



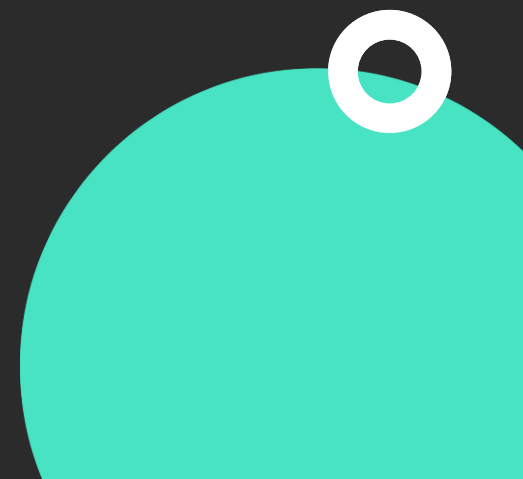
Mastering Android DownloadManager for Efficient File Downloads with Jetpack Compose



Musab Nasreldin

 @musabnasreldin

 in/musabagab



Overview

- Understand Android `DownloadManager` for efficient and reliable file downloads
- Learn how to integrate the Android `DownloadManager API`
- Explore techniques for handling network `connectivity changes` and interruptions
- Learn how to `track` download progress, handle `notifications`, and `cancel` downloads



Understand DownloadManager

- System Service
- Simplifies Download Process
- Centralized Solution



DownloadManager Integration

- Obtain `DownloadManager` Instance
- Create Download `Request`
- `Enqueue` Download Request



Code Snippet

```
// Get DownloadManager Instance
val downloadManager = this.getSystemService(DownloadManager::class.java)

// Create a download request
val request = DownloadManager.Request(Uri.parse(downloadUrl))

// Enqueue the download request
downloadManager.enqueue(request)
```



Handling Network Connectivity Changes

- Setting Network Types
 - `NETWORK_WIFI`
 - `NETWORK_MOBILE`
- Impact of Network Switch
 - DownloadManager adapts by `pausing` the download
 - Displaying a message as `notification`
 - The download `resumes` on specified network availability



Code Snippet

```
request.setAllowedNetworkTypes(DownloadManager.Request.NETWORK_WIFI)  
  
// or  
  
request.setAllowedNetworkTypes(DownloadManager.Request.NETWORK_MOBILE)
```



Notification Visibility in DownloadManager

- Setting Notification Visibility: Use `setNotificationVisibility()`
- Options for Visibility:
 - **VISIBILITY_VISIBLE**: Download is visible only while in progress.
 - **VISIBILITY_VISIBLE_NOTIFY_COMPLETED**: Download is visible during progress and after completion.
 - **VISIBILITY_HIDDEN**: Download doesn't show in UI or notifications.
 - **VISIBILITY_VISIBLE_NOTIFY_ONLY_COMPLETION**: Download appears in notifications only after completion.



Track Download Progress

- DownloadManager Query
- Query Filtering By ID
- Cursor To Extract Bytes Downloaded and Total
- Progress Calculation



Code Snippet

```
val query = DownloadManager.Query().setFilterById(downloadId)
val cursor = downloadManager.query(query)

cursor.use {
    if (it.moveToFirst()) {
        val bytesDownloaded = it.getLong(it.getColumnIndex
(COLUMN_BYTES_DOWNLOADED_SO_FAR)) * 100L

        val bytesTotal = it.getLong(it.getColumnIndex
(COLUMN_TOTAL_SIZE_BYTES))

        val progress = (bytesDownloaded / bytesTotal).div(100F)
    }
}
```





Clone Starter Project

Up Next



Summary

- In Android, DownloadManager is used to efficiently download files.
- DownloadManager can be easily integrated into your apps.
- Configure the allowed network type for downloading to a specific network.
- Query the DownloadManager to calculate download progress.
- To cancel a download by ID, use `downloadManager.remove(downloadId)`.

