Mobile Application Development

Higher Diploma in Science in Computer Science

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit

IntentServices, BroadcastReceivers & Alarms

IntentServices

Intent Service

- The IntentService class provides a straightforward structure for running an operation on a single background thread.

- This allows it to handle long-running operations without affecting your user interface's responsiveness.

- Also, an IntentService isn't affected by most user interface lifecycle events, so it continues to run in circumstances that would shut down an AsyncTask

Limitations

- It can't interact directly with your user interface. To put its results in the UI, you have to send them to an Activity.

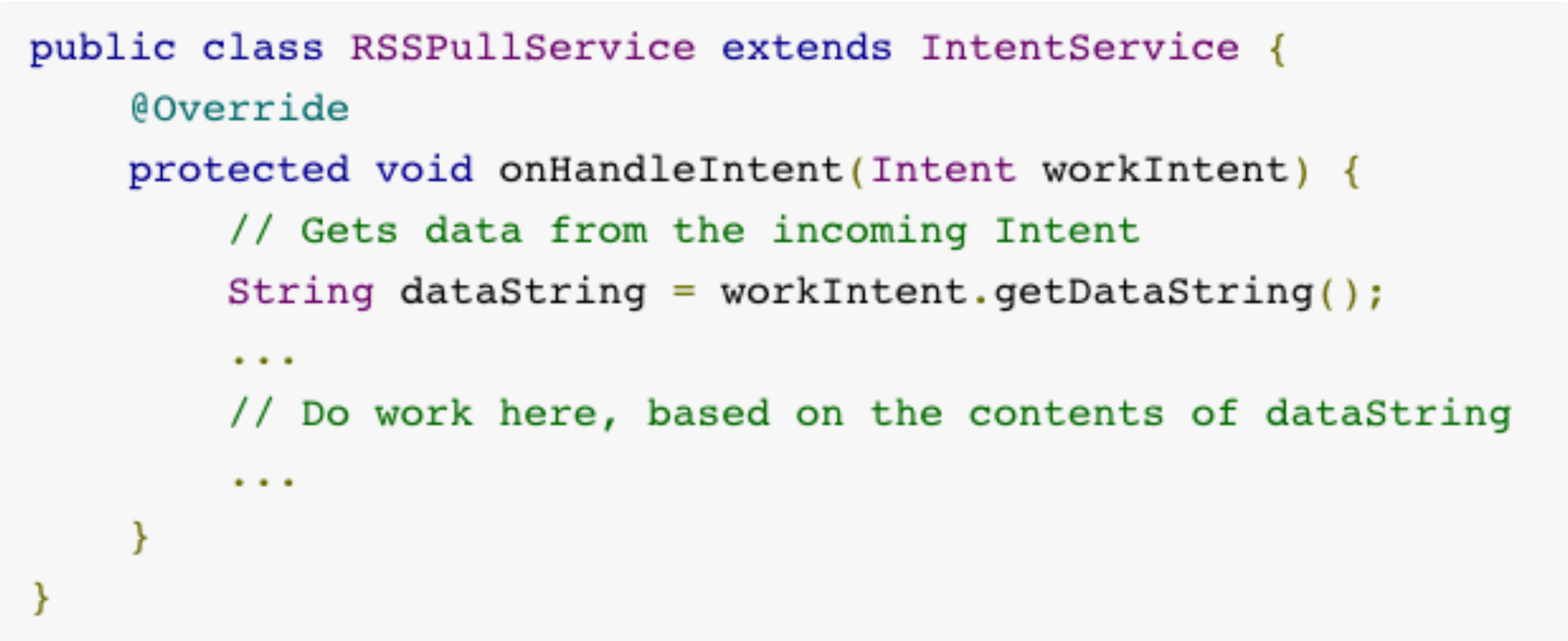
- Work requests run sequentially. If an operation is running in an IntentService, and you send it another request, the request waits until the first operation is finished.

