package com.twitter.visibility.interfaces.conversations

import com.twitter.timelines.render.thriftscala.TombstoneDisplayType

import com.twitter.timelines.render.thriftscala.TombstoneInfo

import com.twitter.visibility.rules.\_

case class VfTombstone(

tombstoneId: Long,

includeTweet: Boolean,

action: Action,

tombstoneInfo: Option[TombstoneInfo] = None,

tombstoneDisplayType: TombstoneDisplayType = TombstoneDisplayType.Inline,

truncateDescendantsWhenFocal: Boolean = false) {

val isTruncatable: Boolean = action match {

case Interstitial(Reason.ViewerBlocksAuthor, \_, \_) => true

case Interstitial(Reason.ViewerHardMutedAuthor, \_, \_) => true

case Interstitial(Reason.MutedKeyword, \_, \_) => true

case Tombstone(Epitaph.NotFound, \_) => true

case Tombstone(Epitaph.Unavailable, \_) => true

case Tombstone(Epitaph.Suspended, \_) => true

case Tombstone(Epitaph.Protected, \_) => true

case Tombstone(Epitaph.Deactivated, \_) => true

case Tombstone(Epitaph.BlockedBy, \_) => true

case Tombstone(Epitaph.Moderated, \_) => true

case Tombstone(Epitaph.Deleted, \_) => true

case Tombstone(Epitaph.Underage, \_) => true

case Tombstone(Epitaph.NoStatedAge, \_) => true

case Tombstone(Epitaph.LoggedOutAge, \_) => true

case Tombstone(Epitaph.SuperFollowsContent, \_) => true

case Tombstone(Epitaph.CommunityTweetHidden, \_) => true

case \_: LocalizedTombstone => true

case \_ => false

}

}