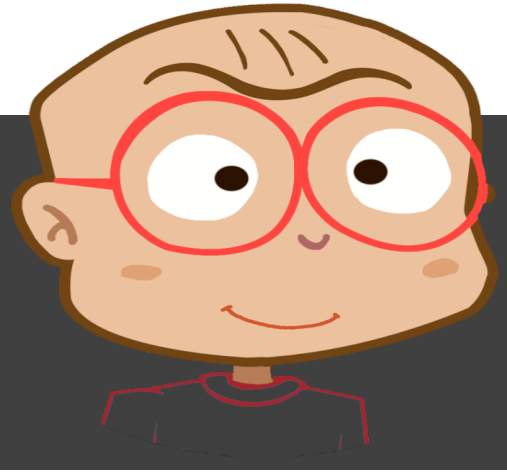


Services

START

Learning Goals

After the course, attendees will be able to:

- Understanding about Service component in Android
- Understanding how to implement services in Android

What is Service?

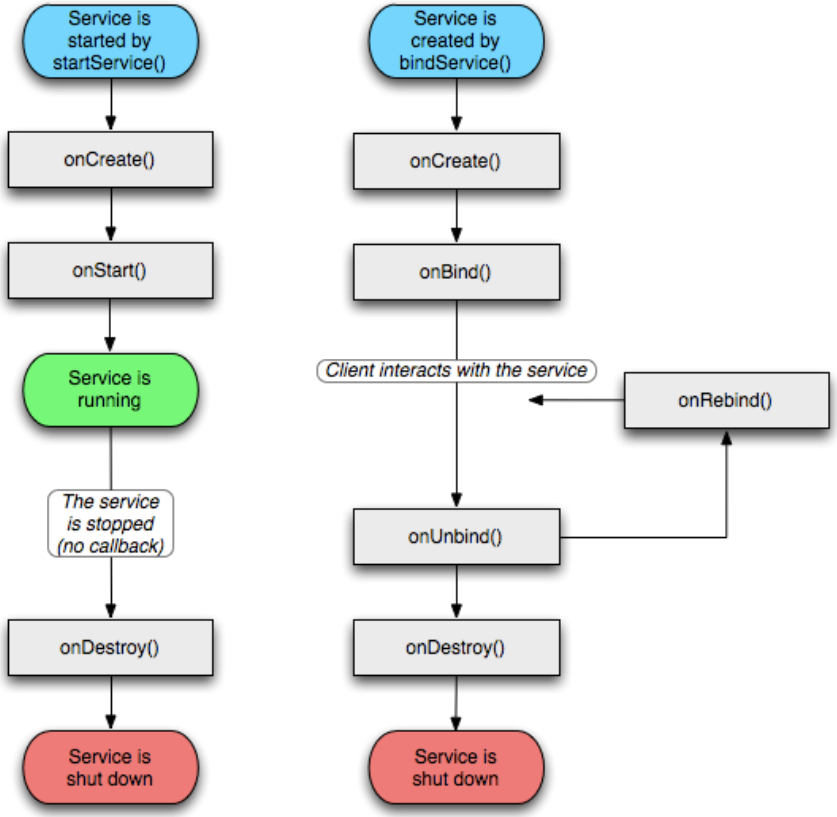
- Service is an Android component which runs in the same process as the application it is part of.
- Service used to execute tasks with no UI.
- Service can perform long-running operations in the background

Service Types

There are 2 types of Service

- **Started service** : is "started" when an application component (such as an activity) starts it by calling startService(). run in the background indefinitely, even if the component that started it is destroyed.
- **Bound Service :** is "bound" when an application component binds to it by calling bindService() run at the lifespan of the calling activities

Service Life cycle



Started Services

There are 2 ways to implement of Started Service

- **Extending the IntentService class** : uses a worker thread to handle all start requests, one at a time.
- **Extending the Service class :** uses caller's main thread, by default, need to create a new thread in which to do all the service's work.

Started Services - Extending IntentService

To make Intent Service work, all we need to do is implement onHandleIntent()

```
public class SampleIntentService extends IntentService {
  @Override
  protected void onHandleIntent(Intent intent) {
     long endTime = System.currentTimeMillis() + 5*1000;
     while (System.currentTimeMillis() < endTime) {
        synchronized (this) {
           try {
              wait(endTime - System.currentTimeMillis());
              Toast.makeText(this, “”+ endTime+ “ is over”+, Toast.LENGTH_SHORT).show();
           } catch (Exception e) {
           }
        }
     }
  }
```

Started Services – Extending Service class

To make Service work, all we need to do is implement

onStartCommand()

```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
  long endTime = System.currentTimeMillis() + 5*1000;
   while (System.currentTimeMillis() < endTime) {
      synchronized (this) {
         try {
            wait(endTime - System.currentTimeMillis());
            Toast.makeText(this, ""+ endTime+ " is over"+, Toast.LENGTH_SHORT).show();
         } catch (Exception e) {
         }
      }
   // If we get killed, after returning from here, restart
   return START_STICKY;
}
```

Bound Services

There are 3 ways to implement Bound Services

- **Extending the Binder class**: allow local application to access public methods of Service
- **Using a Messenger:** perform interprocess communication (IPC) to other applications by Messenger object and allow the service to handle one call at a time.
- **AIDL :** create an .aidl file that defines IPC to allow clients from different applications to access service and handle multithreading to service.

Bound Services - References

Here is some references of Bound Services

- **Extending the Binder class**:
- **Using a Messenger:** http://developer.android.com/guide/components/bound-services.html
- **AIDL :** http://developer.android.com/guide/components/aidl.html

Exit Course

THANK YOU