


```
class EspressoMachine {
    private val portafilterOne: SendChannel<CoffeeBean.GroundBeans> = actor {...}


    private val portafilterTwo: SendChannel<CoffeeBean.GroundBeans> = actor {...}

    suspend fun pullEspressoShot(groundBeans: CoffeeBean.GroundBeans): Espresso {
        return select {
            portafilterOne.onSend(groundBeans) {
                TODO("sent to portafilter one - wait for result")
            }
            portafilterTwo.onSend(groundBeans) {
                TODO("sent to portafilter two - wait for result")
            }
        }
    }
    ...
}
```

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



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

```
class EspressoMachine {  
    private val portafilterOne: SendChannel<CoffeeBean.GroundBeans> = actor {...}  
  
    private val portafilterTwo: SendChannel<CoffeeBean.GroundBeans> = actor {...}  
  
    suspend fun pullEspressoShot(groundBeans: CoffeeBean.GroundBeans): Espresso {  
        return select {  
            portafilterOne.onSend(groundBeans) {  
                 TODO("sent to portafilter one - wait for result")  
            }  
            portafilterTwo.onSend(groundBeans) {  
                TODO("sent to portafilter two - wait for result")  
            }  
        }  
    }  
    ...  
}
```

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

```
class EspressoMachine {
    private data class EspressoShotRequest(
        val deferredEspressoShot: CompletableDeferred<Espresso>,
        val groundBeans: CoffeeBean.GroundBeans
    )

    private val portafilterOne: SendChannel<CoffeeBean.GroundBeans> = actor {...}

    private val portafilterTwo: SendChannel<CoffeeBean.GroundBeans> = actor {...}

    suspend fun pullEspressoShot(groundBeans: CoffeeBean.GroundBeans): Espresso {
        return select {
            portafilterOne.onSend(groundBeans) {
                TODO("sent to portafilter one - wait for result")
            }
            portafilterTwo.onSend(groundBeans) {
                TODO("sent to portafilter two - wait for result")
            }
        }
    }
    ...
}
```