package com.twitter.visibility.rules

import com.twitter.visibility.configapi.params.RuleParam

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

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

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

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

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

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

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

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

import com.twitter.visibility.rules.Condition.IsQuotedInnerTweet

import com.twitter.visibility.rules.Condition.NonAuthorViewer

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

import com.twitter.visibility.rules.Condition.Retweet

import com.twitter.visibility.rules.Condition.SoftViewer

import com.twitter.visibility.rules.Reason.\_

object DropAllRule

extends AlwaysActRule(

Drop(Unspecified)

)

object AllowAllRule

extends AlwaysActRule(

Allow

)

object TestRule

extends AlwaysActRule(

Drop(Unspecified)

)

object DeactivatedAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(DeactivatedAuthor),

Condition.DeactivatedAuthor

)

object ErasedAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(ErasedAuthor),

Condition.ErasedAuthor

)

object OffboardedAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(OffboardedAuthor),

Condition.OffboardedAuthor

)

object DropNsfwUserAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Nsfw),

Condition.NsfwUserAuthor

)

object DropNsfwUserAuthorViewerOptInFilteringOnSearchRule

extends ViewerOptInFilteringOnSearchRule(

Drop(Nsfw),

Condition.NsfwUserAuthor

)

object InterstitialNsfwUserAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Nsfw),

Condition.NsfwUserAuthor

)

object DropNsfwAdminAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Nsfw),

Condition.NsfwAdminAuthor

)

object DropNsfwAdminAuthorViewerOptInFilteringOnSearchRule

extends ViewerOptInFilteringOnSearchRule(

Drop(Nsfw),

Condition.NsfwAdminAuthor

)

object InterstitialNsfwAdminAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Nsfw),

Condition.NsfwAdminAuthor

)

object ProtectedAuthorDropRule

extends RuleWithConstantAction(

Drop(Reason.ProtectedAuthor),

And(Condition.LoggedOutOrViewerNotFollowingAuthor, Condition.ProtectedAuthor)

)

object ProtectedAuthorTombstoneRule

extends RuleWithConstantAction(

Tombstone(Epitaph.Protected),

And(Condition.LoggedOutOrViewerNotFollowingAuthor, Condition.ProtectedAuthor)

)

object DropAllProtectedAuthorRule

extends RuleWithConstantAction(

Drop(Reason.ProtectedAuthor),

Condition.ProtectedAuthor

) {

override def enableFailClosed: Seq[RuleParam[Boolean]] = Seq(RuleParams.True)

}

object ProtectedQuoteTweetAuthorRule

extends RuleWithConstantAction(

Drop(Reason.ProtectedAuthor),

And(Condition.OuterAuthorNotFollowingAuthor, Condition.ProtectedAuthor)

)

object DropProtectedViewerIfPresentRule

extends RuleWithConstantAction(

Drop(Reason.Unspecified),

And(Condition.LoggedInViewer, Condition.ProtectedViewer)

) {

override def enableFailClosed: Seq[RuleParam[Boolean]] = Seq(RuleParams.True)

}

object SuspendedAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(SuspendedAuthor),

Condition.SuspendedAuthor

)

object SuspendedViewerRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Unspecified),

Condition.SuspendedViewer

)

object DeactivatedViewerRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Unspecified),

Condition.DeactivatedViewer

)

object ViewerIsUnmentionedRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.ViewerIsUnmentioned),

Condition.ViewerIsUnmentioned

)

abstract class AuthorBlocksViewerRule(override val action: Action)

extends OnlyWhenNotAuthorViewerRule(

action,

Condition.AuthorBlocksViewer

)

object AuthorBlocksViewerDropRule

extends AuthorBlocksViewerRule(

Drop(Reason.AuthorBlocksViewer)

)

object DeciderableAuthorBlocksViewerDropRule

extends AuthorBlocksViewerRule(

Drop(Reason.AuthorBlocksViewer)

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableAuthorBlocksViewerDropRuleParam)

}

object AuthorBlocksViewerTombstoneRule

extends AuthorBlocksViewerRule(

Tombstone(Epitaph.BlockedBy)

)

object ViewerBlocksAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.ViewerBlocksAuthor),

Condition.ViewerBlocksAuthor

)

object ViewerBlocksAuthorViewerOptInBlockingOnSearchRule

extends ViewerOptInBlockingOnSearchRule(

Drop(Reason.ViewerBlocksAuthor),

Condition.ViewerBlocksAuthor

)

object ViewerMutesAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.ViewerMutesAuthor),

