Subscribing to Lifecycle Events

Jim Wilson

MOBILE SOLUTIONS DEVELOPER & ARCHITECT

@hedgehogjim   blog.jwhh.com

What to Expect from This Module

Cooperating with activity lifecycle

Our app's "Get Together" feature
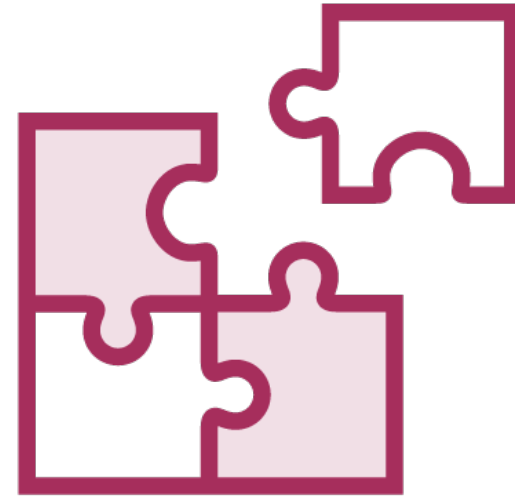
Lifecycle-related types

Lifecycle events

Creating a lifecycle observer

Cooperating with Activity Lifecycle
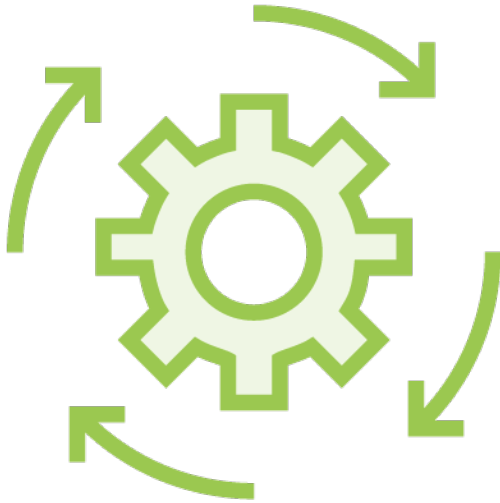

Activities have a lifecycle

Our code needs to cooperate with that lifecycle

App's leverage other components

Some components need to cooperate with that lifecycle

Components and Activity Lifecycle
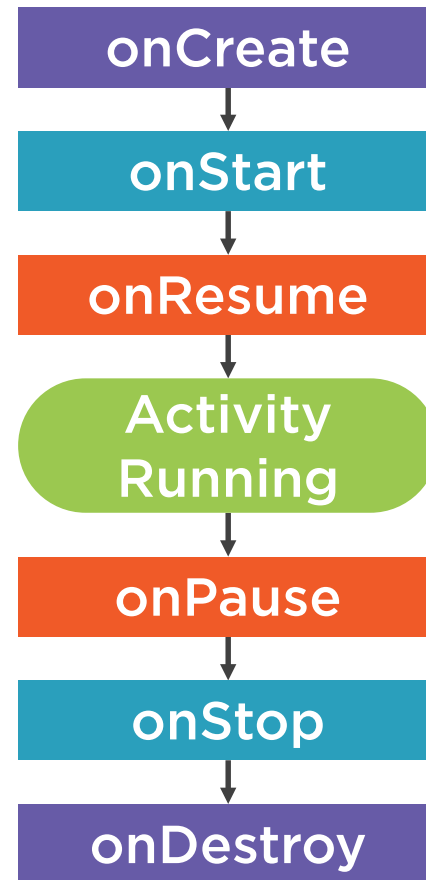
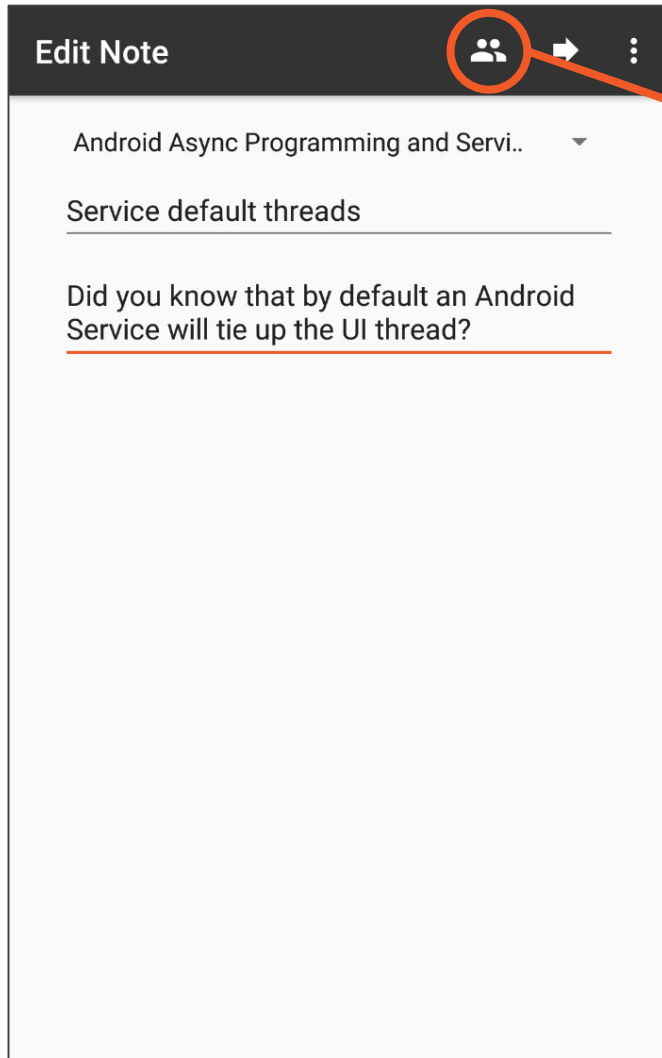**Start or connect to a local service**

Connect to a remote backend

May leak resources if not cleaned up

Activity Lifecycle Methods

```
┌─────────────────┐
│    onCreate     │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│    onStart      │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│   onResume      │
└─────────────────┘
         │
         ▼
╭─────────────────╮
│    Activity     │
│    Running      │