- An operation running on an IntentService can't be interrupted.

- However, in most cases an IntentService is the preferred way to simple background operations.

Creating an IntentService

- To create an IntentService define a class that extends IntentService, and within it, define a method that overrides onHandleIntent().

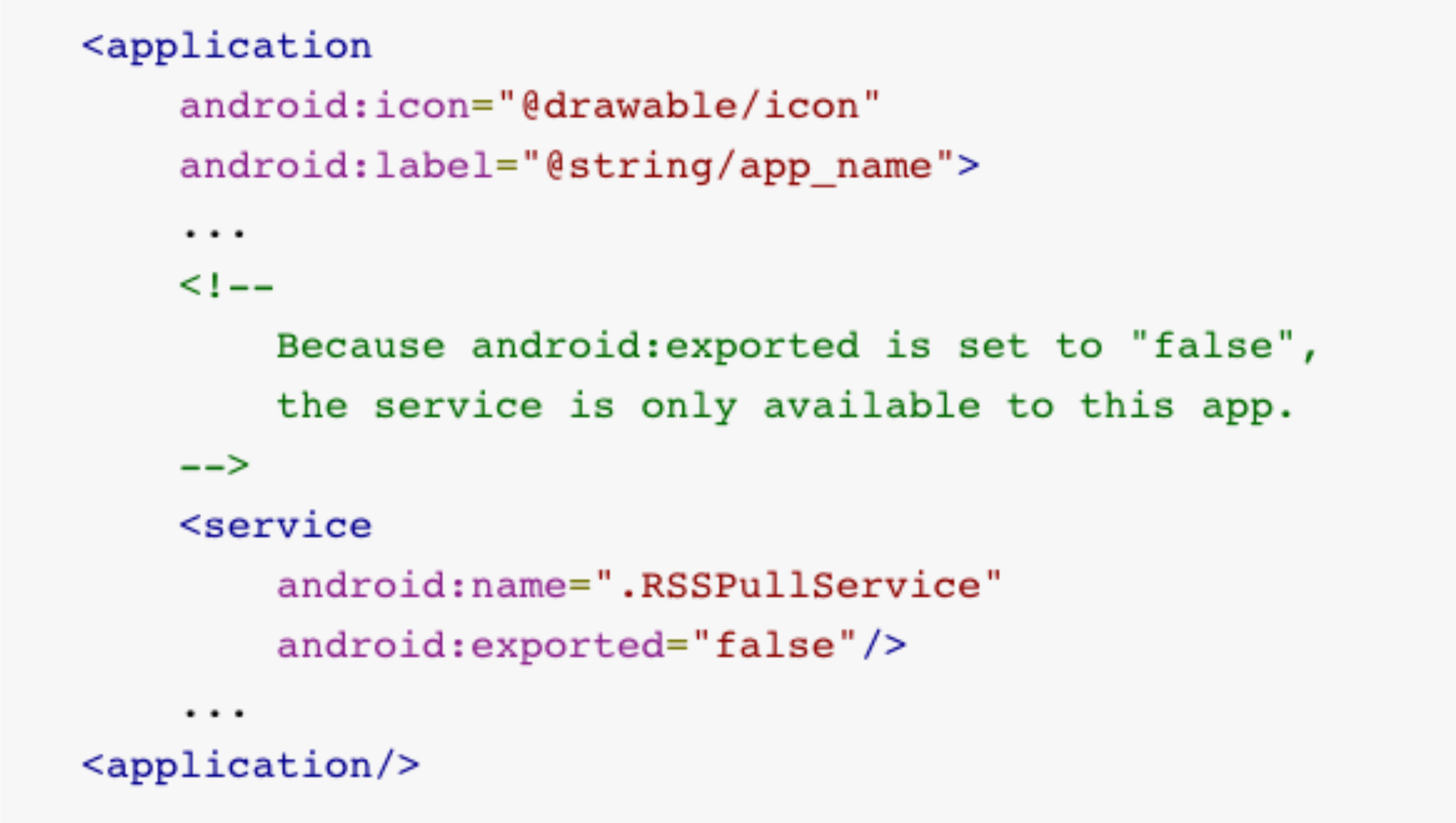


```
public class RSSPullService extends IntentService {
    @Override
    protected void onHandleIntent(Intent workIntent) {
        // Gets data from the incoming Intent
        String dataString = workIntent.getDataString();
        ...
        // Do work here, based on the contents of dataString
        ...
    }
}
```

