package com.twitter.visibility.rules

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.servo.util.Gate

import com.twitter.timelines.configapi.HasParams

import com.twitter.timelines.configapi.Params

import com.twitter.visibility.configapi.VisibilityParams

import com.twitter.visibility.models.SafetyLevel

import com.twitter.visibility.models.UnitOfDiversion

import com.twitter.visibility.models.ViewerContext

case class EvaluationContext(

visibilityPolicy: VisibilityPolicy,

params: Params,

statsReceiver: StatsReceiver)

extends HasParams {

def ruleEnabledInContext(rule: Rule): Boolean = {

visibilityPolicy.policyRuleParams

.get(rule)

.filter(\_.ruleParams.nonEmpty)

.map(policyRuleParams => {

(policyRuleParams.force || rule.enabled.forall(params(\_))) &&

policyRuleParams.ruleParams.forall(params(\_))

})

.getOrElse(rule.isEnabled(params))

}

}

object EvaluationContext {

def apply(

safetyLevel: SafetyLevel,

params: Params,

statsReceiver: StatsReceiver

): EvaluationContext = {

val visibilityPolicy = RuleBase.RuleMap(safetyLevel)

new EvaluationContext(visibilityPolicy, params, statsReceiver)

}

case class Builder(

statsReceiver: StatsReceiver,

visibilityParams: VisibilityParams,

viewerContext: ViewerContext,

unitsOfDiversion: Seq[UnitOfDiversion] = Seq.empty,

memoizeParams: Gate[Unit] = Gate.False,

) {

private[this] val emptyContentToUoDCounter =

statsReceiver.counter("empty\_content\_id\_to\_unit\_of\_diversion")

def build(safetyLevel: SafetyLevel): EvaluationContext = {

val policy = RuleBase.RuleMap(safetyLevel)

val params = if (memoizeParams()) {

visibilityParams.memoized(viewerContext, safetyLevel, unitsOfDiversion)

} else {

visibilityParams(viewerContext, safetyLevel, unitsOfDiversion)

}

new EvaluationContext(policy, params, statsReceiver)

}

def withUnitOfDiversion(unitOfDiversion: UnitOfDiversion\*): Builder =

this.copy(unitsOfDiversion = unitOfDiversion)

def withMemoizedParams(memoizeParams: Gate[Unit]) = this.copy(memoizeParams = memoizeParams)

}

}