droidcon academy

# Executing Background Work with WorkManager in Android


## Baljeet Singh

@yetanotherdev_

in/devbaljeet

# Overview

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

droidcon academy

# Introduction

- What is WorkManager
  - WorkManager is a library that makes is easy to schedule Deferrable, Asynchronous and Guaranteed Work that is expected to run even if the app exits or device Restarts.
  - WorkManager is Part of Android Jetpack.
  - WorkManager is Compatible with API 14+
  - WorkManager handles work based on Battery level, CPU, Storage, Network etc.

- 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.

# WorkManager

Popular apps using WorkManager

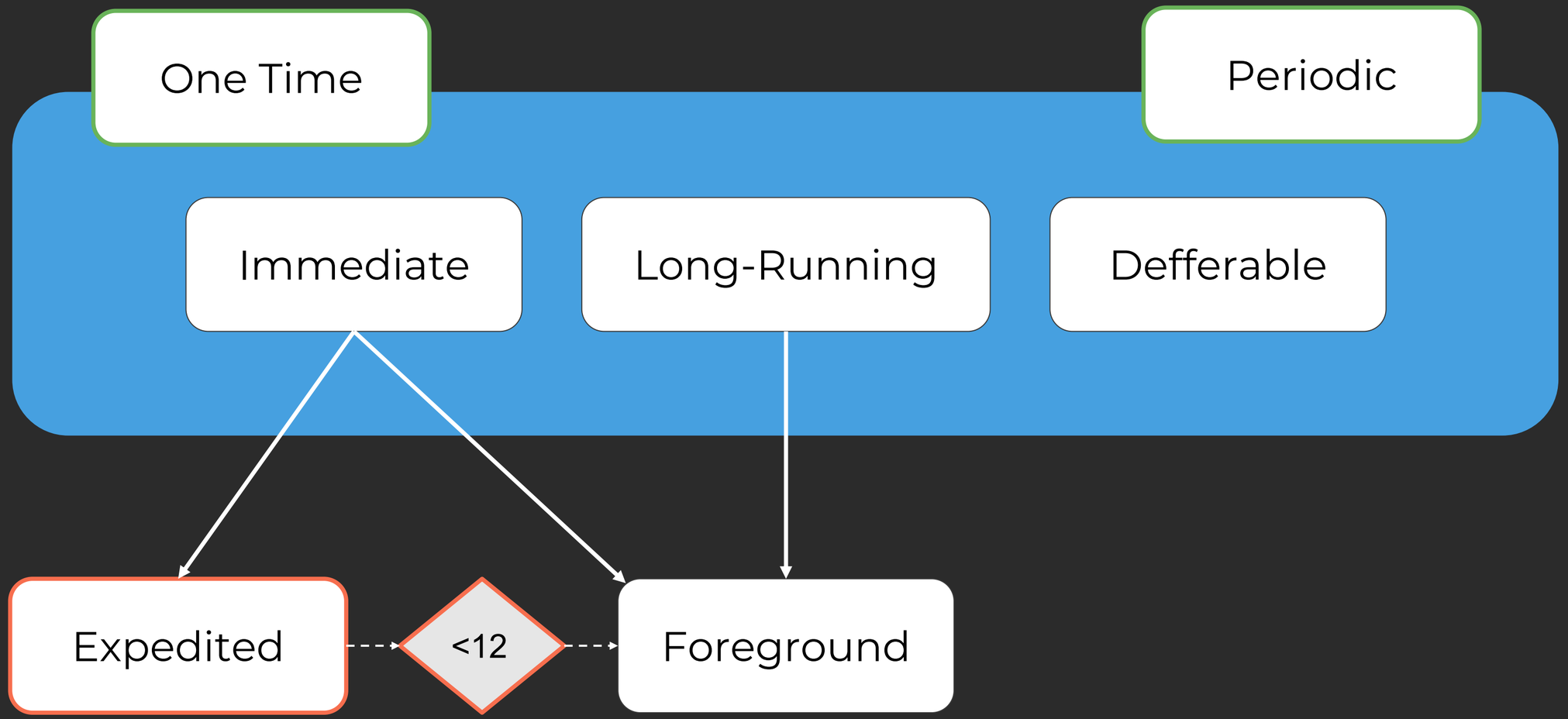
- Google Maps, WhatsApp, Slack etc.

