package com.twitter.graph\_feature\_service.util

import com.twitter.graph\_feature\_service.thriftscala.EdgeType.\_

import com.twitter.graph\_feature\_service.thriftscala.{FeatureType, PresetFeatureTypes}

object FeatureTypesCalculator {

final val DefaultTwoHop = Seq(

FeatureType(Following, FollowedBy),

FeatureType(Following, FavoritedBy),

FeatureType(Following, RetweetedBy),

FeatureType(Following, MentionedBy),

FeatureType(Following, MutualFollow),

FeatureType(Favorite, FollowedBy),

FeatureType(Favorite, FavoritedBy),

FeatureType(Favorite, RetweetedBy),

FeatureType(Favorite, MentionedBy),

FeatureType(Favorite, MutualFollow),

FeatureType(MutualFollow, FollowedBy),

FeatureType(MutualFollow, FavoritedBy),

FeatureType(MutualFollow, RetweetedBy),

FeatureType(MutualFollow, MentionedBy),

FeatureType(MutualFollow, MutualFollow)

)

final val SocialProofTwoHop = Seq(FeatureType(Following, FollowedBy))

final val HtlTwoHop = DefaultTwoHop

final val WtfTwoHop = SocialProofTwoHop

final val SqTwoHop = DefaultTwoHop

final val RuxTwoHop = DefaultTwoHop

final val MRTwoHop = DefaultTwoHop

final val UserTypeaheadTwoHop = SocialProofTwoHop

final val presetFeatureTypes =

(HtlTwoHop ++ WtfTwoHop ++ SqTwoHop ++ RuxTwoHop ++ MRTwoHop ++ UserTypeaheadTwoHop).toSet

def getFeatureTypes(

presetFeatureTypes: PresetFeatureTypes,

featureTypes: Seq[FeatureType]

): Seq[FeatureType] = {

presetFeatureTypes match {

case PresetFeatureTypes.HtlTwoHop => HtlTwoHop

case PresetFeatureTypes.WtfTwoHop => WtfTwoHop

case PresetFeatureTypes.SqTwoHop => SqTwoHop

case PresetFeatureTypes.RuxTwoHop => RuxTwoHop

case PresetFeatureTypes.MrTwoHop => MRTwoHop

case PresetFeatureTypes.UserTypeaheadTwoHop => UserTypeaheadTwoHop

case \_ => featureTypes

}

}

}