

A professional portrait of a man with short brown hair, smiling, wearing a dark suit jacket, a white shirt, and a light-colored tie. He is positioned on the left side of the slide, partially obscured by a large blue diagonal shape.

Xamarin Evolve 2014

# Backgrounding in Android

Adrian Stevens  
[adrian.stevens@xamarin.com](mailto:adrian.stevens@xamarin.com)

 Xamarin  
University

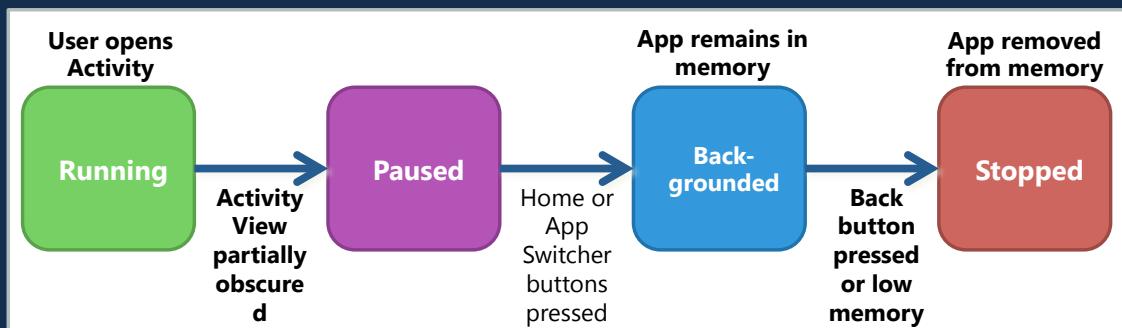
## Agenda

- Android Services
- Starting an Android Service
- Binding to an Android Service

Two brown plush monkeys wearing black t-shirts with the "Xamarin" logo. One monkey is in the foreground, slightly out of focus, and the other is in the background, also slightly out of focus. They are positioned on the right side of the slide, partially obscured by a yellow diagonal shape.

Xamarin University

## Android Activity Lifecycle



Xamarin University

## Android Background Tasks

- Activities lifetime is not well suited to long-running background tasks
- Android uses *Services* to process tasks in the background
- Services can be used in three ways:
  - Start Service
  - Bind to Service
  - Start & Bind to Service



Xamarin University

## The Service Component

```
[Service]  
public class MyService : Service  
{  
    public override StartCommandResult OnStartCommand(...)  
  
    public override IBinder OnBind(Intent intent)  
}
```

Xamarin University

## Types of Services

- **Started** Service (started manually)



Xamarin University

## Types of Services

- **Started** Service (started manually)
- **Bound** Service (on demand)



Xamarin University

## Types of Services

- **Started** Service (started manually)
- **Bound** Service (on demand)
- **Hybrid** Service (started + bound)



Xamarin University

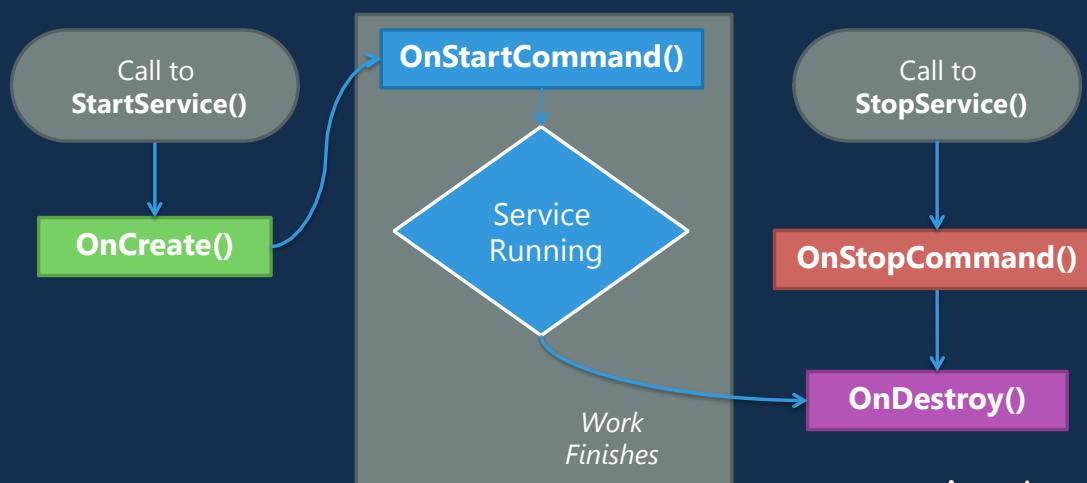
## Types of Services

- **Started** Service (started manually)
- **Bound** Service (on demand)
- **Hybrid** Service (started + bound)