Usecase

- Updating Location in the Background
- Uploading Logs when app is Not Running
- Saving local messages to the Cloud

droidcon academy

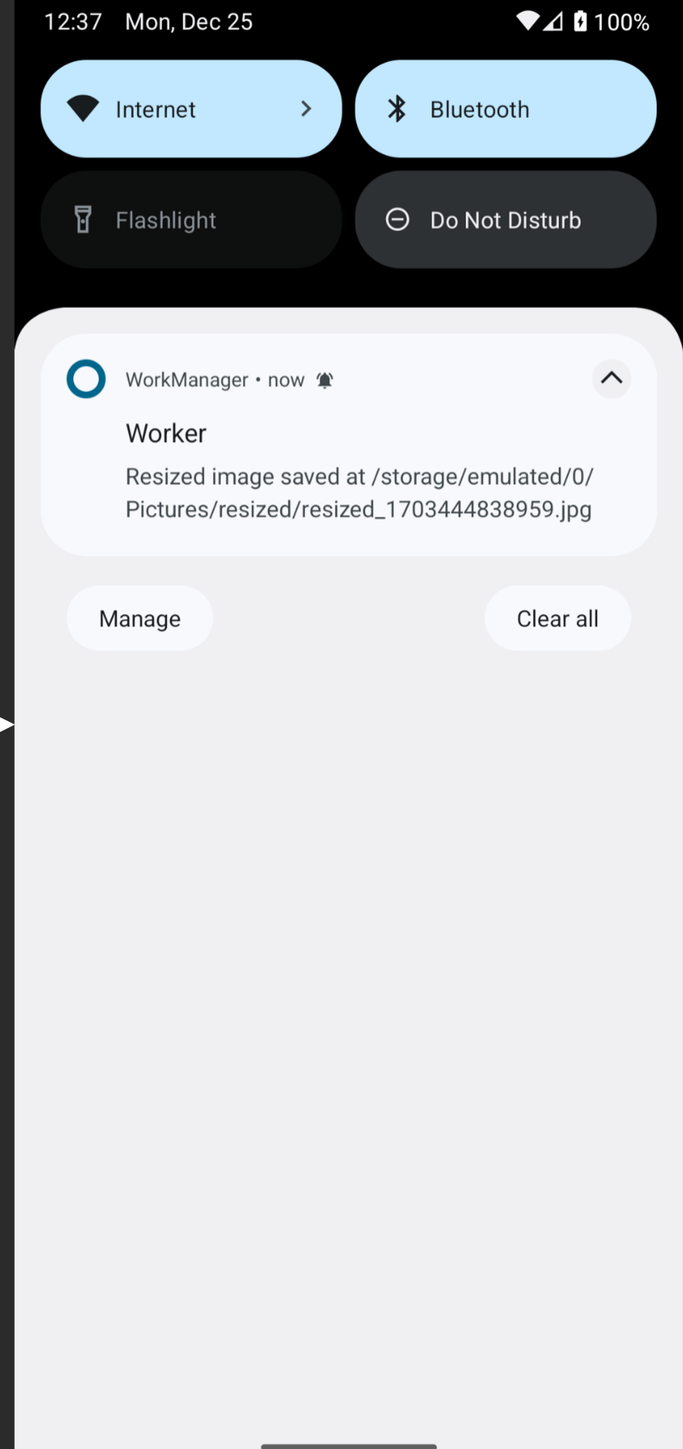
# Types of Persistent Work




# Course Demo App

Image Resizer

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

# Up Next

## Advanced Usage: Threading and Chaining in Work Manager

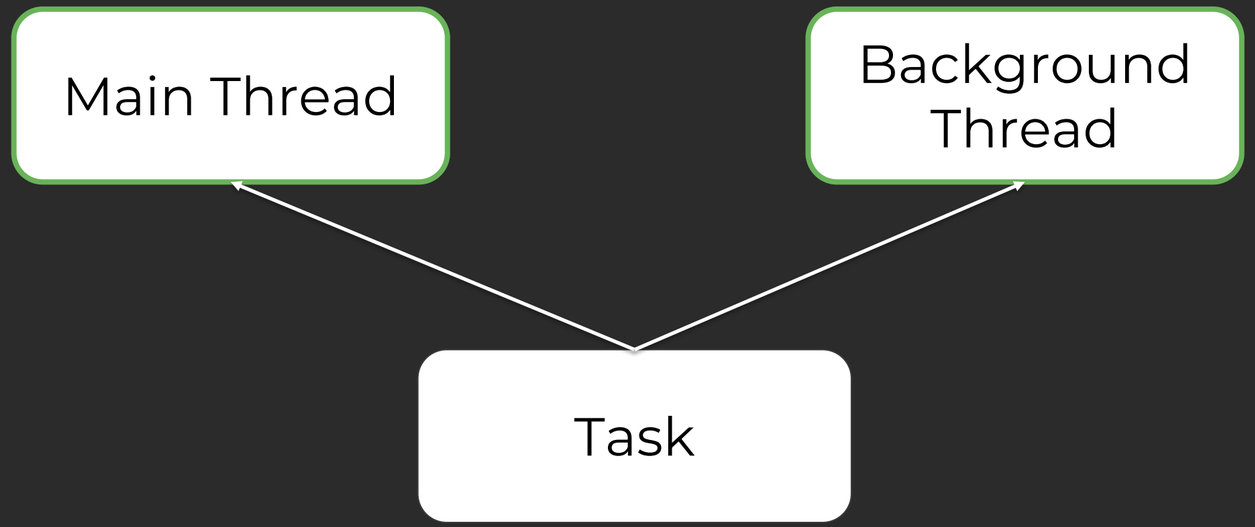
droidcon academy

droidcon academy

# Advanced Usage: Threading and Chaining in Work Manager

# Threading

- What is Threading




droidcon academy

# Threading

- Types of threading mechanisms
  - Handlers
  - Threads
  - Executors
  - RxJava
  - Coroutines

droidcon academy

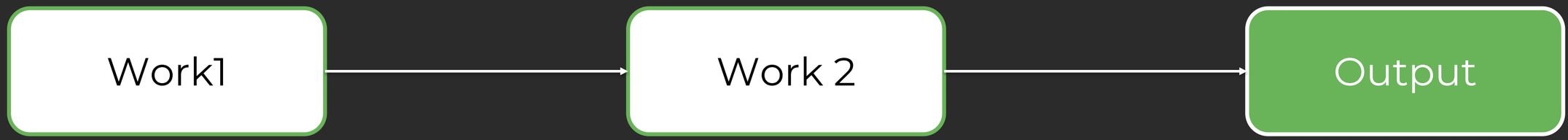
# Threading

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

droidcon academy

# Chaining

- What is chaining ?




