package com.twitter.product\_mixer.core.pipeline

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.MisconfiguredFeatureMapFailure

import com.twitter.product\_mixer.core.pipeline.pipeline\_failure.PipelineFailureCategory

/\*\*

\* [[FailOpenPolicy]] determines what should happen in the event that a candidate pipeline fails

\* to execute successfully.

\*

\* Exercise caution when creating new fail open policies. Product Mixer will fail open by default in

\* certain error cases (e.g. closed gate on a candidate pipeline) but these might inadvertently be

\* excluded by a new policy.

\*/

trait FailOpenPolicy {

def apply(failureCategory: PipelineFailureCategory): Boolean

}

object FailOpenPolicy {

/\*\*

\* Always fail open on candidate pipeline failures except

\* for [[MisconfiguredFeatureMapFailure]]s because it's a programmer error

\* and should always fail loudly, even with an [[Always]] p[[FailOpenPolicy]]

\*/

val Always: FailOpenPolicy = (category: PipelineFailureCategory) => {

category != MisconfiguredFeatureMapFailure

}

/\*\*

\* Never fail open on candidate pipeline failures.

\*

\* @note this is more restrictive than the default behavior which is to allow gate closed

\* failures.

\*/

val Never: FailOpenPolicy = (\_: PipelineFailureCategory) => false

// Build a policy that will fail open for a given set of categories

def apply(categories: Set[PipelineFailureCategory]): FailOpenPolicy =

(failureCategory: PipelineFailureCategory) =>

categories

.contains(failureCategory)

}