package com.twitter.visibility.rules

import com.twitter.visibility.features.Feature

import com.twitter.visibility.rules.State.FeatureFailed

import com.twitter.visibility.rules.State.MissingFeature

import com.twitter.visibility.rules.State.RuleFailed

abstract class FailClosedException(message: String, state: State, ruleName: String)

extends Exception(message) {

def getState: State = {

state

}

def getRuleName: String = {

ruleName

}

}

case class MissingFeaturesException(

ruleName: String,

missingFeatures: Set[Feature[\_]])

extends FailClosedException(

s"A $ruleName rule evaluation has ${missingFeatures.size} missing features: ${missingFeatures

.map(\_.name)}",

MissingFeature(missingFeatures),

ruleName) {}

case class FeaturesFailedException(

ruleName: String,

featureFailures: Map[Feature[\_], Throwable])

extends FailClosedException(

s"A $ruleName rule evaluation has ${featureFailures.size} failed features: ${featureFailures.keys

.map(\_.name)}, ${featureFailures.values}",

FeatureFailed(featureFailures),

ruleName) {}

case class RuleFailedException(ruleName: String, exception: Throwable)

extends FailClosedException(

s"A $ruleName rule evaluation failed to execute",

RuleFailed(exception),

ruleName) {}