droidcon academy

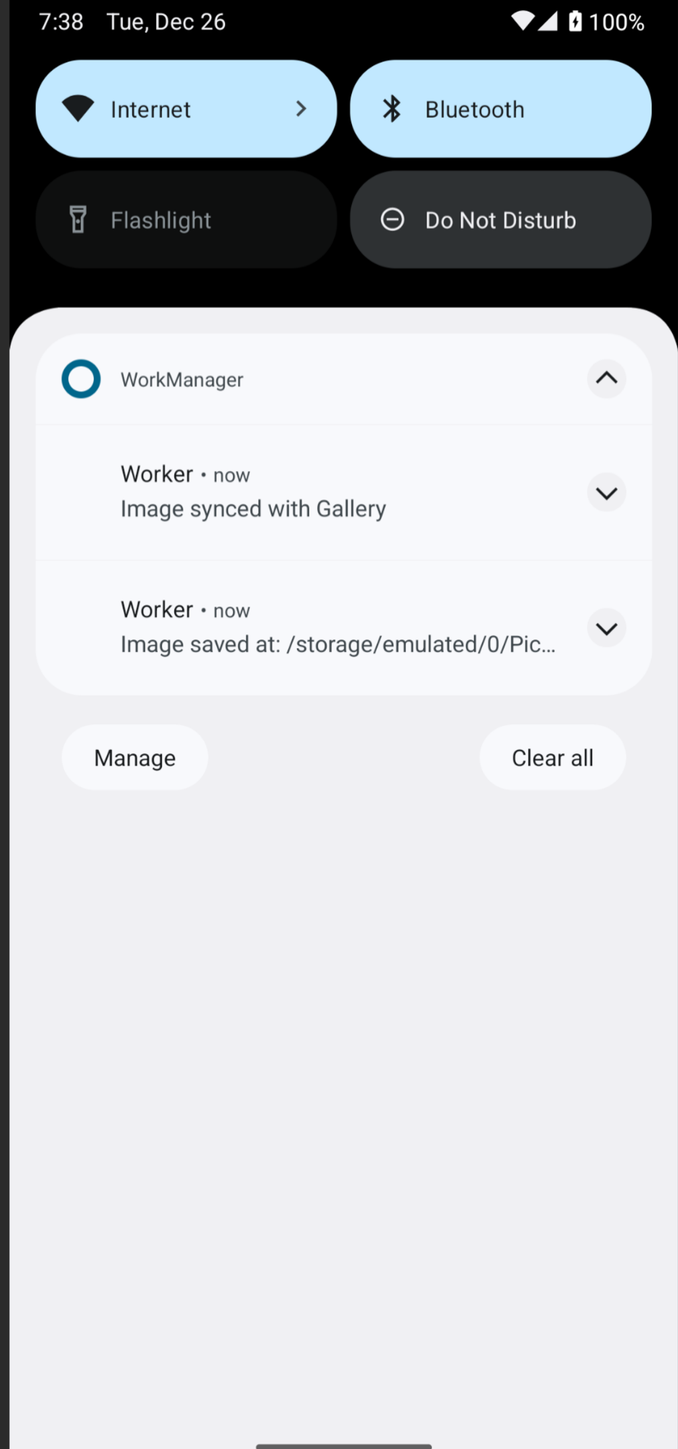
# Chaining

- Why is chaining used in Workers ?
- How to chain multiple works together ?

droidcon academy

# Chaining

- Demo App



# Up Next

## Advanced Usage: Retrying Failed Work, Setting Constraints for Work and Starting Work Periodically

droidcon academy

droidcon academy

# Advanced Usage: Retrying Failed Work, Setting Constraints for Work and Starting Work Periodically

# Overview

- Retrying a Work
- Constraints In WorkManager
- Periodic Work

droidcon academy

# Retrying a Work

- What does Retrying a Work means ?


- How to Retry a Failed Work ?

# Constraints In WorkManager

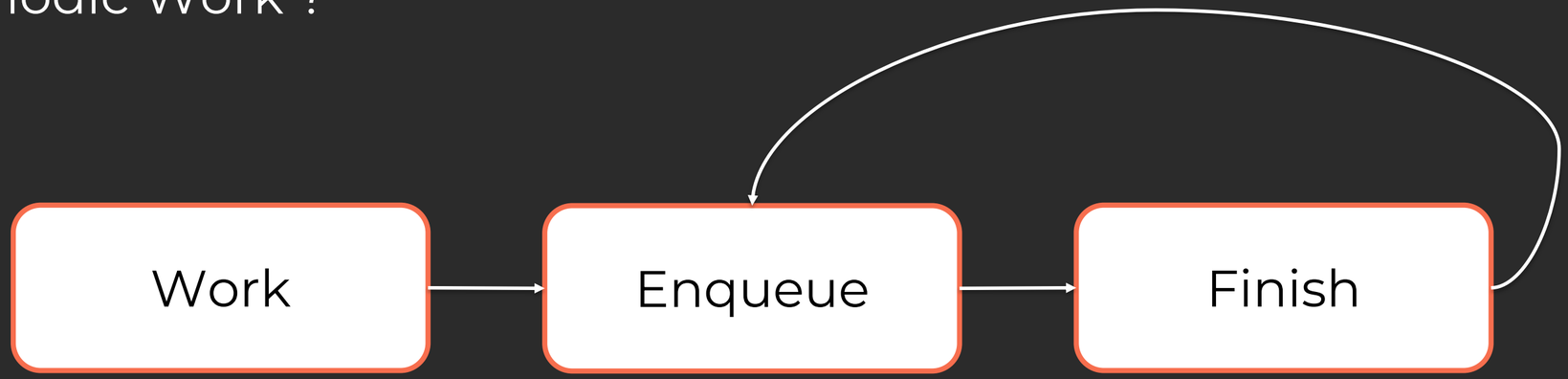
- What are Constraints


- 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

droidcon academy

# 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.

droidcon academy

# 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.
  - Setting constraints on periodic work can miss the worker to start at specified interval.

droidcon academy

# Up Next

## Advanced Usage: Managing Work and Cancelling Work

droidcon academy

droidcon academy

# Advanced Usage: Managing Work and Cancelling Work

# Managing Work

- What is Managing a Work ?
  - Adding Parameters to Work to Identify Work and to Perform Actions on Work.

