A dispatcher determines which thread or thread pool the coroutine runs on

A dispatcher determines which thread or thread pool the coroutine runs on

Different dispatchers are available depending on the task specificity

A dispatcher determines which thread or thread pool the coroutine runs on

Different dispatchers are available depending on the task specificity

launch(Dispatchers.Default) {
 // do some CPU intensive processing task here
}

Common dispatchers:

• Main

Main thread update in UI driven applications (e.g. Android) Main dispatcher needs to be defined in Gradle

Common dispatchers:

• Main

Main thread update in UI driven applications (e.g. Android) Main dispatcher needs to be defined in Gradle

• Default

Useful for CPU intensive work

Common dispatchers:

• Main

Main thread update in UI driven applications (e.g. Android) Main dispatcher needs to be defined in Gradle

• Default

Useful for CPU intensive work

• 10

Useful for network communication or reading/writing files

Common dispatchers:

• Main

Main thread update in UI driven applications (e.g. Android) Main dispatcher needs to be defined in Gradle

• Default

Useful for CPU intensive work

• 10

Useful for network communication or reading/writing files

• Unconfined

Starts the coroutine in the inherited dispatcher that called it

Common dispatchers:

• Main

Main thread update in UI driven applications (e.g. Android) Main dispatcher needs to be defined in Gradle

• Default

Useful for CPU intensive work

• 10

Useful for network communication or reading/writing files

• Unconfined

Starts the coroutine in the inherited dispatcher that called it

newSingleThreadContext("MyThread")
 Forces creation of a new thread