

# BLM401 Mobil Cihazlar için ANDROİD İşletim Sistemi

## SERVİS BİLEŞENLERİ



# GİRİŞ (1/5)

- **Servisler** arka planda görev yapan görsel olmayan Android bileşenleridir.
- Ekranda herhangi bir aktivite görünürken arka planda bir veya daha fazla sayıda **servisler** çalışıyor olabilir.
- Müzik çalma, web üzerinden veri aktarımı, kalıcı hafızaya veri yazma okuma gibi zaman alan işlemler kullanıcının ekran üzerindeki çalışmasını etkilemeden arka planda **servisler** aracılığı ile yapılabilirler.

# GİRİŞ (2/5)

- **Servisler** de diğer bileşenler gibi ana aktivite ile aynı tread içerisinde çalışırlar.
- Fazla zaman gerektiren **servisler** ekranın kilitlemesine neden olmaktadır.
- Kullanıcı bir tuşa bastığında çalışmaya başlayan **servis** işini bitirene kadar ekran kullanılamaz hale gelmektedir.
- Bu tür istenmeyen özellikleri önlemek için fazla zaman gerektiren **servislerin** ayrı tread içerisinde çalışması gerekmektedir.

# GİRİŞ (3/5)

- **startService()** – bir servisi başlatma fonksiyonu;
- **onStartService()** – servis fonksiyonu;
- **stopService()** – bir servisi durdurma fonksiyonu;
- **bindService()** – bir servise bağlanma fonksiyonu;

# GİRİŞ (4/5)

- **startService()** ile başlatılmak istenen servis daha önce çalıştırılmamışsa bir nesnesi yaratılır ve çalıştırmaya başlanır, daha önceden çalıştırılmışsa da çalışmaya devam eder.
- **startService()** fonksiyonu her çağırıldığında sistem, servisin **onStartService()** fonksiyonunu çağırır ve servis **stopService()** fonksiyonu çağırılincaya kadar çalışmaya devam eder.
- **startService()** birden fazla çağrılmasıyla servisin birden fazla nesnesi yaratılmaz.

# GİRİŞ (5/5)

- **startService()** ile başlatılan servisi durdurmak için **stopService()** fonksiyonunun bir defa çağrılması yeterlidir.
- **bindService()** bir servise bağlanmak içindir.
- Servis yaratılmamışsa bir nesnesi yine yaratılır. Ancak bu kez **onStartService()** çağrılmaz.
- Servise bağlanan aktivite ile servis arasında bir bağlantı kurulur ve aktivite servisle haberleşir.
- Servise bağlı herhangi bir nesne kalmadığında ise servis otomatik olarak durdurulur.

# SERVİSİN YARATILMASI (1/5)

```
public class DenemeServisi extends Service {
    @Override
    public void onCreate() {
    }
    @Override
    public IBinder onBind(Intent intent) {
    }
    @Override
    public int onStartCommand (Intent intent, int flags, int startId) {
    }
    @Override
    public void onDestroy() {
    }
}
```

# SERVİSİN YARATILMASI (2/5)

- **onCreate() metodu:** servis ilk kez yaratıldığında sistem bu metodu çağırır.
- Servisin daha sonraki durdurulup başlatılmalarında bu fonksiyon çağrılmamaktadır.
- İlk kez yaratılma esnasında yapılması gerekenler burada kodlanmalıdır.



# SERVİSİN YARATILMASI (3/5)

- **onBind() metodu**: servise bir sınıfın bağlanması durumunda çağrılacak olan metottur.
- Bir sınıf **bindService()** fonksiyonunu çağırdığında sistem **onBind()** metodunu çağırır.
- **onBind()** metodu sonucunda dönen IBinder nesnesi aracılığıyla çağrılan sınıf ile servis haberleşirler.

# SERVİSİN YARATILMASI (4/5)

- `onStartCommand ()` metodu: bir sınıfın `startService()` fonksiyonunu çağırdığı her durumda çağrılan servis metotlarından biridir.
- Bu fonksiyon içerisinde servisin yerine getirmesini istediğimiz işlevleri kodlayabiliriz.
- Servisin harici olarak başlatılıp durdurulması isteniyorsa bu fonksiyonun `START_STICKY`, yalnızca kendisiyle haberleşirken çalışması bekleniyorsa bu durumda `START_NOT_STICKY` sabitini döndürmesi gerekmektedir.

# SERVİSİN YARATILMASI (5/5)

- **onDestroy() metodu:** servisin stopService() fonksiyonu ile durdurulması veya herhangi bir bağlantısının kalmaması durumunda sistem servisi ortadan kaldırmadan hemen önce çağrılan metottur.
- Bu metotla servis içerisinde oluşturulan veri yapılarının hafızadan silinmesi gerekir.

# AndroidManifest.xml Kaydı(1/1)

- Bir uygulamanın tüm bileşenleri **AndroidManifest.xml** dosyası içerisinde kaydedilmelidir.
- Servisler de bu dosya içerisinde ve **<application>** sekmesi altında tanımlanmalıdır.

# Servisin Çalıştırılması (1/3)

- Bir servisin yaşamaya başlaması için en az bir bileşenin o servisi başlatması lazım.
- Servis başlatma işi **startService()** veya **bindService()** fonksiyonları ile yapılır.
- Servis başlatma için intent mesajları kullanılır.
- Genel olarak bir aktivite başladığı anda servisler başlatılmaz.

# Servisin Çalıştırılması (2/3)

- Genel olarak servislerin başlaması kullanıcının bir tuşa basması veya bir menü maddesini seçmesi ile tetiklenir.
- Bu olaylar gerçekleştiğinde de `startService()` veya `bindService()` fonksiyonları, ilgili aktiviteler içindeki `onClick()`, `onOptionsItemSelected()`, vb. gibi kullanıcı ara yüzlerinden alınan kullanıcı tepkilerine uygun yerlerde çağrılırlar.

# Servisin Çalıştırılması (3/3)

- Kontrolü tanımladığımız aktivitenin elinde olmayan bir servise bağlanmak için **bindService()** fonksiyonu çağrılır.
- Bu tür çağırma genellikle sistem servislerine erişmek için kullanılır ve bu servisle zaman zaman mesaj alışverişi yapılır.
- Bu amaçla aktivite içinde tanımlanan **ServiceConnection** sınıfı kullanılır.

# Servis Bileşeni ile ilgili linkler

<http://developer.android.com/reference/android/app/Service.html>

<http://www.vogella.com/articles/AndroidServices/article.html>

<http://examples.javacodegeeks.com/android/core/service/android-service-example/>

[http://www.techotopia.com/index.php/Implementing\\_an\\_Android\\_Started\\_Service\\_%E2%80%93\\_A\\_Worked\\_Example](http://www.techotopia.com/index.php/Implementing_an_Android_Started_Service_%E2%80%93_A_Worked_Example)

[https://thenewcircle.com/s/post/60/servicesdemo\\_using\\_android\\_services](https://thenewcircle.com/s/post/60/servicesdemo_using_android_services)



(son)

BAŞARILAR ...