Condition.ViewerMutesAuthor

)

object ViewerMutesAuthorViewerOptInBlockingOnSearchRule

extends ViewerOptInBlockingOnSearchRule(

Drop(Reason.ViewerMutesAuthor),

Condition.ViewerMutesAuthor

)

object AuthorBlocksOuterAuthorRule

extends RuleWithConstantAction(

Drop(Reason.AuthorBlocksViewer),

And(Not(Condition.IsSelfQuote), Condition.AuthorBlocksOuterAuthor)

)

object ViewerMutesAndDoesNotFollowAuthorRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.ViewerHardMutedAuthor),

And(Condition.ViewerMutesAuthor, Not(Condition.ViewerDoesFollowAuthor))

)

object AuthorBlocksViewerUnspecifiedRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.Unspecified),

Condition.AuthorBlocksViewer

)

object ViewerHasMatchingMutedKeywordForNotificationsRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.MutedKeyword),

Condition.ViewerHasMatchingKeywordForNotifications

)

object ViewerHasMatchingMutedKeywordForHomeTimelineRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.MutedKeyword),

Condition.ViewerHasMatchingKeywordForHomeTimeline

)

trait HasPromotedTweetHealthEnforcement extends WithGate {

override def holdbacks: Seq[RuleParam[Boolean]] = Seq(PromotedTweetHealthEnforcementHoldback)

override def enabled: Seq[RuleParam[Boolean]] = Seq(

EnableTimelineHomePromotedTweetHealthEnforcementRules)

}

object ViewerHasMatchingMutedKeywordForHomeTimelinePromotedTweetRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.MutedKeyword),

Condition.ViewerHasMatchingKeywordForHomeTimeline

)

with HasPromotedTweetHealthEnforcement

object ViewerHasMatchingMutedKeywordForTweetRepliesRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.MutedKeyword),

Condition.ViewerHasMatchingKeywordForTweetReplies

)

object MutedKeywordForTweetRepliesInterstitialRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Reason.MutedKeyword),

Condition.ViewerHasMatchingKeywordForTweetReplies

)

object MutedKeywordForQuotedTweetTweetDetailInterstitialRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Reason.MutedKeyword),

And(Condition.IsQuotedInnerTweet, Condition.ViewerHasMatchingKeywordForTweetReplies)

)

object ViewerMutesAuthorInterstitialRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Reason.ViewerMutesAuthor),

Condition.ViewerMutesAuthor

)

object ViewerMutesAuthorInnerQuotedTweetInterstitialRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Reason.ViewerMutesAuthor),

And(Condition.ViewerMutesAuthor, IsQuotedInnerTweet)

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableInnerQuotedTweetViewerMutesAuthorInterstitialRuleParam)

}

object ViewerMutesAuthorHomeTimelinePromotedTweetRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.ViewerMutesAuthor),

Condition.ViewerMutesAuthor

)

with HasPromotedTweetHealthEnforcement

object ViewerBlocksAuthorInterstitialRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Reason.ViewerBlocksAuthor),

Condition.ViewerBlocksAuthor

)

object ViewerBlocksAuthorInnerQuotedTweetInterstitialRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Reason.ViewerBlocksAuthor),

And(Condition.ViewerBlocksAuthor, IsQuotedInnerTweet)

) {

override def enabled: Seq[RuleParam[Boolean]] =

Seq(EnableInnerQuotedTweetViewerBlocksAuthorInterstitialRuleParam)

}

object ViewerBlocksAuthorHomeTimelinePromotedTweetRule

extends OnlyWhenNotAuthorViewerRule(

Drop(Reason.ViewerBlocksAuthor),

Condition.ViewerBlocksAuthor

)

with HasPromotedTweetHealthEnforcement

object ViewerReportsAuthorInterstitialRule

extends OnlyWhenNotAuthorViewerRule(

Interstitial(Reason.ViewerReportedAuthor),

Condition.ViewerReportsAuthor

)

object ViewerIsAuthorDropRule

extends RuleWithConstantAction(Drop(Unspecified), Not(NonAuthorViewer))

object ViewerIsNotAuthorDropRule extends RuleWithConstantAction(Drop(Unspecified), NonAuthorViewer)

object RetweetDropRule extends RuleWithConstantAction(Drop(Unspecified), Retweet)

object ViewerIsSoftUserDropRule extends RuleWithConstantAction(Drop(ViewerIsSoftUser), SoftViewer) {

override val enabled: Seq[RuleParam[Boolean]] = Seq(EnableViewerIsSoftUserDropRuleParam)

}