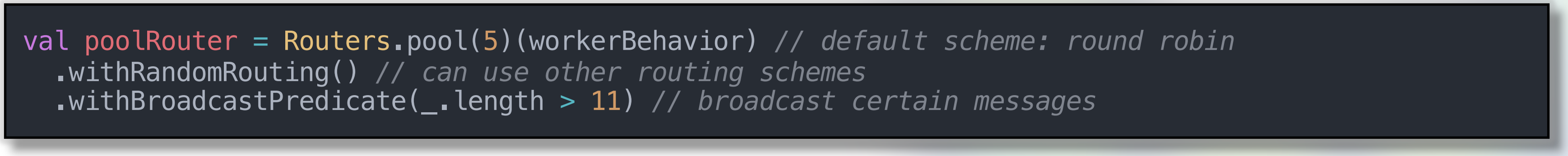
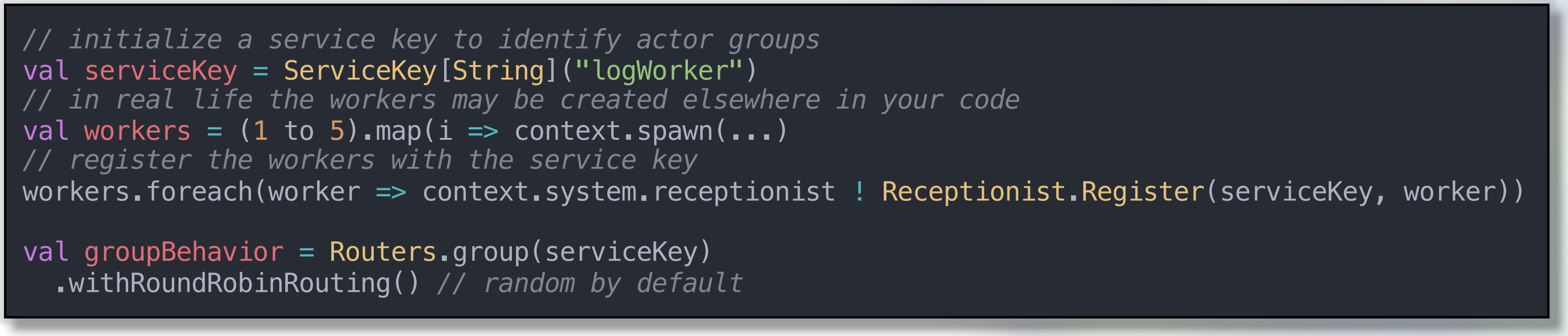
Routers

Routers

Distribute work in between many actors



```scala
val poolRouter = Routers.pool(5)(workerBehavior) // default scheme: round robin
  .withRandomRouting() // can use other routing schemes
  .withBroadcastPredicate(_.length > 11) // broadcast certain messages
```



```scala
// initialize a service key to identify actor groups
val serviceKey = ServiceKey[String]("logWorker")
// in real life the workers may be created elsewhere in your code
val workers = (1 to 5).map(i => context.spawn(...))
// register the workers with the service key
workers.foreach(worker => context.system.receptionist ! Receptionist.Register(serviceKey, worker))

val groupBehavior = Routers.group(serviceKey)
  .withRoundRobinRouting() // random by default
```

Pros/cons

- pool routers: easy to use
- group routers: harder to set up, more flexible at runtime, hard to remove routees

Akka rocks