



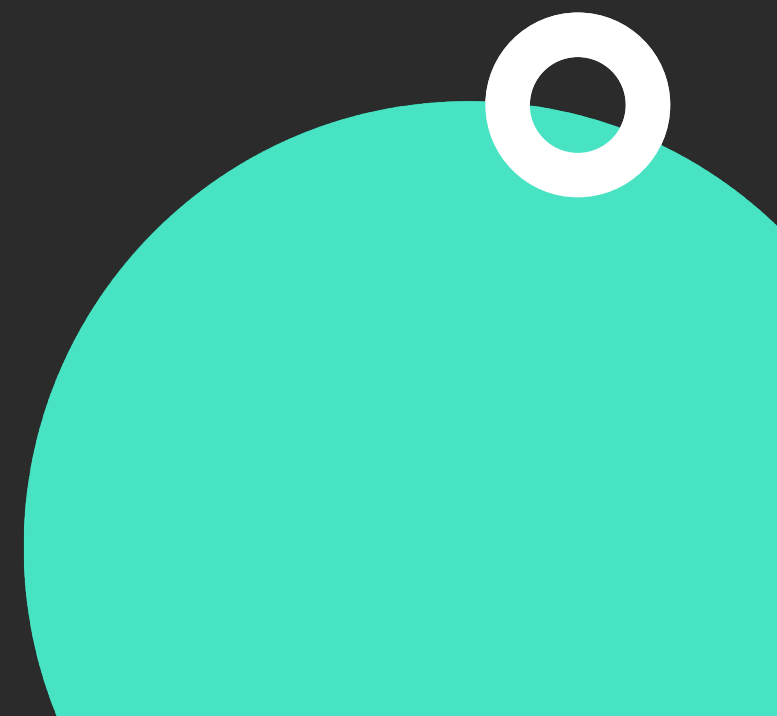
Executing Background Work with WorkManager in Android



