Day 11: Android Main Concepts: Services

Today we will explore what Services in Android world are.

Services in Android is just another Application Component that are used to perform long running tasks. Services are generally started by another components such as Activities in your Android Application. Services can be used in two ways –

1. Started Services
2. Bounded Services

# Started Services

A Started Service is triggered by an Application Component such as Activity by calling “StartService” method. The Service can also be stopped by the Application Component by calling the “StopServiceThis” method. This type of service can run in the background even if the Activity that started is destroyed. General rule of using Started Services is more like fire and forget where any result to the calling Activity need not be returned.

# Bounded Services

A Bounded Service is triggered by an Application Component such as Activity by calling “BindService” method. This type of service offers a programmatic interface for the users to interact with.

Let’s take a look at Simple Service Example (Please note, this sample is based on the Simple Service Sample written by Xamarin –

A Service in Xamarin.Android is defined as a class that inherits from “Service” Base class, and for declaring the service in the manifest, Xamarin users C# Attributes decoration (similar to Activities) . Let’s see that in action –

|  |
| --- |
| using Android.App;  using Android.Content;  using Android.OS;  using Android.Util;  namespace ServicesSample  {  public class DemoService : Service  {  public override void OnStart(Intent intent, int startId)  {  base.OnStart(intent, startId);  Log.Debug("DemoService", "OnStart DemoService");  DoBackgroundWork();  }  public override void OnDestroy()  {  base.OnDestroy();  Log.Debug("DemoService", "Destroying DemoService");  }  private void DoBackgroundWork()  {  Log.Debug("DemoService", "Doing Background work in the DemoService");  }  public override IBinder OnBind(Intent intent)  {  return null;  }  }  } |

Gist file link: <https://gist.github.com/vkoppaka/6cf2110f787cbcace53b>

The OnStart and OnDestroy methods get called when the Service is Started and Stopped respectively.

And the Activity to call the Service would have the following code –

|  |
| --- |
| using System;  using Android.App;  using Android.Content;  using Android.OS;  using Android.Widget;  namespace ServicesSample  {  [Activity(Label = "ServicesSample", MainLauncher = true, Icon = "@drawable/icon")]  public class MainActivity : Activity  {  protected override void OnCreate(Bundle bundle)  {  base.OnCreate(bundle);  // Set our view from the "main" layout resource  SetContentView(Resource.Layout.Main);  // Get our button from the layout resource,  // and attach an event to it  Button startServiceButton = FindViewById<Button>(Resource.Id.StartServiceButton);  startServiceButton.Click += startServiceButton\_Click;  Button stopServiceButton = FindViewById<Button>(Resource.Id.StopServiceButton);  stopServiceButton.Click += stopServiceButton\_Click;  }  private void stopServiceButton\_Click(object sender, EventArgs e)  {  StopService(new Intent(this, typeof (DemoService)));  }  void startServiceButton\_Click(object sender, EventArgs e)  {  StartService(new Intent(this, typeof (DemoService)));  }  }  } |

Gist file link: <https://gist.github.com/vkoppaka/562457a72e6294cc8d57>

Notice the StartService and StopService methods which take an Intent as an argument and the intent type is DemoService that we created above.

This article talks about how to create a really simple service and this concept can be expanded hugely, we will reserve that for some other blog post series.

That’s it for today. From tomorrow, we will start exploring different Controls (Views) in Xamarin.Android.

See you tomorrow.

Venkata