

Async programming in Kotlin with coroutines

Victor Kropp

@kropp

victor.kropp.name

Riga
DevDays



Statically typed programming language for modern multiplatform applications

100% interoperable with Java™ and Android™

TRY KOTLIN

Version 1.2.41

Kotlin 1.2 available now [read more](#)

Build Applications For



JVM



Android



Browser



Native

<https://kotlinlang.org>

Async programming

Callback Hell

```
        });  
    });  
});  
});  
});  
});  
});  
});  
});  
});
```



**shared
mutable
state**

Coroutines

Lightweight threads

100 000 threads

Lightweight threads

100 000 threads



unable to create new native thread

Lightweight threads

100 000 threads



unable to create new native thread

100 000 coroutines

Lightweight threads

100 000 threads



unable to create new native thread

100 000 coroutines





Kotlin is fun

suspend fun

suspend fun

```
suspend fun <T> Call<T>.await(): T {
```

```
}
```

suspend fun

```
suspend fun <T> Call<T>.await(): T {  
    return suspendCoroutine { continuation ->
```

```
}
```

```
}
```

suspend fun

```
suspend fun <T> Call<T>.await(): T {  
    return suspendCoroutine { continuation ->  
        enqueue(object : Callback<T> {
```

```
        })  
    }  
}
```

suspend fun

```
suspend fun <T> Call<T>.await(): T {  
    return suspendCoroutine { continuation ->  
        enqueue(object : Callback<T> {  
            override fun onResponse(call: Call<T>, response: Response<T>) {  
                continuation.resume(response.body())  
            }  
        })  
    }  
}
```


suspend fun

```
suspend fun <T> Call<T>.await(): T {  
    return suspendCoroutine { continuation ->  
        enqueue(object : Callback<T> {  
            override fun onResponse(call: Call<T>, response: Response<T>) {  
                continuation.resume(response.body())  
            }  
            override fun onFailure(call: Call<T>, t: Throwable) {  
                continuation.resumeWithException(t)  
            }  
        })  
    }  
}
```

suspend fun

```
suspend fun func1() {  
    func2()  
    delay(1000)  
    func3()  
}
```

```
suspend fun func2() {  
    func3()  
}
```

```
suspend fun func3() {  
}
```

suspend fun

```
suspend fun func1() {  
->    func2()  
->    delay(1000)  
->    func3()  
}
```

```
suspend fun func2() {  
->    func3()  
}
```

```
suspend fun func3() {  
}
```

Coroutine builders

```
Launch {  
}
```

```
async {  
}
```

```
runBlocking {  
}
```

Coroutine builders

```
Launch {  
}
```

 → Job

```
async {  
}
```

```
runBlocking {  
}
```

Coroutine builders

```
launch {  
}
```

 → Job

```
async {  
}
```

 → Deferred<T>

```
runBlocking {  
}
```

Coroutine builders

launch {
} → Job

async {
} → Deferred<T>

runBlocking {
} → T

Coroutine builders

```
launch(Swing) {  
}
```

```
async(CommonPool) {  
}
```

```
runBlocking(Unconfined) {  
}
```


Coroutine context

```
async(CoroutineName("test")) {  
}
```

Coroutine context

```
async(CommonPool + CoroutineName("test")) {  
}
```

Cancellation

Combine coroutines

Channels

Bonus: Generator

```
fun fibonacci() = buildSequence {  
    var previous = 0  
    var current = 1  
    while (true) {  
        yield(current)  
        val next = previous + current  
        previous = current  
        current = next  
    }  
}
```

Bonus 2: Mutex

```
val mutex = kotlinx.coroutines.experimental.sync.Mutex()
```

```
mutex.withLock { // suspending function
```

```
    ...
```

```
}
```

Experimental?

```
compile 'org.jetbrains.kotlinx:kotlinx-coroutines-core:0.22.5'
```

```
kotlin {  
    experimental {  
        coroutines = 'enable'  
    }  
}
```

build.gradle



Kotlin

ktor.io



Links

`kotlinx.coroutines`

<https://github.com/Kotlin/kotlinx.coroutines>

Coroutines Design Document (KEEP)

<https://github.com/Kotlin/kotlin-coroutines/blob/master/kotlin-coroutines-informal.md>

Coroutines by example

<https://github.com/Kotlin/kotlinx.coroutines/blob/master/coroutines-guide.md>

Thank you!

Victor Kropp

@kropp

victor.kropp.name

Questions?

Victor Kropp

@kropp

victor.kropp.name