

# Services und Job Scheduler



@[https://appdeveloper magazine.com/images/news\\_images/Android-Test-Butler-App-Developer-Magazine\\_b80f71bi.jpg](https://appdeveloper magazine.com/images/news_images/Android-Test-Butler-App-Developer-Magazine_b80f71bi.jpg)

Ein Vortrag von Christopher Kluck

# Inhalt

- Was sind Services?
- Wichtige Methoden
- Gestartete Services
- Gebundene Services
- Lifecycles
- Job Scheduler
- Code Beispiele

# Services

# Was sind Services?

Anwendungsbausteine, die eine länger andauernde Tätigkeit im Hintergrund ausführen

# Was sind Services?

- Zwei Arten von Services:
  - Gestartete Services
  - Gebundene Services
- Müssen im Manifest deklariert werden
- Eigene Services ableiten von `android.app.Service` oder einer Unterklasse

# Deklarieren im Manifest

```
<manifest ... >
```

```
...
```

```
<application ... >
```

```
    <service android:name=".ExampleService" />
```

```
...
```

```
</application>
```

```
</manifest>
```

# Wichtige Methoden

- onStartCommand()

Aufgerufen durch startService()

Gibt Integer zurück

Nur wichtig für gestartete Services

- onCreate()

One-Time Setup bei Start des Services

# Wichtige Methoden

- **onBind()**

Client bindet sich an Service durch `bindService()`

Gibt `IBinder` oder null zurück

Muss immer implementiert werden

- **onDestroy()**

Aufgerufen, wenn Service nicht länger gebraucht wird und zerstört werden soll

Freigeben von Ressourcen



# onStartCommand() Konstanten

- **START\_NOT\_STICKY**

Service nur bei ausstehenden Intents neugestartet

- **START\_STICKY**

Service auf jeden Fall neu starten. Falls keine Intents vorhanden, wird null übergeben

- **START\_REDELIVER\_INTENT**

Bei Neustart wird letztes Intent übergeben

# Gestartete Services

- Unabhängig von Komponente, die Service gestartet hat
- Können auf unbestimmte Zeit im Hintergrund laufen
- Stoppen sich selbst mit `stopSelf()`, wenn Job erledigt ist
- Durch Komponenten gestoppt mit `stopService()`

# Erben von IntentService

- Intents in separatem Thread zur App ausgeführt
- Intents in Queue abgearbeitet
- Service stoppt nachdem alle Intents bearbeitet
- Standardimplementation für `onBind()` und `onStartCommand()`, die Intents an `onHandleIntent()` weiterleitet

# Erben von Service

- Ermöglicht Multi-Threading
- Anfragen unabhängig voneinander bearbeitet
- Service stoppt, wenn alle Threads beendet

# Gebundene Services

- Laufen solange mindestens ein Client verbunden
- Kommunikation zwischen Client und Service über IBinder Interface
- Client verbindet mit `bindService()` und beendet Verbindung mit `unbindService()`
- Ermöglichen Interprozesskommunikation

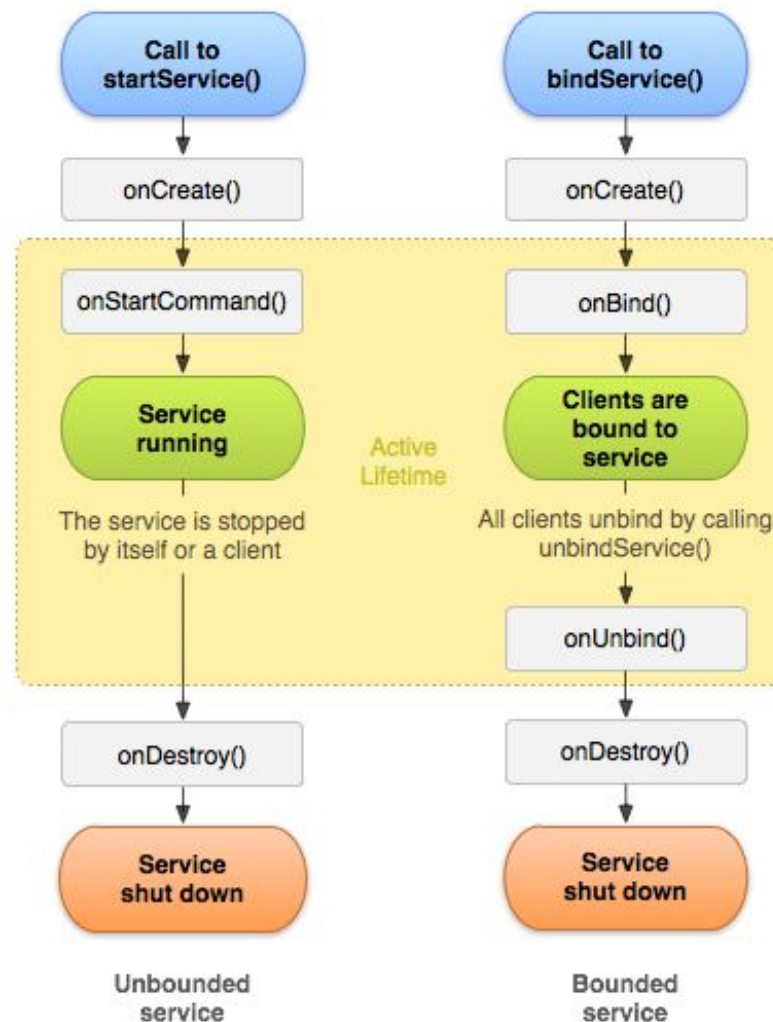
# Erweitern der Binder Klasse

- **Voraussetzung:**  
Service und Client in selber Applikation und selbem Prozess
- Interface IBinder nicht selbst implementiert, sondern von android.os.Binder abgeleitet

# Nutzen eines Messengers

- Funktion über Prozessgrenzen hinweg
- How to:
  1. Service implementiert Handler
  2. Service nutzt Handler um Messenger zu erstellen
  3. Messenger erstellt IBinder
  4. Client nutzt IBinder um Messenger zu instanziiieren
  5. Service empfängt Message in seinem Handler

# Lifecycles





# Job Scheduler

# Job Scheduler

- Job-Scheduler-API mit Android 5 eingeführt
- Ausführen von Hintergrundaktionen bei Eintreten bestimmter Bedingungen
- Automatischer Wiederanlauf, wenn Aufgabe nicht erfolgreich
- Regelmäßige Wiederholung von Jobs

# Quellen

Android Developers:

<https://developer.android.com/guide/components/services.html> (Stand 27.10.2017)

Thomas Künneth:

Android 7 „Das Praxisbuch für Entwickler“  
(S. 237-263)