package com.twitter.product\_mixer.core.functional\_component.gate

/\*\*

\* A [[Gate]] controls if a pipeline or other component is executed.

\*

\* Application logic should usually use `GateResult.continue: Boolean` to interpret a GateResult. `continue` will be

\* true if we should continue with execution, and false if we should stop.

\*

\* You can case match against the `GateResult` to understand how exactly execution happened. See `object GateResult`

\* below, but this is useful if you want to know if we are continuing due to the skip or main predicates.

\*/

sealed trait GateResult {

/\*\* Should we continue? \*/

val continue: Boolean

}

object GateResult {

/\*\*

\* Continue Execution

\*

\* the Skip predicate evaluated to true,

\* so we Skipped execution of the Main predicate and should continue

\*/

case object Skipped extends GateResult {

override val continue = true

}

/\*\*

\* Continue Execution

\*

\* the main predicate evaluated to true

\*/

case object Continue extends GateResult {

override val continue = true

}

/\*\*

\* Stop execution

\*

\* the main predicate evaluated to false

\*/

case object Stop extends GateResult {

override val continue = false

}

}