- Uses
  - Current Status of the Work.
  - Progress of the Work

- Ways to Manage Work.
  - Work Id
  - Work Tag
  - Unique Name
    - Existing Work Policy
      - REPLACE
      - KEEP
      - APPEND
      - APPEND_OR_REPLACE

# Cancelling Work

- Ways to cancel a Work
  - cancelWorkById(WorkId)
  - cancelUniqueWork(UniqueName)
  - cancelAllWorkByTag(Tag)
  - cancelAllWork()

# Cancelling Work

- Points To Remember
  - There is no guarantee that a Work will be cancelled.
  - An already started Work will not be cancelled.

droidcon academy

# Up Next

## Advanced Usage: Observing Work Progress and State

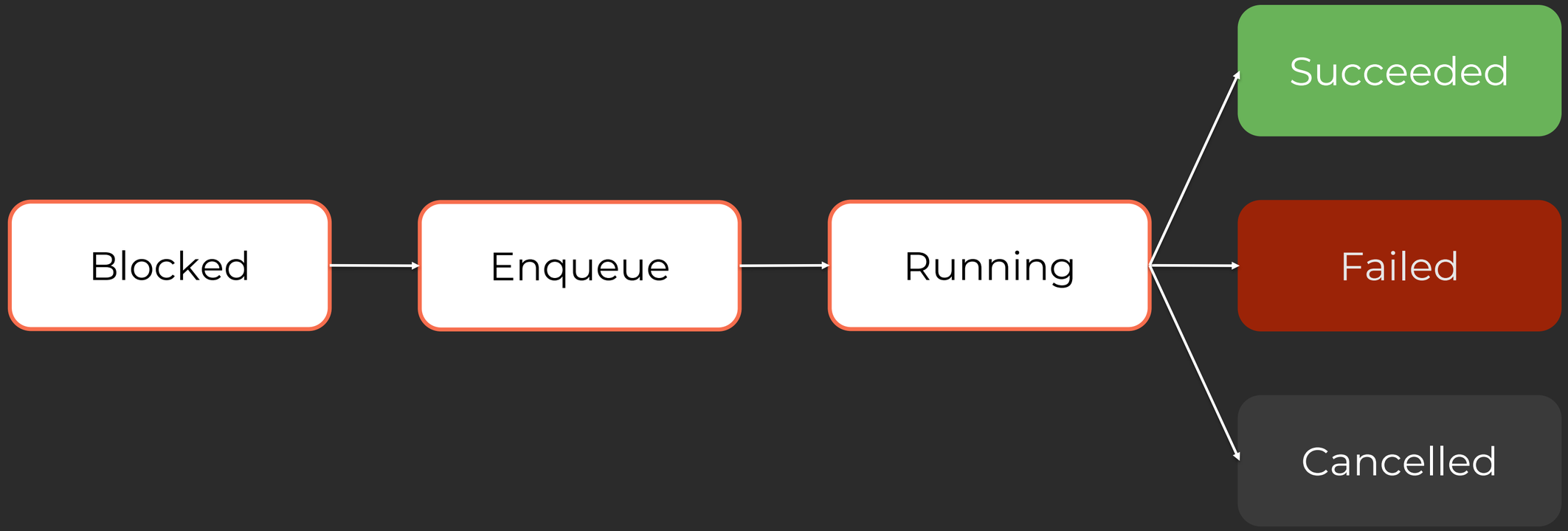
droidcon academy

droidcon academy

# Advanced Usage: Observing Work Progress and State

# Observing a Work

- When and Why we need to Observe for Work state




droidcon academy

# 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)
```

droidcon academy

# Coding Challenge Solution

droidcon academy

# Up Next

## Advanced Usage: Expedited Work and Foreground Work

droidcon academy

droidcon academy

# Advanced Usage: Expedited Work and Foreground Work

# Expedited Work

- What is Expedited Work and When to use it ?
  - Starting a time-critical work that needs to start immediately.

- Out Of Quota Policy
  - DROP_WORK_REQUEST
  - RUN_AS_NON_EXPEDITED_WORK_REQUEST

- Limitations
  - System allocated execution time quota for Expedited Jobs.

droidcon academy

# Foreground Work

- What is Foreground Work ?
  - When the worker is started with a notification till the time it is running.

- When to use Foreground Work ?
  - Running a long task
  - Informing user about the progress

# 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
- Managing and Cancelling Work
- Observing Task Progress and Status
- Periodic Work
- Expedited and Foreground Work

droidcon academy

droidcon academy

# Thank You!


## Baljeet Singh

@yetanotherdev_

in/devbaljeet