

# Using Coroutines in UI Applications

---



**Kevin Jones**

@kevinrjones [www.rocksolidknowledge.com](http://www.rocksolidknowledge.com)



# The Problem with UIs



**All UIs require that controls are accessed on main thread**

- Does not require synchronization

**What about long running tasks in the UI**

- Getting data from the network
- Writing to disk
- Calculations

**Want to perform these away from the main thread**

- Better performance
- But have to come back to UI thread to update

**Equally as true for Android and JavaFx**



```
compile "org.jetbrains.kotlinx:kotlinx-coroutines-  
javaafx:$kotlin_coroutines_javaafx_version"
```

```
import kotlinx.coroutines.experimental.javaafx.JavaFx as UI
```

## Coroutines in JavaFx

**Import the correct library**



```
compile "org.jetbrains.kotlinx:kotlinx-coroutines-  
android:$kotlin_coroutines_javafx_version"
```

```
import kotlinx.coroutines.experimental.android.UI
```

## Coroutines in Android

**Import the correct library**



```
launch(UI) { increment() }
```

```
suspend fun increment() {  
    delay(5000)  
    counter.value += 1  
}
```

Use the 'launch' coroutine builder

**In UI context**



# Demo



## Using coroutines in the UI



# Summary



**Kotlin coroutines make it easy to run work in the background**

- Use the `launch(UI)` coroutine builder