Define the IntentService in the Manifest

- An IntentService also needs an entry in your application manifest.

- Provide this entry as a <service> element that's a child of the <application> element:

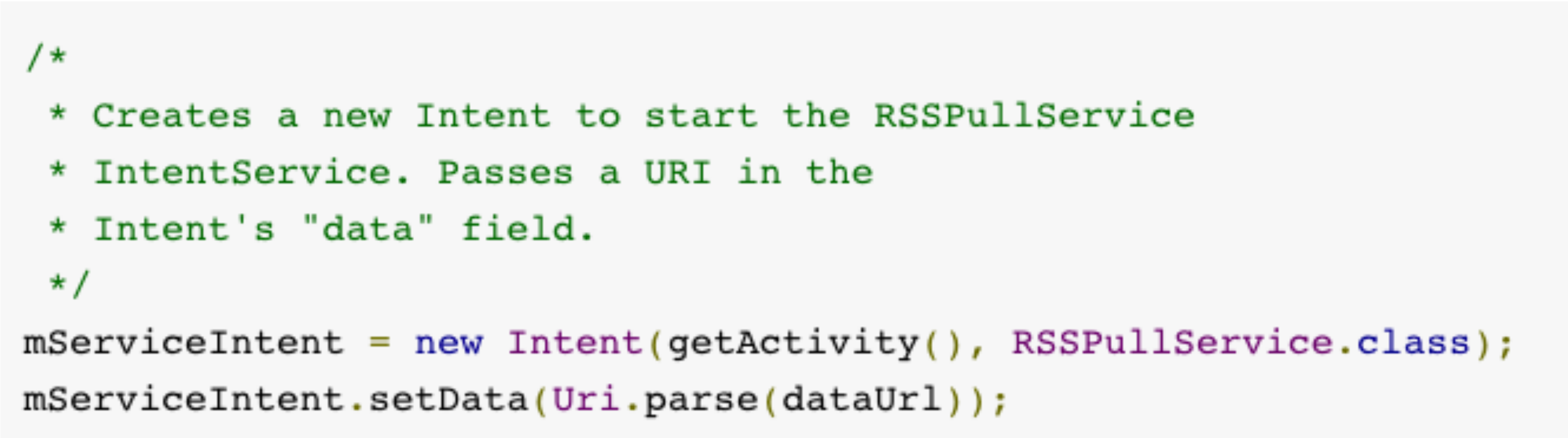


```
<application
    android:icon="@drawable/icon"
    android:label="@string/app_name">
    ...
    <!--
        Because android:exported is set to "false",
        the service is only available to this app.
    -->
    <service
        android:name=".RSSPullService"
        android:exported="false"/>
    ...
<application/>
```

- The attribute android:name specifies the class name of the IntentService.

