package com.twitter.visibility.rules

import com.twitter.visibility.configapi.params.RuleParams

import com.twitter.visibility.rules.Condition.And

import com.twitter.visibility.rules.Condition.DmConversationLastReadableEventIdIsValid

import com.twitter.visibility.rules.Condition.DmConversationTimelineIsEmpty

import com.twitter.visibility.rules.Condition.ViewerIsDmConversationParticipant

import com.twitter.visibility.rules.Condition.DmConversationInfoExists

import com.twitter.visibility.rules.Condition.DmConversationTimelineExists

import com.twitter.visibility.rules.Condition.Not

import com.twitter.visibility.rules.Condition.DeactivatedAuthor

import com.twitter.visibility.rules.Condition.ErasedAuthor

import com.twitter.visibility.rules.Condition.OneToOneDmConversation

import com.twitter.visibility.rules.Condition.Or

import com.twitter.visibility.rules.Condition.SuspendedAuthor

import com.twitter.visibility.rules.Reason.Unspecified

object DmConversationRules {

object DropEmptyDmConversationRule

extends RuleWithConstantAction(

Drop(Unspecified),

Or(

Not(DmConversationLastReadableEventIdIsValid),

And(OneToOneDmConversation, DmConversationTimelineIsEmpty))) {

override def enableFailClosed = Seq(RuleParams.True)

}

object DropInaccessibleDmConversationRule

extends RuleWithConstantAction(Drop(Unspecified), Not(ViewerIsDmConversationParticipant)) {

override def enableFailClosed = Seq(RuleParams.True)

}

object DropDmConversationWithUndefinedConversationInfoRule

extends RuleWithConstantAction(Drop(Unspecified), Not(DmConversationInfoExists)) {

override def enableFailClosed = Seq(RuleParams.True)

}

object DropDmConversationWithUndefinedConversationTimelineRule

extends RuleWithConstantAction(Drop(Unspecified), Not(DmConversationTimelineExists)) {

override def enableFailClosed = Seq(RuleParams.True)

}

object DropOneToOneDmConversationWithUnavailableParticipantsRule

extends RuleWithConstantAction(

Drop(Unspecified),

And(OneToOneDmConversation, Or(SuspendedAuthor, DeactivatedAuthor, ErasedAuthor))) {

override def enableFailClosed = Seq(RuleParams.True)

}

}