Xamarin University

## Started Service Lifecycle



Xamarin University

## Starting and Stopping a Service

```
context.StartService(new Intent(this, typeof(MyService)));  
context.StopService(new Intent(this, typeof(MyService)));
```

Xamarin University

Demo

Xamarin University

## OnStartCommand Method

- Called when a service is started
- Returns a `StartCommandResult`

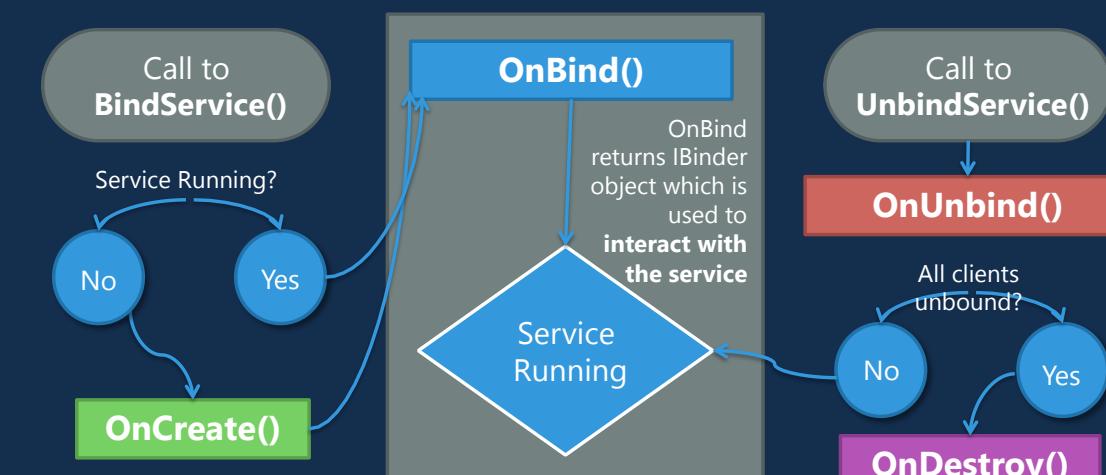
returns

- Sticky
- NotSticky
- RedeliverIntent

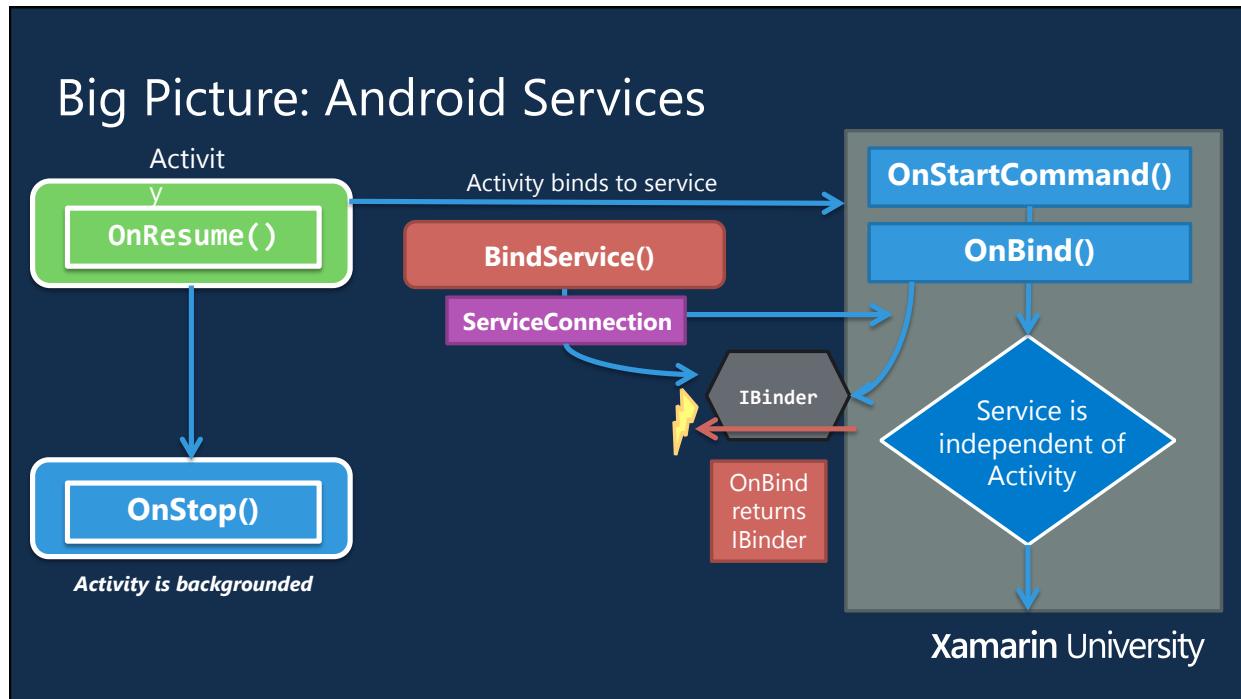
```
public override StartCommandResult OnStartCommand(Intent intent,  
StartCommandFlags flags, int startId)  
{  
    ...  
    return StartCommandResult.Sticky;  
}
```