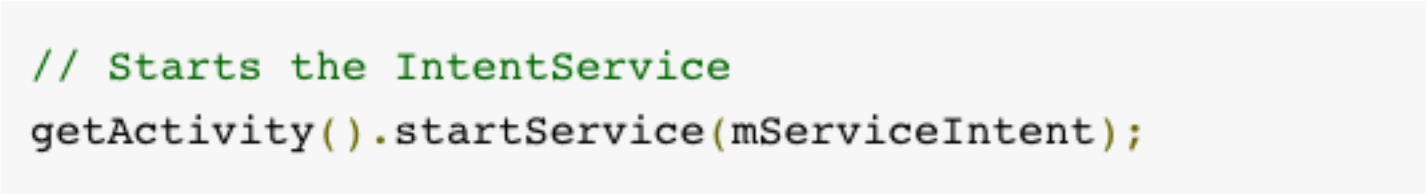
Creating an IntentService Request

- To create a work request and send it to an IntentService, create an explicit Intent, add work request data to it, and send it to IntentService by calling startService().



```
/*
 * Creates a new Intent to start the RSSPullService
 * IntentService. Passes a URI in the
 * Intent's "data" field.
 */
mServiceIntent = new Intent(getActivity(), RSSPullService.class);
mServiceIntent.setData(Uri.parse(dataUrl));
```

Starting the IntentSerivce



```
// Starts the IntentService
getActivity().startService(mServiceIntent);
```

- Notice that you can send the work request from anywhere in an Activity or Fragment. For example, if you need to get user input first, you can send the request from a callback that responds to a button click or similar gesture.

- Once you call startService(), the IntentService does the work defined in its onHandleIntent() method, and then stops itself.

