Android VIII

JobDispatchers, Broadcast Receivers
Scheduling repeating jobs with conditions

- A Job is a scheduled service
- The old way: create an alarm and manually check all of the above conditions
JobScheduler and FirebaseJobDispatcher

- JobScheduler solves this by allowing the setting of constraints
- FirebaseJobDispatcher is a way to include earlier versions of Android

<table>
<thead>
<tr>
<th>Class</th>
<th>Compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>JobScheduler</td>
<td>API 21</td>
</tr>
<tr>
<td>FirebaseJobDispatcher</td>
<td>API 9</td>
</tr>
</tbody>
</table>
Google play services

● Google Play Services is an app
● A collection of Services to leverage Google products
  ○ Places API
  ○ Google Sign in
  ○ FirebaseJobDispatcher
● Distributed with most phones
Steps to Schedule a JobService

- Add the Gradle dependency
- Create a new task in our ReminderTasks
- Create a new Service that extends from JobService
- Add the JobService to the manifest
- Schedule with FirebaseJobDispatcher
BUILDING A JOB

- Get a GooglePlayDriver
- Get a dispatcher
- Build a job
  - Set attributes of job
- Schedule with
  
  dispatcher.schedule(myJob);
Broadcast Receivers

- Android components that listen for broadcasts via intent filters
- Have a callback method to implement: `onReceive(Context c, Intent i)`
Some system-wide broadcasts

android.intent.action.SCREEN_ON
android.intent.action.MEDIA_MOUNTED
android.intent.action.HEADSET_PLUG
android.intent.action.BATTERY_LOW
android.intent.action.DOWNLOAD_COMPLETE
android.media.AUDIO_BECOMING_NOISY
Other things than broadcast receivers have intent filters

```xml
<activity
    android:name=".MainActivity"
    android:launchMode="singleTop"
    android:screenOrientation="portrait">
  <intent-filter>
    <action android:name="android.intent.action.MAIN"/>
    <category android:name="android.intent.category.LAUNCHER"/>
  </intent-filter>
</activity>
```
Static and Dynamic Broadcast Receivers

**Static**
Will receive and execute even when app is not started

**Dynamic**
Executes only when app is started
Registering a Static Broadcast Receiver

AndroidManifest.xml

```xml
<receiver android:name="NewPictureBroadcastReceiver">
    <intent-filter>
        <action android:name="com.android.camera.NEW_PICTURE" />
        <action android:name="android.hardware.action.NEW_PICTURE" />
        <data android:mimeType="image/*" />
    </intent-filter>
</receiver>
```
Creating a Broadcast Receiver

NewPictureBroadcastReceiver.java

```java
public class NewPictureBroadcastReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        //do something
    }
}
```
Static Broadcast Receiver anti-pattern

Take a picture, and a bunch of apps awake and do something, chewing up battery
Dynamic Receivers

Created the same way as static receivers, but are registered dynamically

```java
@Override
protected void onResume() {
    super.onResume(); IntentFilter receiverFilter = new IntentFilter(Intent.ACTION_HEADSET_PLUG);
    mReceiver = new HeadsetStateReceiver(); registerReceiver( receiver, receiverFilter );
}
@override
protected void onPause() {
    super.onPause();
    unregisterReceiver(mReciever);
}
```
Hydration Reminder: Show when Charging

- Method that changes the plug from grey to pink
- Intent filter for charging status
- Broadcast receiver updates color
- Register broadcast receiver with intent filter
- Cleanup broadcast receiver in onPause