Baljeet Singh

 @yetanotherdev_

 in/devbaljeet



Overview

- WorkManager
- Types of Persistent Work
- Running a One-Time Work

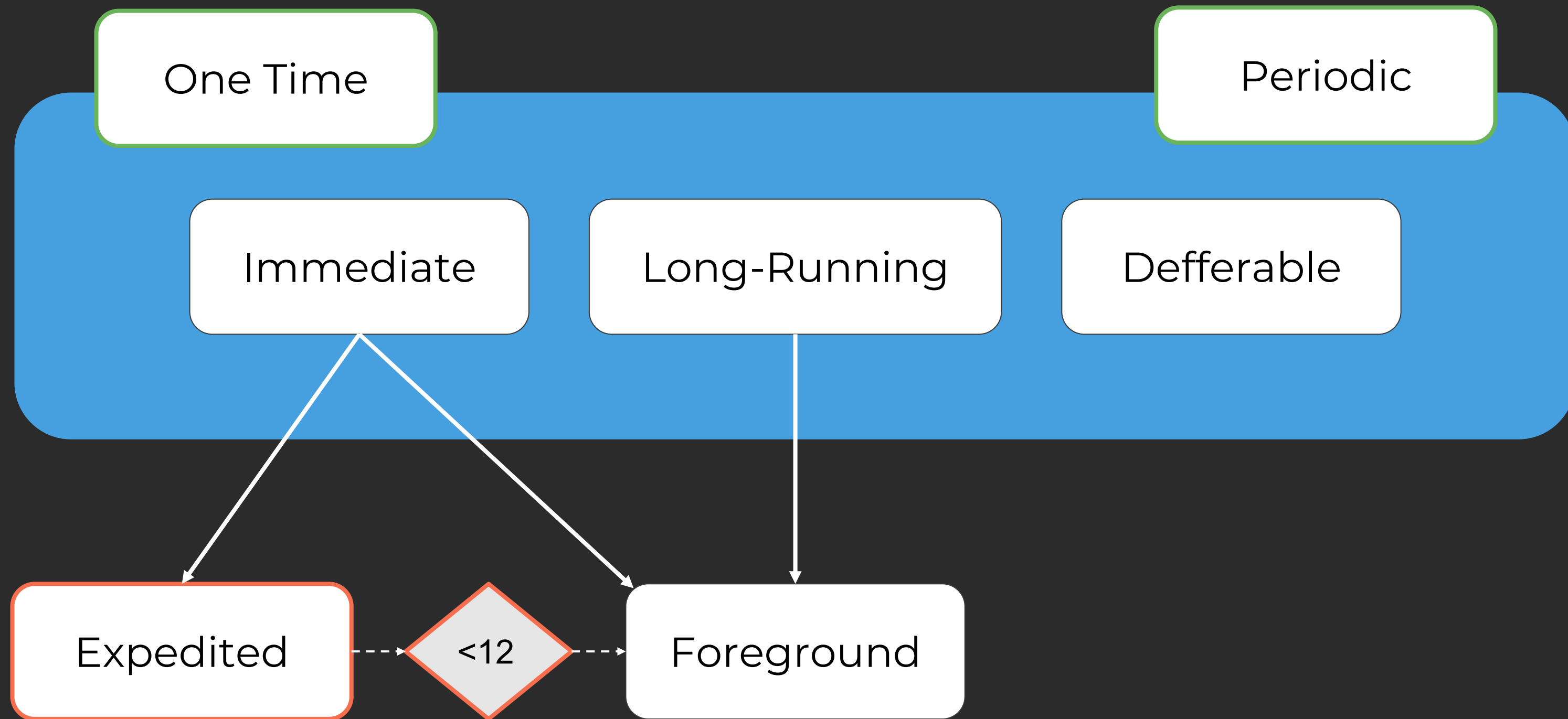


Introduction

- What is WorkManager
- When to use WorkManager
 - Start Background Work even when App process is not running.
 - Start Periodic Work in the background.
 - Start a work when certain conditions are satisfied.



