

<https://jagstalk.page.link/examples>

```
fun main() {...}

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

private suspend fun grindCoffeeBeans(beans: CoffeeBean): CoffeeBean.GroundBeans {
    ...
}

private suspend fun pullEspressoShot(groundBeans: CoffeeBean.GroundBeans): Espresso {
    ...
}

private suspend fun steamMilk(milk: Milk): Milk.SteamedMilk {
    ...
}

private suspend fun makeCappuccino(order: Menu.Cappuccino, espresso: Espresso,
    steamedMilk: Milk.SteamedMilk): Beverage.Cappuccino {
    ...
}
```

<https://jagstalk.page.link/examples>

```
fun main() = runBlocking {
    val orders = listOf(...)
    val time = measureTimeMillis {
        launch(CoroutineName("barista-1")) { processOrders(orders) }
        launch(CoroutineName("barista-2")) { processOrders(orders) }
    }
    log("time: $time ms")
}
```

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

```
private suspend fun grindCoffeeBeans(beans: CoffeeBean): CoffeeBean.GroundBeans {
    ...
}
```

```
private suspend fun pullEspressoShot(groundBeans: CoffeeBean.GroundBeans): Espresso {
    ...
}
```

```
private suspend fun steamMilk(milk: Milk): Milk.SteamedMilk {
    ...
}
```

```
private suspend fun makeCappuccino(order: Menu.Cappuccino, espresso: Espresso,
    steamedMilk: Milk.SteamedMilk): Beverage.Cappuccino {
    ...
}
```