Understanding Coroutine Contexts



Kevin Jones

@kevinrjones www.rocksolidknowledge.com



Coroutine Context



All coroutines run as part of a context

- Created by the launcher

Context can flow to 'child' coroutines

Different contexts supplied

Contexts can be joined



Dispatchers

Contexts provide a coroutine dispatcher

Determines which thread the coroutine is run on

Coroutines can run on:

- Pool thread
- Unconfined thread
- Specific thread



Can Specify Context in Coroutine Builder

Unconfined

coroutineContext

CommonPool

new Single Thread Context

DefaultDispatcher





Specifying the context



CommonPool

The fork/join pool, which is the default pool in the current implementation



coroutineContext

Inherit the context of the current coroutine



DefaultDispatcher

The fork/join pool, which is the default pool in the current implementation



newSingleThreadContext

Runs the coroutine on a new thread. This is an expensive operation. The new thread needs to be managed by your code



Unconfined

Starts the coroutine in the calling thread but only until the first suspension point. Resumes on thread determined by the suspending function





Unconfined context



Debugging Coroutines

Can add a system property when starting

- - Dkotlinx.coroutines.debug

Coroutines can be named





Debugging coroutines



Accessing the 'job'

The current 'job' is in the context

- Use 'Job' as a key





Accessing the job



Parent-child Relationships

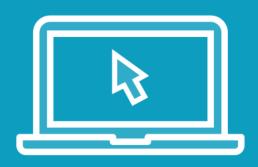
Coroutines can exist in a hierarchy

- If you flow the context any created tasks

Has consequences

- For example, can cancel 'child' tasks





Child coroutines and cancellation



Combining Contexts

Contexts can be 'combined'

- Contexts are maps and combine like maps
- Keys in the left context are replaced by matching values in the right
- Missing keys are not added
- Order may be important





Managing newSingleThreadContext



Be Aware

newSingleThreadContext should be treated with care

- Should close the thread when finished with the context





Managing newSingleThreadContext



Summary



Contexts are one of the things at the heart of coroutines

- May flow from parent to child
- Determines how and where child executes
- Care may need to be taken with contexts
- Can be combined



What's Next