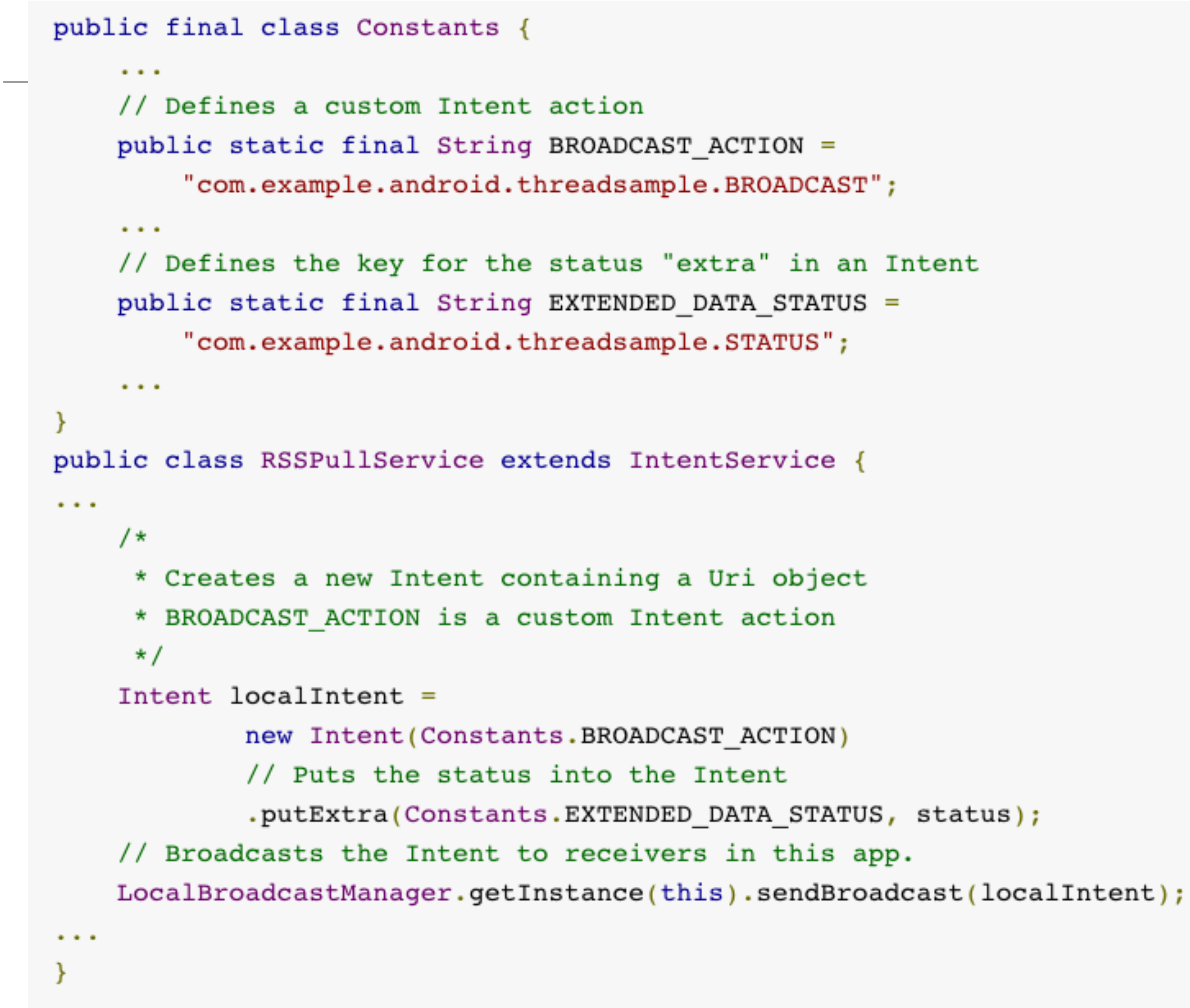
Broadcast Receivers

Report Status From an IntentService

- To send the status of a work request in an IntentService to other components, first create an Intent that contains the status in its extended data

- The send the Intent by calling LocalBroadcastManager.sendBroadcast().

- This sends the Intent to any component registered to receive it.

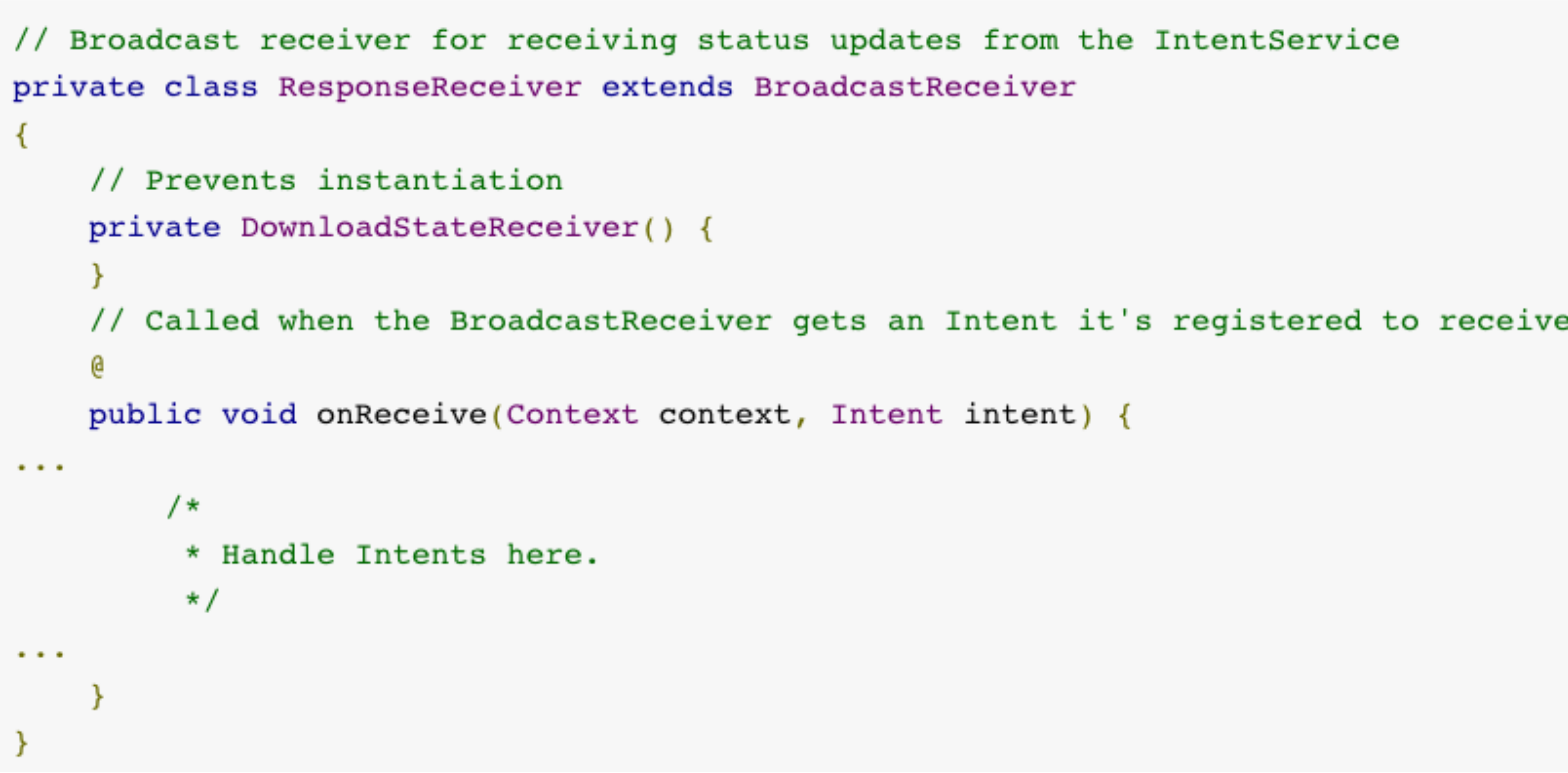


```
public final class Constants {
    ...
    // Defines a custom Intent action
    public static final String BROADCAST_ACTION =
        "com.example.android.threadsample.BROADCAST";
    ...
    // Defines the key for the status "extra" in an Intent
    public static final String EXTENDED_DATA_STATUS =
        "com.example.android.threadsample.STATUS";
    ...
}
public class RSSPullService extends IntentService {
...
    /*
     * Creates a new Intent containing a Uri object
     * BROADCAST_ACTION is a custom Intent action
     */
    Intent localIntent =
            new Intent(Constants.BROADCAST_ACTION)
            // Puts the status into the Intent
            .putExtra(Constants.EXTENDED_DATA_STATUS, status);
    // Broadcasts the Intent to receivers in this app.
    LocalBroadcastManager.getInstance(this).sendBroadcast(localIntent);
...
}
```