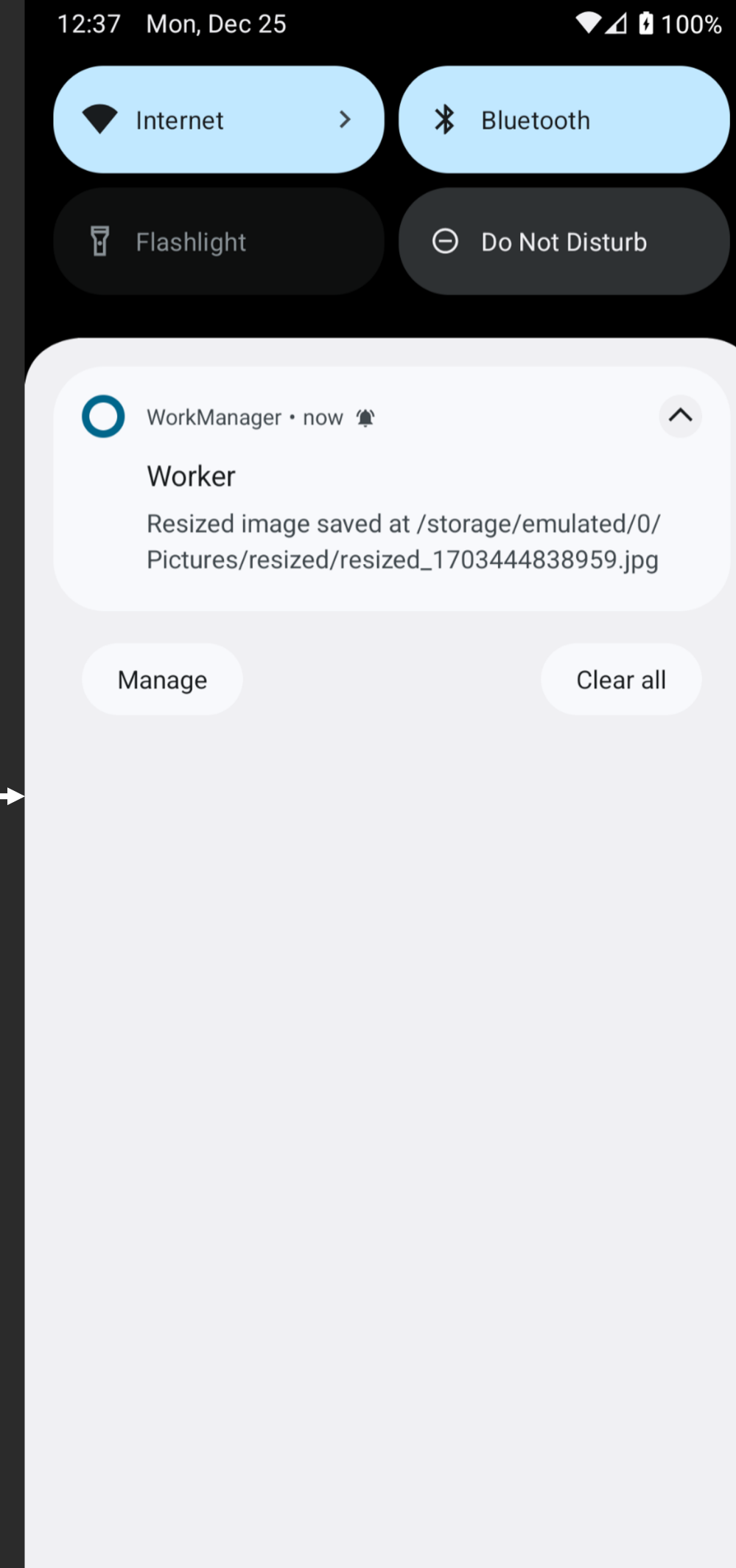
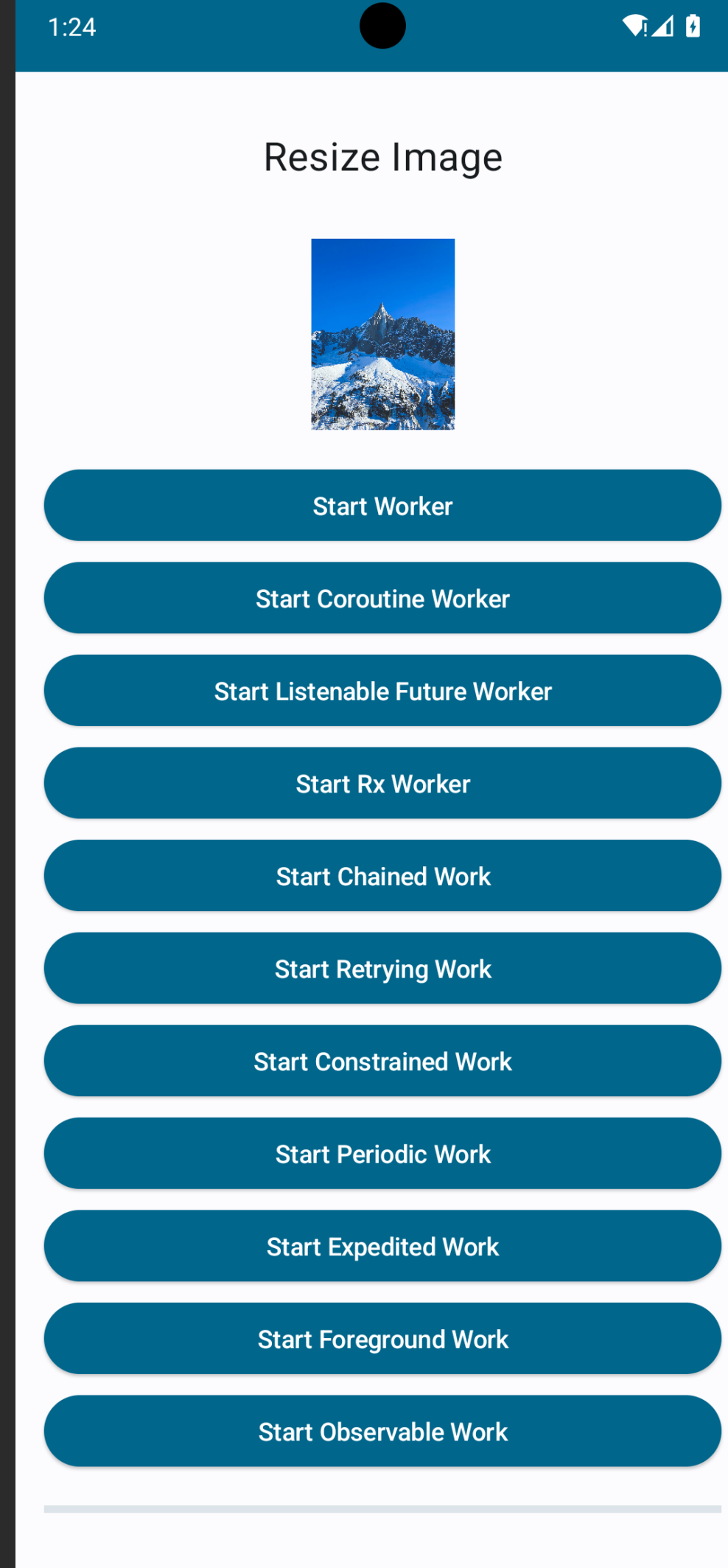
Types of Persistent Work



