ITW202: Mobile Application

Unit IV: Developing for Android

Ms. Sonam Wangmo

Gyalpozhing College of Information Technology
Royal University of Bhutan

May 26, 2021

Broadcasts

Broadcasts

Broadcasts are messages sent by Android system and other Android apps, when an event of interest occurs.
Broadcasts are wrapped in an Intent object. This Intent object's contains the event details such as, android.intent.action.HEADSET_PLUG, sent when a wired headset is plugged or unplugged.

Types of broadcasts

1. System broadcast: System broadcasts are delivered by the system.
2. Custom broadcast: Custom broadcasts are delivered by your app.

System broadcasts

System broadcast are the messages sent by the Android system, when a system event occurs, that might affect your app.

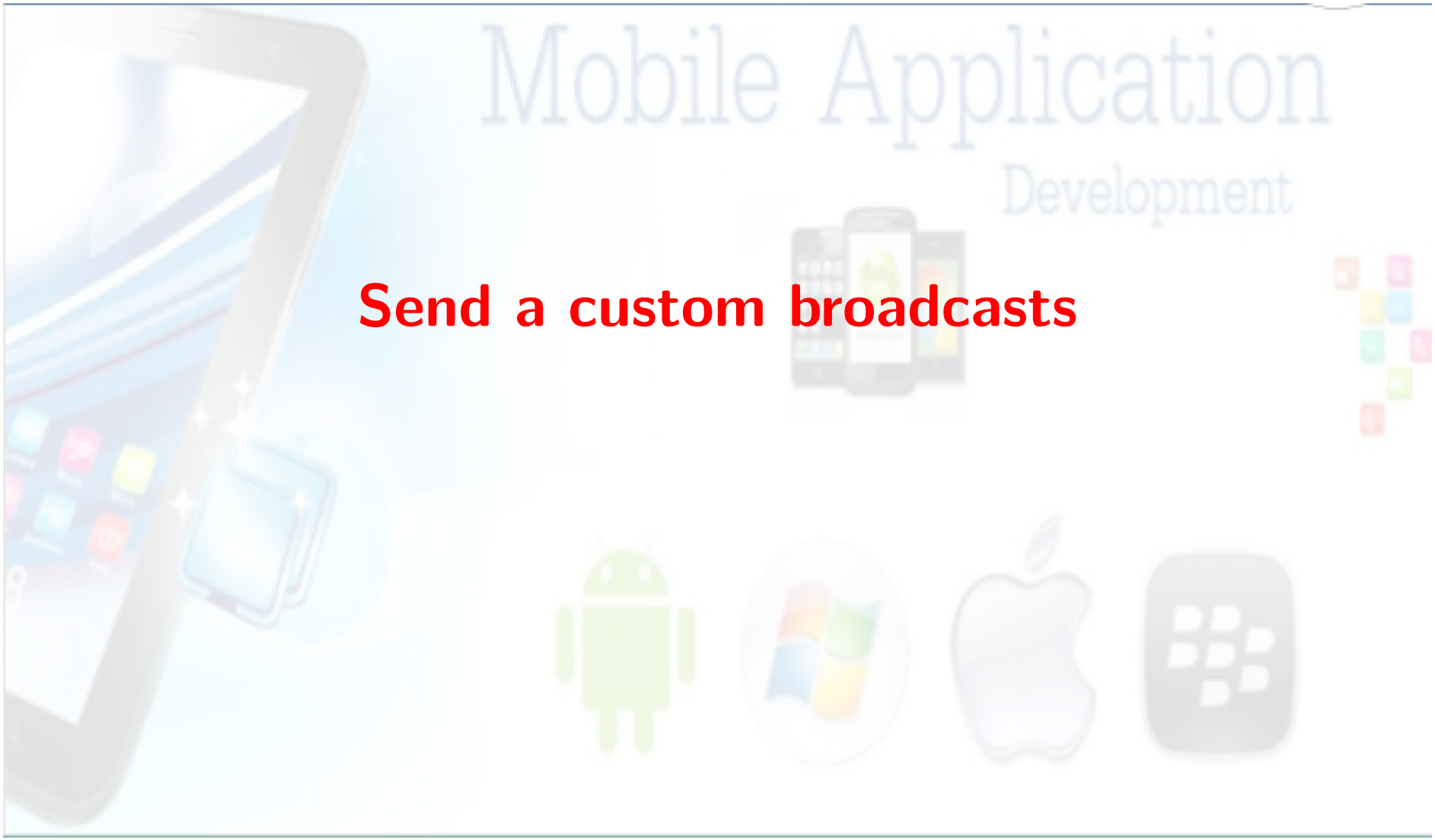
System broadcasts

Few examples:

- An Intent with action, ACTION_BOOT_COMPLETED is broadcasted when the device boots.
- An Intent with action, ACTION_POWER_CONNECTED is broadcasted when the device is connected to the external power.

