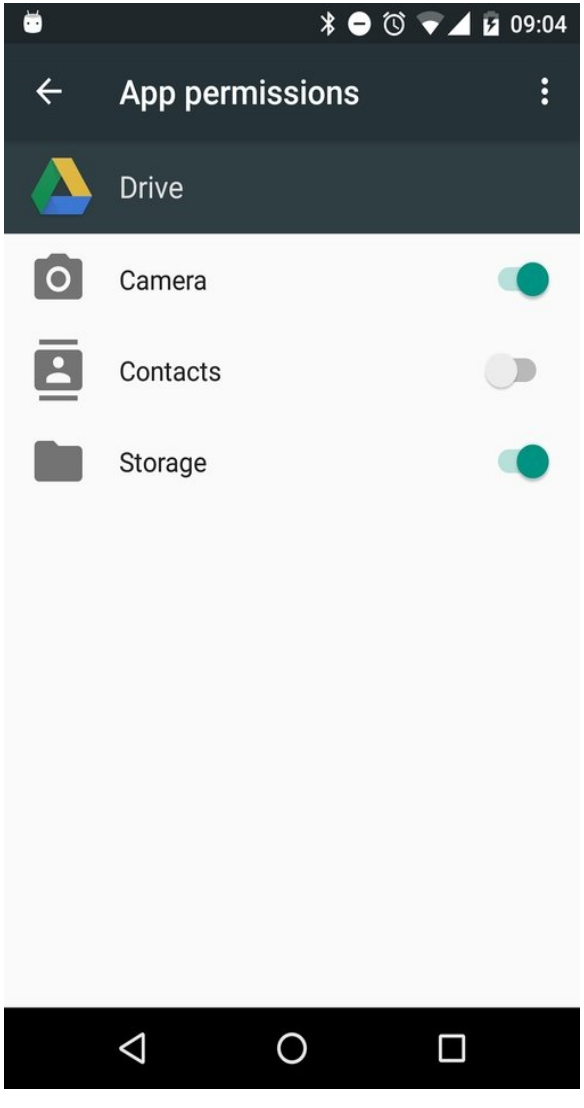


Now: Runtime permissions





User has full control of permissions



Fingerprint APIs

Two APIs for fingerprints

- **FingerprintManager.authenticate()**
  - Verify that authorized user is present
  - Your app controls all UI
- **KeyguardManager.createConfirmDeviceCredentialIntent()**
  - Present lock screen to user
  - startActivityForResult(), check for Activity.RESULT_OK
- **New permission**
  - android.permission.USE_FINGERPRINT

Android Backup

Finally!

- ***All* data are now backed up by default on apps targeting API 23+ and/or are on Android 6.0**
  - Up to 25MB per app
- **Developer control via optional scheme file in *xml/* resource**
  - includes/exlcludes

Power and Doze

Improvements

- **Better screen-off battery life**
- **Doze**
  - Untouched or idle devices will become "inactive"
  - Upon user interaction detected, will resume to normal operation
  - Longer to wakeup background tasks
- **App standby**
  - Unused apps lose network access
  - Will restore access upon user interaction
- **AlarmManager API changes**
  - setAndAllowWhileIdle *instead of* set
  - setExactAndAllowWhileIdle *instead of* setExact

Assistant Support

Assist API (Now on Tap)

- **Offers app developers to summon an assistant, Now on Tap**
- **Developers can provide additional contextual info on top**
  1. Implement Application.OnProvideAssistDataListener
  2. Register via registerOnProvideAssistDataListener()
  3. Override onProvideAssistData() and onProvideAssistContent()

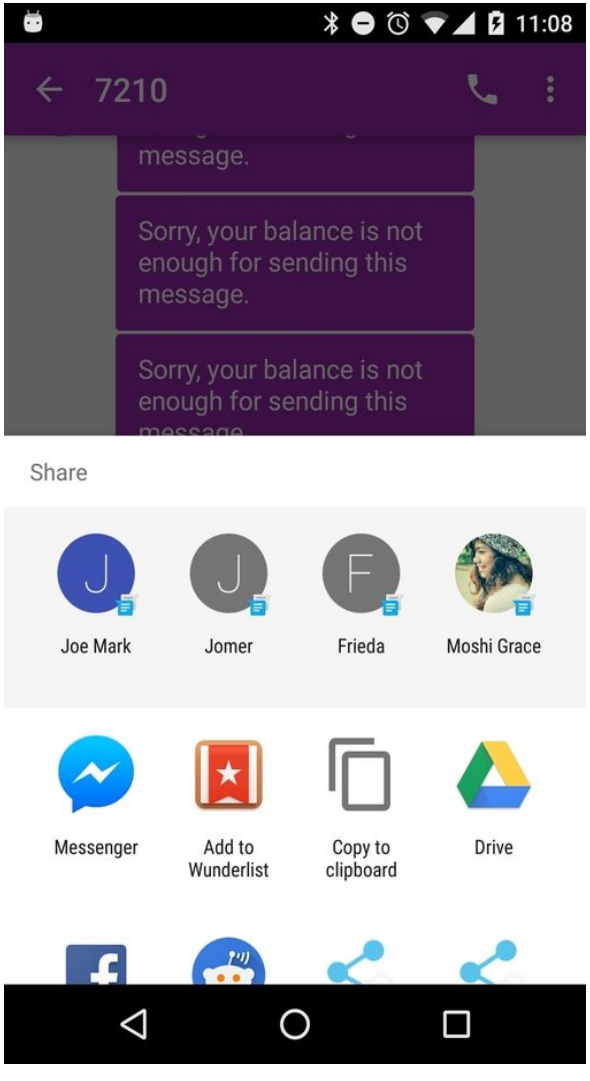
App Linking

Removing ambiguity

- **Enhances the current intent handling/resolution**
- **User can override app associations in the Settings**
- **Now able to create a direct link to your app's web domain**
  1. Declare your intent filters in your AndroidManifest.xml to handle your website's URIs
  2. Setup app link verification, autoVerify="true"
  3. Provide a digital asset links JSON on your website for verification
- **Gotchas**
  - Verification will be done on HTTPS
  - Only when all app link patterns have been verified will your app be the default handler

Direct Share

Contextual, intuitive sharing



Contextual, intuitive sharing

- Extend *ChooserTargetService* class, a Service
- Define in your *AndroidManifest.xml*
  1. Declare android.permission.BIND_CHOOSER_TARGET_SERVICE permission
  2. Add android.service.chooser.ChooserTargetService action in your <intent-filter>
- For each activity that you would want exposed, declare a <meta-data>

```
<meta-data
    android:name="android.service.chooser.chooser_target_service"
    android:value="<your ChooserTargetService class>" />
```

Other Stuff

Other stuff

- Adoptable Storage Devices
- Confirm Credential
- Bluetooth Stylus Support
- Improved BLE Scanning
- Hotspot 2.0 Release 1 Support
- 4K Display Mode
- Themeable ColorStateLists
- Audio + Video features
- Camera APIs
- Android for Work
- Android Data Binding*
- Google Play Services*
- Android Design Library (see Material Design Code Lab)

End

References and further readings

- Android 6.0 Marshmallow