Course Demo App

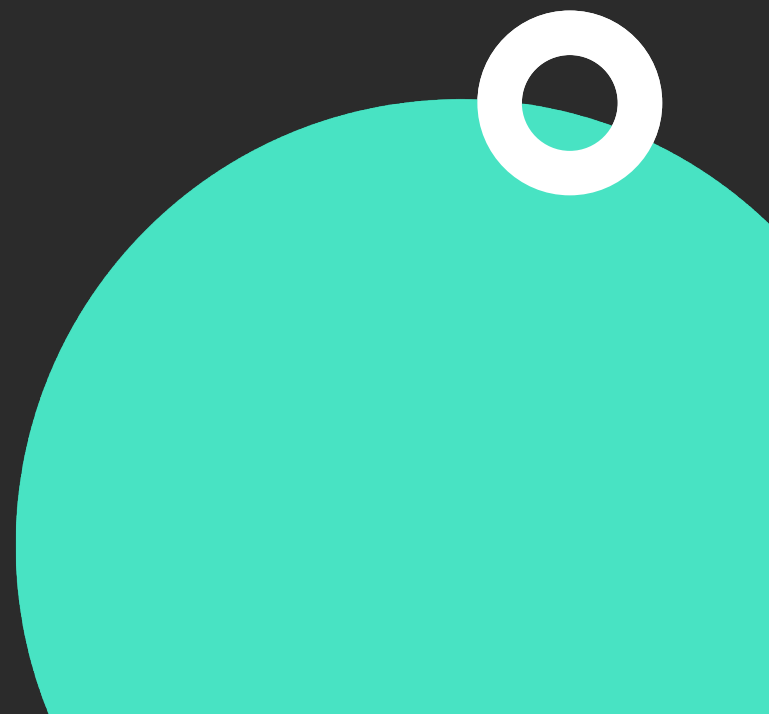
Image Resizer

- Libraries used:
 - WorkManager
 - Coroutines
 - Compose (UI)





Threading & Chaining in WorkManager



Overview

- Threading in WorkManager
- Chaining in WorkManager



Threading

- What are different types of threading mechanisms provided by WorkManager ?
 - Worker
 - CoroutineWorker
 - RxWorker
 - ListenableWorker



Chaining

- How to chain multiple works together ?



Chaining



7:38 Tue, Dec 26

Internet >

Bluetooth

Flashlight

Do Not Disturb

WorkManager

Worker • now

Image synced with Gallery

Worker • now

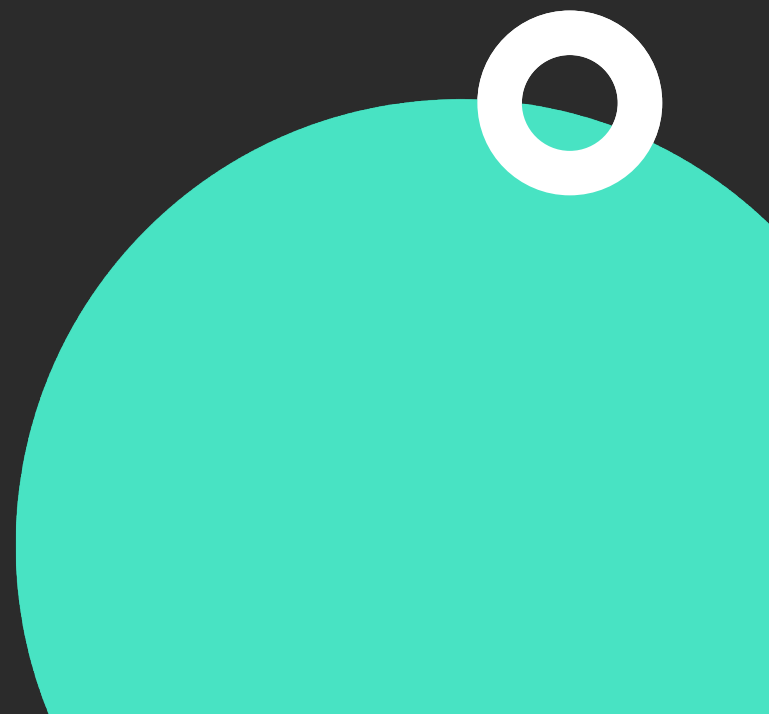
Image saved at: /storage/emulated/0/Pic...

Manage

Clear all



Constraints in Work, Periodic Workers and Retrying Failed Work



Overview

- Retrying a Work
- Constraints In WorkManager
- Periodic Work



Retrying a Work

- How to Retry a Failed Work ?