Xamarin University

## Bound Service Lifecycle



Xamarin University



## Binding to a Service

- Create an object that implements **IServiceConnection** and wire up the **ServiceConnected** event on it.
- Create an Intent that contains the Context of the client (such as an Activity), and the Type of the Service to be connected to.
- Call **BindService** on a Context and pass the Intent. Android will then locate the Service, create it, and call **OnBind** on it.

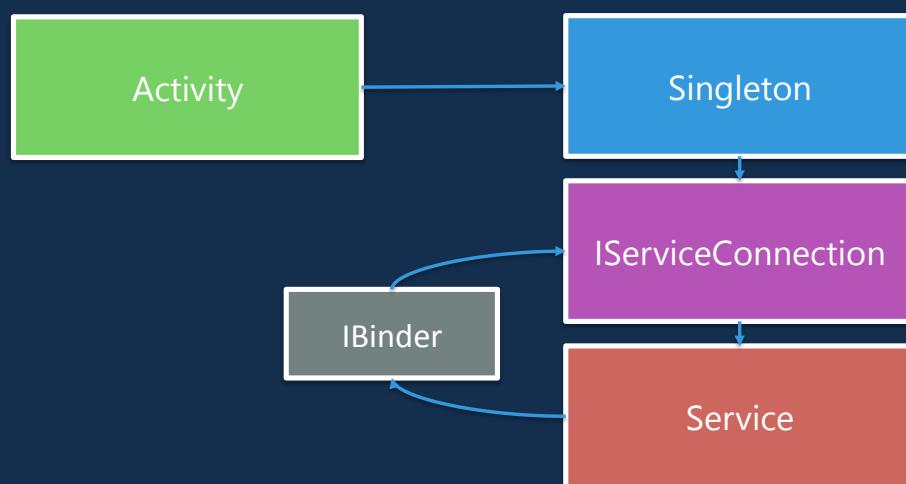
Xamarin University

## Steps to Bind to a Hybrid Service

- Create a Service Object
- Create a Binder class
- Create an `IServiceConnection` to locate the service
- Return a Binder implementation from the Service (`OnBind`)
- Start the Service
- Bind to the Service

Xamarin University

## Project Overview



Xamarin University

# Demo

Xamarin University

## Connecting to a Bound Service

Required to implement two Interfaces:

- **IBinder** allows us to reference a running service
- **ServiceConnection** is used by Android to locate the running service



Xamarin University

## Summary

- Activity Lifecycle
- Starting an Android Service
- Binding to a Service

Xamarin University



Xamarin Evolve 2014

## Backgrounding in Android

Adrian Stevens  
[adrian.stevens@xamarin.com](mailto:adrian.stevens@xamarin.com)

 **Xamarin**  
University