```scala
/**
  * Created by psc on 14/7/17.
  */

package au.com.carringbushsw.Akka.InvestigatorManager

import MessageTag._

import akka.actor._
import akka.event._
import akka.actor.OneForOneStrategy
import akka.actor.SupervisorStrategy._
import scala.concurrent.duration._

// messages
sealed case class Search(msg: MessageTag)
object Search { // companion object
  val name = "Search"
}
// duplicate messages
sealed case class Request(msg: MessageTag)
object Request { // companion object
  val name = "Request"
}

object InvestigatorManager { // companion object
  val name = "InvestigatorManager"
}

class InvestigatorManager extends Actor {
  val name = "InvestigatorManager"
  val log = Logging(context.system, this)

  override val supervisorStrategy =
    OneForOneStrategy(
      maxNrOfRetries = 5, withinTimeRange = Duration.create("1 minute")) {
      case _: NullPointerException        => Restart
      case _: ArithmeticException         => Resume
      case _: IllegalArgumentException    => Stop
      case _: UnsupportedOperationException => Stop
      case _: Exception                   => Escalate
    }

  override def preStart: Unit = {
    log.info(s"${InvestigatorManager.name}.preStart")
    context.become(search)
  }

  override def postStop: Unit = log.info(s"${InvestigatorManager.name}.postStop")

  def receive = search

  def search: Receive = {
    case Search(MessageTag.search) => {
      log.info(s"${Search.name}('${MessageTag.search}...')")
      // search for
      // requesterManager at "akka.tcp://TestAkkaSystem@127.0.0.1:2552/user/RequesterManager"
      // responderManager at "akka.tcp://ResponderAkkaSystem@127.0.0.1:2553/user/ResponderManager"
      TheFatController.requesterManager ! Request(MessageTag.request)
    }

    case _ => log.info(s"${InvestigatorManager.name}.${Search.name}.help!!")

  } // search: Receive
} // InvestigatorManager
```

common method (handwritten annotation)