Constraints In WorkManager

- What are Constraints
- When to use Constraints
 - When there is a need to meet some condition in order for the Work to execute. For example: Active Network connection, Charging, Not Low-Storage, Not Low-Battery etc



Periodic Work

- What is Periodic Work
- When to use Periodic Work
 - When we need to start a work and keep it re-running periodically using a specified interval between two Works.



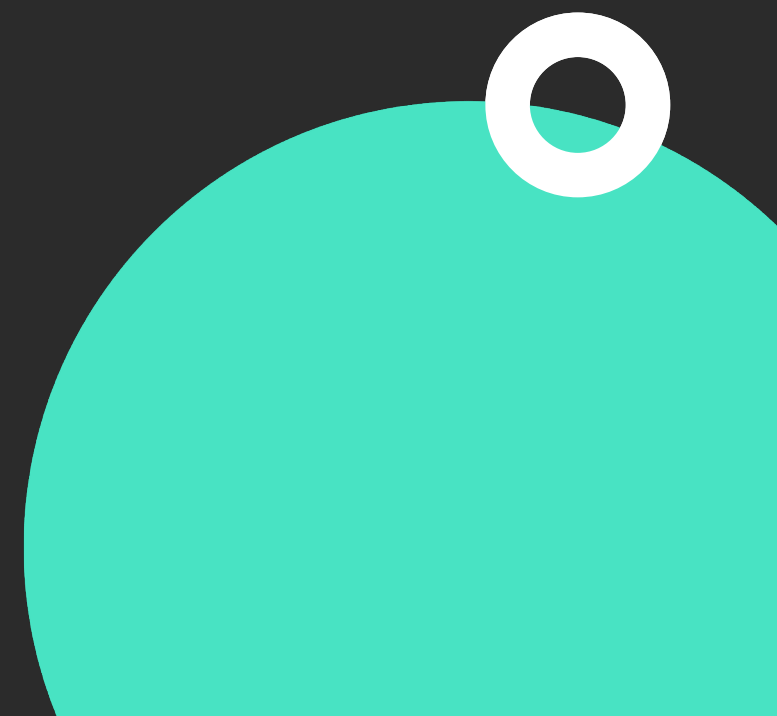
Limitations of Periodic Work

- What are the limitations of Periodic Work
 - Minimum interval between Workers is 15 minutes.
 - Output for Periodic work is not available due to its states.
 - Enqueued -> Running -> Enqueued
 - Periodic Work doesn't guarantee to run on specific time.





Work Progress & Status, Expedited & Foreground Work



Overview

- Observing Task Progress and Status
- Expedited and Foreground Work



Observing a Work

- `getWorkInfoById(workId)`
 - Gets a `ListenableFuture` of the `WorkInfo` for a given work id.
- `getWorkInfosByTag(tagName)`
 - Gets a `ListenableFuture` of the `WorkInfo` for all work for a given tag.
- `getWorkInfosForUniqueWork(uniqueWorkName)`
 - Gets a `ListenableFuture` of the `WorkInfo` for all work in a work chain with a given unique name.
- `getWorkInfos(workQuery)`
 - Gets the `ListenableFuture` of the List of `WorkInfo` for all work referenced by the `WorkQuery` specification.



Observing a Work

- LiveData
 - `getWorkInfoByIdLiveData(workId)`
 - `getWorkInfosByTagLiveData(tagName)`
 - `getWorkInfosForUniqueWorkLiveData(uniqueWorkName)`
 - `getWorkInfosLiveData(workQuery)`
- Flow
 - `getWorkInfoByIdFlow(workId)`
 - `getWorkInfosByTagFlow(tagName)`
 - `getWorkInfosForUniqueWorkFlow(uniqueWorkName)`
 - `getWorkInfosFlow(workQuery)`



Code Challenge

```
workManager.getWorkInfoByIdFlow(workId)
```





Coding Challenge Solution☀



Expedited Work

- What is Expedited Work and When to use it ?
 - Starting a time-critical work that needs to start immediately.
- Limitations
 - System allocated execution time quota for Expedited Jobs.



Foreground Work

- What is Foreground Work and When to use Foreground Work ?
 - When we need to notify user about the running task.



Summary

- Introduction To WorkManager
- When to use WorkManager?
- Types of Persistent Work: Immediate, Long Running and Deferrable
- Running a One-Time Work
- Chaining In WorkManager
- Threading In WorkManager
- Retry Workers
- Constraints In WorkManager
- Periodic Work
- Expedited and Foreground Work
- Observing Task Progress and Status





Thank You!



Baljeet Singh

 @yetanotherdev_

 in/devbaljeet