╰─────────────────╯
         │
         ▼
┌─────────────────┐
│    onPause      │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│    onStop       │
└─────────────────┘
         │
         ▼
┌─────────────────┐
│   onDestroy     │
└─────────────────┘
```

Our App's Get-together Feature

Edit Note

Android Async Programming and Servi.. ▾

Service default threads

Did you know that by default an Android
Service will tie up the UI thread?

Invite a pre-defined list of people to meet you at your current location to discuss the info in your note

Location Component

Messaging Component

Get-together Components

PseudoLocationManager

- Simulates location updates

- Start when activity onStart called

- Stop when activity onStop called

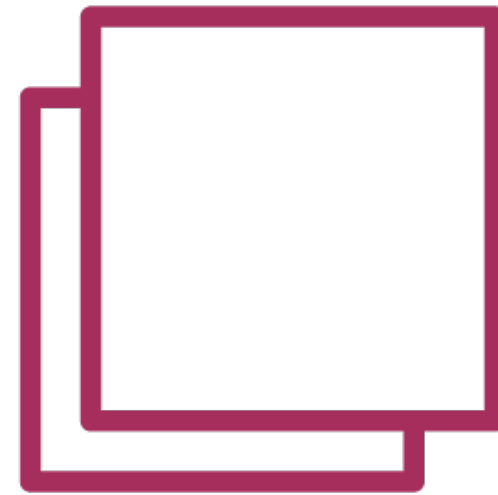- Provide callback for location updates

PseudoMessagingManager

- Simulates messaging behavior

- Connect when activity onStart called

- Disconnect when activity onStop called

- Use to send location and note info

Improving Lifecycle Handling
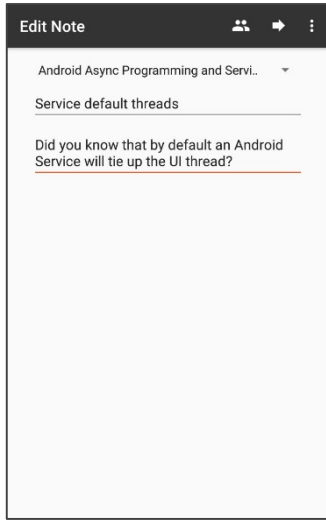
Cooperate with activity lifecycle

