https://jagstalk.page.link/examples

```
fun main() = runBlocking {
    val orders = listOf(...)

    val ordersChannel = Channel<Menu>()
    orders.forEach { ordersChannel.send(it) }

    val time = measureTimeMillis {
        coroutineScope {
            launch(CoroutineName("barista-1")) { processOrders(ordersChannel) }
            launch(CoroutineName("barista-2")) { processOrders(ordersChannel) }
        }
    }
    log("time: $time ms”)
}


private suspend fun processOrders(ordersChannel: Channel<Menu.Cappuccino>) {
    ordersChannel.consumeEach {
        val groundBeans = grindCoffeeBeans(it.beans())
        val espresso = pullEspressoShot(groundBeans)
        val steamedMilk = steamMilk(it.milk())
        val cappuccino = makeCappuccino(it, espresso, steamedMilk)
        log("serve: $cappuccino")
    }
}
```

https://jagstalk.page.link/examples

```
fun main() = runBlocking {
    val orders = listOf(...)
    val ordersChannel = Channel<Menu.Cappuccino>()
    launch(CoroutineName("cashier")) {
        orders.forEach { ordersChannel.send(it) }
    }
    val time = measureTimeMillis {
        coroutineScope {
            launch(CoroutineName("barista-1")) { processOrders(ordersChannel) }
            launch(CoroutineName("barista-2")) { processOrders(ordersChannel) }
        }
    }
    log("time: $time ms")
}


private suspend fun processOrders(ordersChannel: Channel<Menu.Cappuccino>) {
    ordersChannel.consumeEach {
        val groundBeans = grindCoffeeBeans(it.beans())
        val espresso = pullEspressoShot(groundBeans)
        val steamedMilk = steamMilk(it.milk())
        val cappuccino = makeCappuccino(it, espresso, steamedMilk)
        log("serve: $cappuccino")
    }
}
```