

description: Android üzerinde dahili veya harici dosya sistemine erişme ve yönetme

## 📁 Dosya İşlemleri

### ⏏ Internal Storage (Dahili)

- 📁 Kalıcı depolama dizini `getFilesDir()`
- ⌚ Geçici depolama dizini `getCacheDir()`
  - 📁 1 MB ve daha hafif dosyalar için önerilir
  - 🗑 Hafıza yetmemeye başladığında burası silinebilir

```
File file = new File(context.getFilesDir(), filename);
```

```
String filename = "myfile";
String string = "Hello world!";
FileOutputStream outputStream;

try {
    outputStream = openFileOutput(filename, Context.MODE_PRIVATE);
    outputStream.write(string.getBytes());
    outputStream.close();
} catch (Exception e) {
    e.printStackTrace();
}
```

```
public File getTempFile(Context context, String url) {
    File file;
    try {
        String fileName = Uri.parse(url).getLastPathSegment();
        file = File.createTempFile(fileName, null, context.getCacheDir());
    } catch (IOException e) {
        // Error while creating file
    }
    return file;
}
```

### ⬆ External Storage (Harici)

- 🧑 Öncelikle okuma izinleri alınır
- 😊 Harici depolama sisteme bağlı mı kontrol edilir
- 🤖 Erişim hakkımızın olup olmadığı kontrol edilir
- 📁 Ardından dosyalara erişim yapılır

## Okuma İzinlerini Alma

```
<manifest ...>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
</manifest>
```

## Erişimi ve Bağlantıyı Kontrol Etme

```
/* Checks if external storage is available for read and write */
public boolean isExternalStorageWritable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state)) {
        return true;
    }
    return false;
}

/* Checks if external storage is available to at least read */
public boolean isExternalStorageReadable() {
    String state = Environment.getExternalStorageState();
    if (Environment.MEDIA_MOUNTED.equals(state) ||
        Environment.MEDIA_MOUNTED_READ_ONLY.equals(state)) {
        return true;
    }
    return false;
}
```

## Dosyaların Yönetimi

```
// Harici dosyalara erişebilmek için (public external storage)
File path = Environment.getExternalStoragePublicDirectory(
    Environment.DIRECTORY_PICTURES);
File file = new File(path, "DemoPicture.jpg");
```

```
// Gizli harici dosyalara erişmek için (private external storage)
File file = new File(getExternalFilesDir(null), "DemoFile.jpg");
```

## Dosyaları Temizleme

- 🗑 Kullanılmayan her dosya temizlenmelidir

```
myFile.delete();  
myContext.deleteFile(fileName);
```

## Faydalı Kaynaklar

{% embed url="https://google-developer-training.github.io/android-developer-fundamentals-course-concepts-v2/unit-4-saving-user-data/lesson-9-preferences-and-settings/9-0-c-data-storage/9-0-c-data-storage.html#files" %}