But want to avoid bloating the activity class

Separate handling from the activity

Need a way to be notified when each lifecycle event occurs
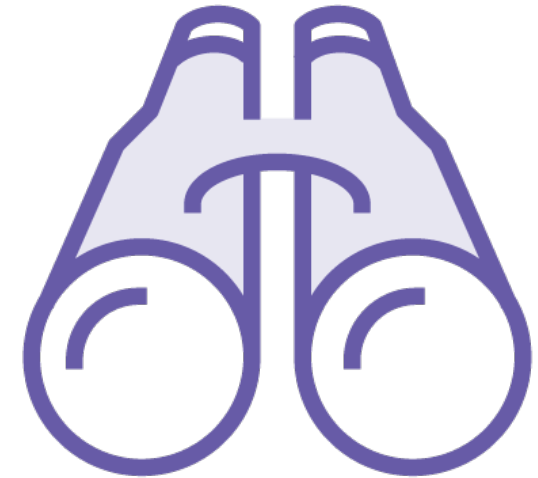
Improving Lifecycle Handling

LifecycleOwner

Implemented by types that have a lifecycle
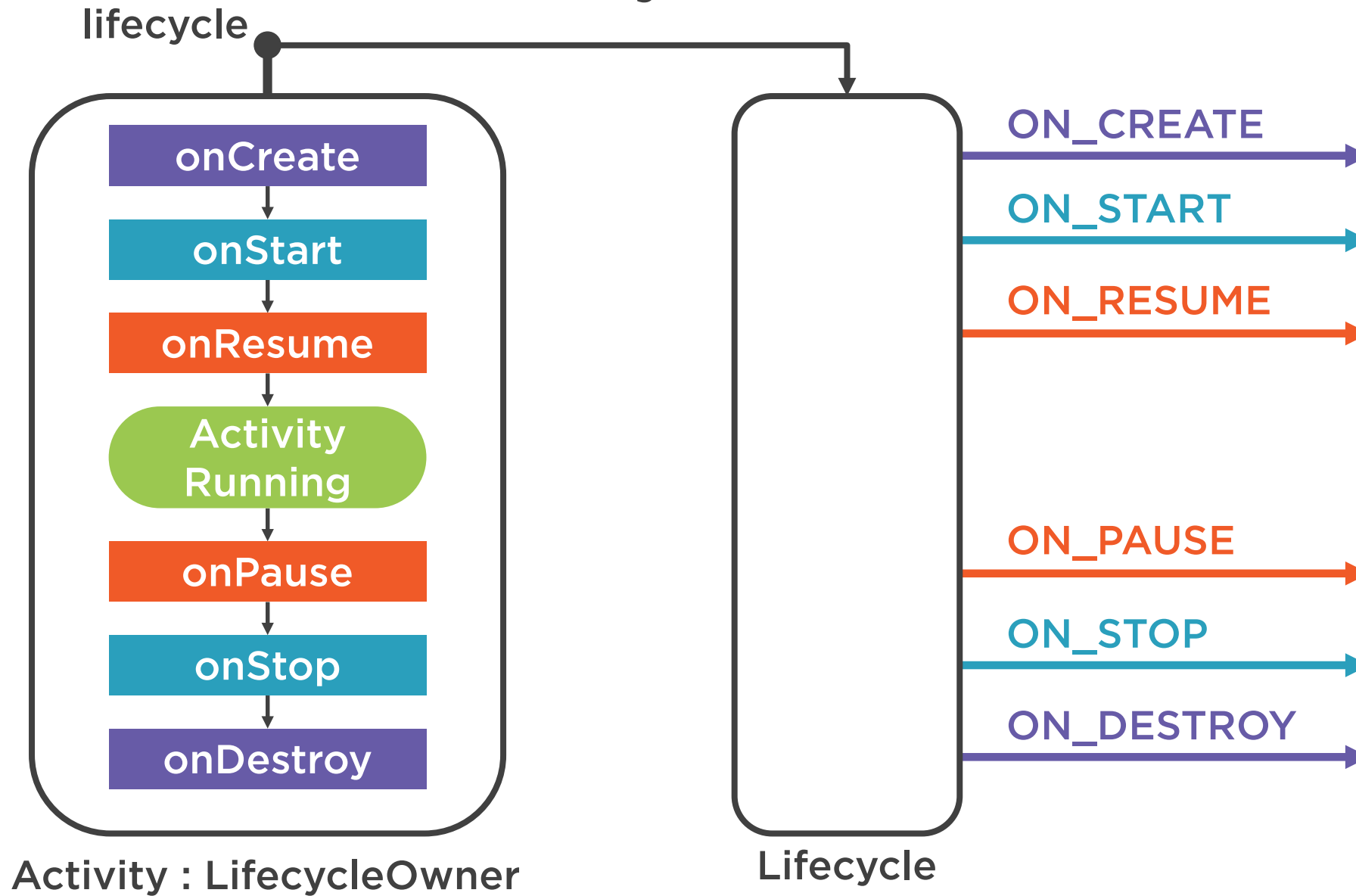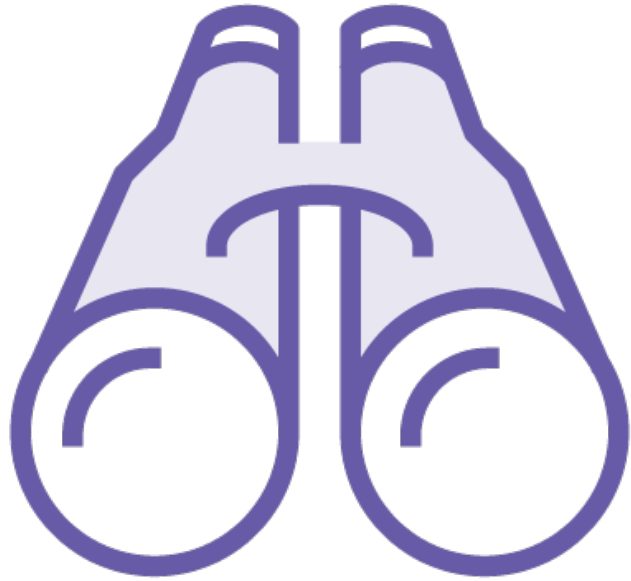
Lifecycle

Represents an object's lifecycle

LifecycleObserver

Implement to indicate support for receiving lifecycle events

Lifecycle Events

lifecycle

onCreate

onStart

onResume

Activity Running

onPause

onStop

onDestroy

Activity : LifecycleOwner

Lifecycle

ON_CREATE

ON_START

ON_RESUME

ON_PAUSE

ON_STOP

ON_DESTROY

Observing Lifecycle Events

Mark class as an observer

- Implement LifecycleObserver

Mark event handling functions within class

- Use OnLifecycleEvent annotation

- Specify event of interest with Lifecycle.Event.ON_xxxx constants

Register as an observer

- Use Lifecycle.addObserver

Summary

Components and activity lifecycle

- Sometimes need to cooperate
- Activity methods can get bloated

Lifecycle and related types

- Allow separating lifecycle work from activities

Summary

LifecycleOwner

- Implemented by types with a lifecycle

Lifecycle

- Represents an object's lifecycle
- Fires lifecycle-related events

Lifecycle events

- Correspond to activity lifecycle-related callback functions

Summary

LifecycleObserver

- Implement to indicate support for receiving lifecycle events

Lifecycle event handling functions

- Mark with @OnLifecycleEvent

- Specify event of interest with Lifecycle.Event.ON_xxxx constants