Custom broadcasts

- Custom broadcasts are broadcasts that your app sends out, similar to the Android system.
- For example, when you want to let other app(s) know that some data has been downloaded by your app, and its available for their use.



Send a custom broadcasts

Send a custom broadcast

There are three ways to deliver a custom broadcast:

- Ordered broadcast.
- Normal broadcast.
- Local broadcast

Ordered Broadcast

- Ordered broadcast is delivered to one receiver at a time.
- To send a ordered broadcast, use the sendOrderedBroadcast() method.
- Receivers can propagate result to the next receiver or even abort the broadcast.
- Control the broadcast order with android:priority attribute in the manifest file.

Normal Broadcast

- Delivered to all the registered receivers at the same time, in an undefined order.
- Most efficient way to send a broadcast.
- Receivers can't propagate the results among themselves, and they can't abort the broadcast.
- The sendBroadcast() method is used to send a normal broadcast.

Local Broadcast

- Sends broadcasts to receivers within your app.
- No security issues since no interprocess communication.

Local Broadcast

To send a local broadcast:

- To get an instance of LocalBroadcastManager, call getInstance() and pass in the application context.
- Call sendBroadcast() on the instance. Pass in the intent that you want to broadcast

```
LocalBroadcastManager.getInstance(this)
.sendBroadcast(customBroadcastIntent);
```

Custom Broadcast

- Sender and receiver must agree on unique name for intent (action name)
- Define in activity and broadcast receiver

```
private static final String
ACTION_CUSTOM_BROADCAST =
"com.example.android.powerreceiver
.ACTION_CUSTOM_BROADCAST";
```

Broadcast Receivers

What is a broadcast receiver?

- Broadcast receivers are app components.
- They register for various system broadcast and or custom broadcast.

What is a broadcast receiver?

- They are notified (via an Intent):
  - By the system, when an system event occurs that your app is registered for.
  - By another app, including your own if your app is registered for that custom event.

Register your broadcast receiver

Broadcast receivers can be registered in two ways:

- Static receivers: Registered in your AndroidManifest.xml, also called as Manifest-declared receivers.
- Dynamic receivers: Registered using app or activities' context in your Java files, also called as Context-registered receivers.

Receiving a system broadcast

Broadcast receivers can be registered in two ways:

- Starting from Android 8.0 (API level 26), static receivers can't receive most of the system broadcasts.
- Use a dynamic receiver to register for these broadcasts.
- If you register for the system broadcasts in the manifest, the Android system won't deliver them to your app.

Implementing Broadcast Receivers

To create a broadcast receiver

- Subclass the BroadcastReceiver class and override its onReceive() method.
- Register the broadcast receiver and specify the intent-filters:
  - Statically, in the Manifest.
  - Dynamically, with registerReceiver().

What are Intent-filters

Intent-filters specify the types of intents a broadcast receiver can receive. They filter the incoming intents based on the Intent values like action.
To add an intent-filter:

- To your AndroidManifest.xml file, use <intent-filter> tag.
- To your Java file use the IntentFilter object.

Subclass a broadcast receiver

In Android studio, File > New > Other > BroadcastReceiver

```
public class CustomReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // This method is called when the BroadcastReceiver
        //  is receiving an Intent broadcast.
        throw new UnsupportedOperationException("Not yet implemented");
    }
}
```

Implement onReceive()

Example implementation of onReceive() method which handles power connected and disconnected.

```
@Override
public void onReceive(Context context, Intent intent) {
    String intentAction = intent.getAction();
    switch (intentAction){
        case Intent.ACTION_POWER_CONNECTED:
            break;
        case Intent.ACTION_POWER_DISCONNECTED:
            break;
    }
```

Register statically in Android manifest

- `<receiver>` element inside `<application>` tag.
- `<intent-filter>` registers receiver for specific intents.

Register statically in Android manifest

```
<receiver
    android:name=".CustomReceiver"
    android:enabled="true"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
    </intent-filter>
</receiver>
```

Register dynamically

- Register your receiver in `onCreate()` or `onResume()`.

```
// Register the receiver using the activity context.
   this.registerReceiver(mReceiver, filter);
```

- Unregister in `onDestroy()` or `onPause()`.

```
// Unregister the receiver
   this.unregisterReceiver(mReceiver);
```

Register a Local broadcast receiver

Register local receivers dynamically, because static registration in the manifest is not possible for a local broadcasts.

Register a Local broadcast receiver

To register a receiver for local broadcasts:

- Get an instance of LocalBroadcastManager.
- Call registerReceiver().

```
LocalBroadcastManager.getInstance(this).registerReceiver
(mReceiver,
  new IntentFilter(CustomReceiver.ACTION_CUSTOM_BROADCAST));
```

Unregister a Local broadcast receiver

To unregister a local broadcast receiver:

- Get an instance of the LocalBroadcastManager.
- Call LocalBroadcastManager.unregisterReceiver().

Get an instance of the LocalBroadcastManager.
Call LocalBroadcastManager.unregisterReceiver().


Mobile Application
Development
THANK YOU