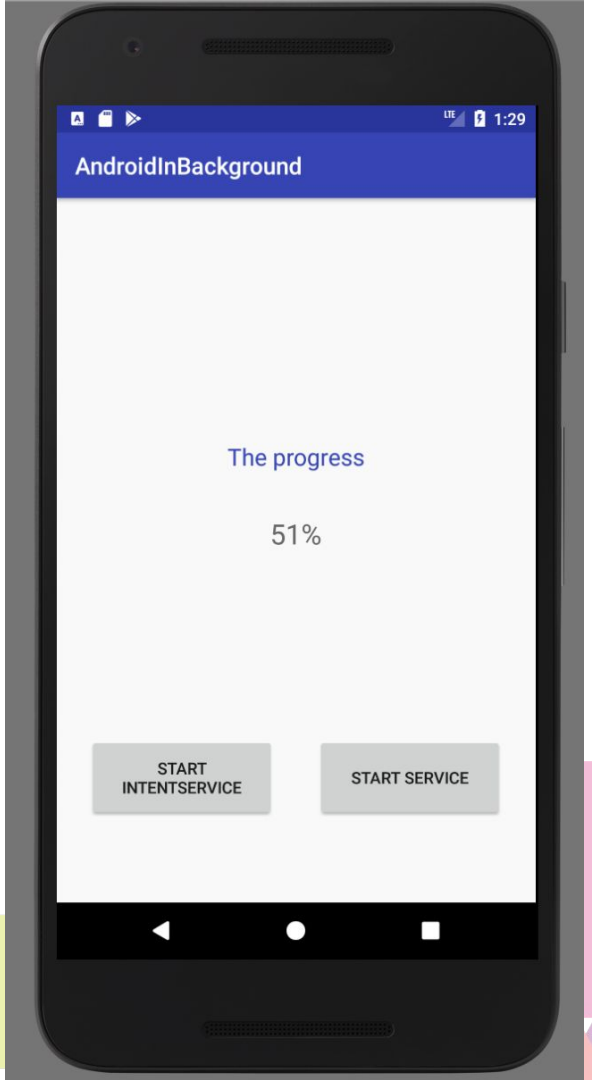
Exercise session 8

03.09.2018

Overview (1 / 3)

We'll will create simple app that will start and use two Services to do very hard job. (at least we will pretend That we are doing very hard job)



Overview (2 / 3)

- Clicking on start service will start a service,
  It will do some very hard job, it will update with a
  Progress and once it done it will show “Done!”.

- Clicking on start intent service will start an intent service,
  It will do some very hard job, it will update with a
  Progress and once it done it will show “Done!”.

- By clicking on start service while other service is running
  Will stop the already running service and will start a new one

Overview (3 / 3)

- We are going to use HandlerThread & Looper as it learned in Lecture #4 and we will see how we can use with a regular service.
- You going to see for the first time BroadcastReceiver and we will explain it more in upcoming lesson.

Android Background Services

You may want to use a background service when you are developing some feature that doesn't requires a user interface or interaction, maybe running a continuous service that check a server status, downloading a file, installing android packages, etc.

Android provides two types of services: `IntentService` and `Service`.

Service is the base class for any Android background service. This class provides you some methods to execute and kill your background service. It runs on your main thread, so it's recommended to start a new thread to avoid UI blocking.

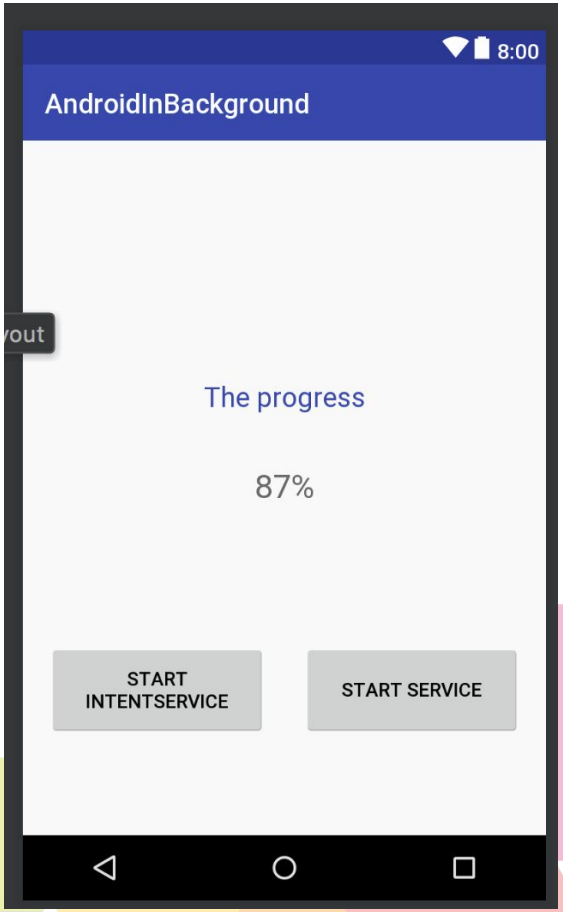
IntentService is a simpler Service that already runs in a separate thread and self destroy after processing everything.

Let's implement the two services and see it in action.

Step1- Add Views to Activity layout

Create simple activity that called MainActivity. UI pretty simple.

Two buttons & two text views.



Step 2- Add ClickListeners to the buttons

```
@Override protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //TODO find reference to progress TextView

    //TODO Add listeners for two buttons

    subscribeForProgressUpdates();
}
```

Step 3- Implement start service for each of the button

For every button implement start service in the activity

```
@Override public void onClick(View v) {
  //TODO implement clicks on two buttons
}
```

Step 4- Go to HardJobIntentService

Check the code and implement //TODO

```
@Override protected void onHandleIntent(@Nullable Intent intent) {
  isDestroyed = false;
  showToast("Starting IntentService");
  try {
   for (int i = 0; i <= 100 && !isDestroyed; i++) {        Thread.sleep(100);
    //TODO call for notifyUI method to pass progress to UI
   }
  } catch (InterruptedException e) {
   Thread.currentThread().interrupt();
  }
  showToast("Finishing IntentService");
 }
```

Step 5 - Go to HardJobService

```
@Override public void onCreate() {
    // To avoid cpu-blocking, we create a background handler to run our service
    //TODO Create HandlerThread
    // start the new handler thread
    //TODO Start a created HandlerThread
    //TODO Get looper out of thread
    // start the service using the background handler
    //TODO Create instance of ServiceHandler class that receives instance of ServiceLooper
}
```

Step 6 - Implement onStartCommand()

```
@Override public int onStartCommand(Intent intent, int flags, int startId) {
  isDestroyed = false;
  Toast.makeText(this, "onStartCommand", Toast.LENGTH_SHORT).show();
  // call a new service handler. The service ID can be used to identify the service
  Message message = mServiceHandler.obtainMessage();
  message.arg1 = startId;
  //TODO Send message to SericeHandler

  //TODO Return START_STICKY
}
```

Step 7 - Run the project

Run :)

DONE?

That’s amazing! Good for you!!

If you have even a small question- don’t forget to ask the mentors:

At the class or on Slack.

Kudos!!

You’re all done with exercise 8!

See you next time!