Receive Status Broadcasts from an IntentService

To receive broadcast Intent objects, use a subclass of BroadcastReceiver.

In the subclass, implement the BroadcastReceiver.onReceive() callback method, which LocalBroadcastManager invokes when it receives an Intent..



```
// Broadcast receiver for receiving status updates from the IntentService
private class ResponseReceiver extends BroadcastReceiver
{
    // Prevents instantiation
    private DownloadStateReceiver() {
    }
    // Called when the BroadcastReceiver gets an Intent it's registered to receive
    @
    public void onReceive(Context context, Intent intent) {
...
        /*
         * Handle Intents here.
         */
...
    }
}
```

Scheduling Repeating Alarms

- Alarms (based on the AlarmManager class) give you a way to perform time-based operations outside the lifetime of your application. For example, you could use an alarm to initiate a long-running operation, such as starting a service once a day to download a weather forecast.

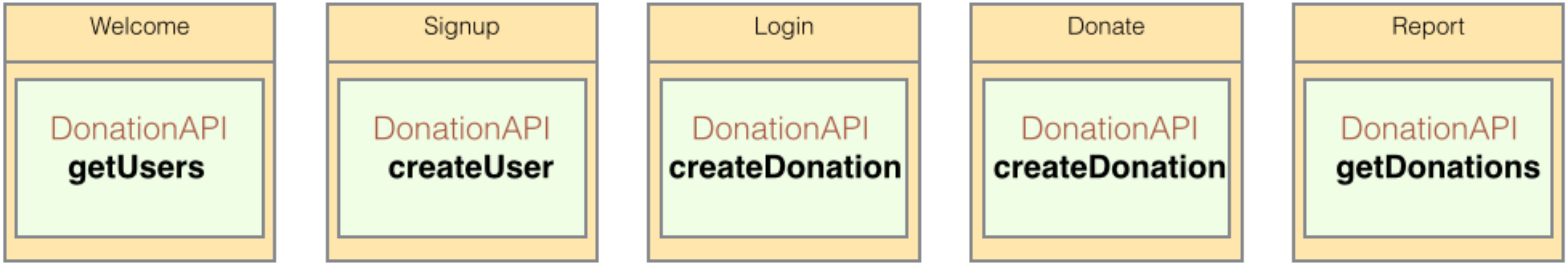
Alarm Characteristics

- They let you fire Intents at set times and/or intervals.
- You can use them in conjunction with broadcast receivers to start services and perform other operations.
- They operate outside of your application, so you can use them to trigger events or actions even when your app is not running, and even if the device itself is asleep.
- They help you to minimize your app's resource requirements. You can schedule operations without relying on timers or continuously running background services.

Set a Repeating Alarm

Donation-Android-V6

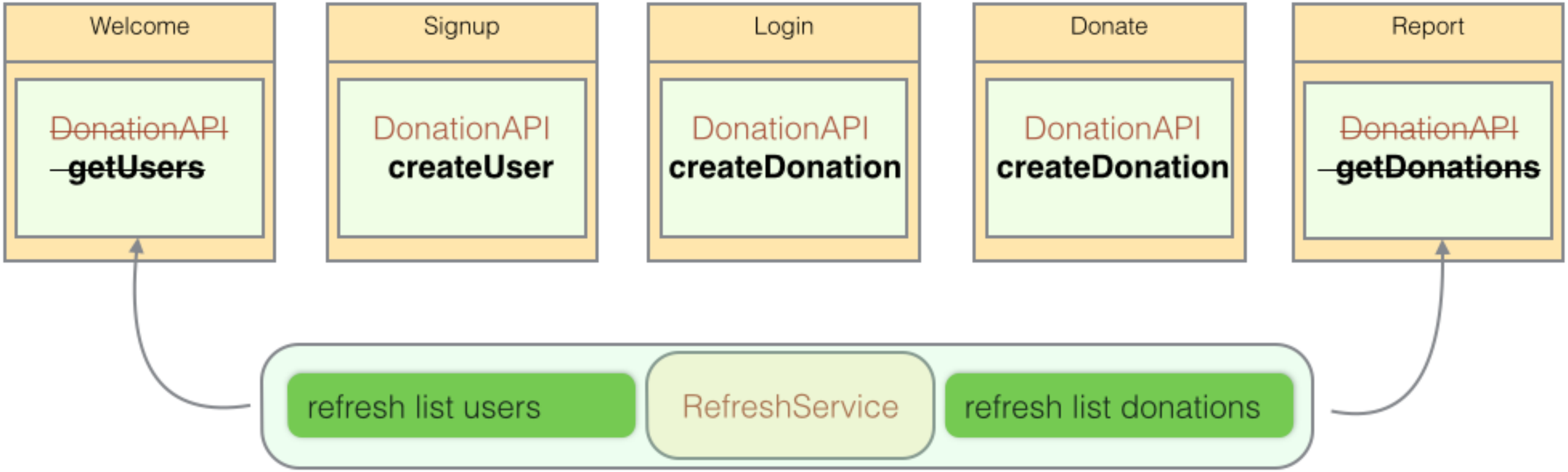
All network calls on AsyncTask thread



- What happens when tweets are posted to Donation-Service-Play from other users?

Donation-Android-V7

IntentService employed for R(ead)



- Explore use of the IntentService by running a background service to periodically read updates from the donation-service-play API

Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see http://creativecommons.org/licenses/by-nc/3.0/

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit