package com.twitter.visibility.models

import com.twitter.visibility.safety\_label\_store.{thriftscala => s}

import com.twitter.visibility.util.NamingUtils

sealed trait SpaceSafetyLabelType extends SafetyLabelType {

lazy val name: String = NamingUtils.getFriendlyName(this)

}

object SpaceSafetyLabelType extends SafetyLabelType {

val List: List[SpaceSafetyLabelType] = s.SpaceSafetyLabelType.list.map(fromThrift)

val ActiveLabels: List[SpaceSafetyLabelType] = List.filter { labelType =>

labelType != Unknown && labelType != Deprecated

}

private lazy val nameToValueMap: Map[String, SpaceSafetyLabelType] =

List.map(l => l.name.toLowerCase -> l).toMap

def fromName(name: String): Option[SpaceSafetyLabelType] = nameToValueMap.get(name.toLowerCase)

private val UnknownThriftSafetyLabelType =

s.SpaceSafetyLabelType.EnumUnknownSpaceSafetyLabelType(UnknownEnumValue)

private lazy val thriftToModelMap: Map[s.SpaceSafetyLabelType, SpaceSafetyLabelType] = Map(

s.SpaceSafetyLabelType.DoNotAmplify -> DoNotAmplify,

s.SpaceSafetyLabelType.CoordinatedHarmfulActivityHighRecall -> CoordinatedHarmfulActivityHighRecall,

s.SpaceSafetyLabelType.UntrustedUrl -> UntrustedUrl,

s.SpaceSafetyLabelType.MisleadingHighRecall -> MisleadingHighRecall,

s.SpaceSafetyLabelType.NsfwHighPrecision -> NsfwHighPrecision,

s.SpaceSafetyLabelType.NsfwHighRecall -> NsfwHighRecall,

s.SpaceSafetyLabelType.CivicIntegrityMisinfo -> CivicIntegrityMisinfo,

s.SpaceSafetyLabelType.MedicalMisinfo -> MedicalMisinfo,

s.SpaceSafetyLabelType.GenericMisinfo -> GenericMisinfo,

s.SpaceSafetyLabelType.DmcaWithheld -> DmcaWithheld,

s.SpaceSafetyLabelType.HatefulHighRecall -> HatefulHighRecall,

s.SpaceSafetyLabelType.ViolenceHighRecall -> ViolenceHighRecall,

s.SpaceSafetyLabelType.HighToxicityModelScore -> HighToxicityModelScore,

s.SpaceSafetyLabelType.DeprecatedSpaceSafetyLabel14 -> Deprecated,

s.SpaceSafetyLabelType.DeprecatedSpaceSafetyLabel15 -> Deprecated,

s.SpaceSafetyLabelType.Reserved16 -> Deprecated,

s.SpaceSafetyLabelType.Reserved17 -> Deprecated,

s.SpaceSafetyLabelType.Reserved18 -> Deprecated,

s.SpaceSafetyLabelType.Reserved19 -> Deprecated,

s.SpaceSafetyLabelType.Reserved20 -> Deprecated,

s.SpaceSafetyLabelType.Reserved21 -> Deprecated,

s.SpaceSafetyLabelType.Reserved22 -> Deprecated,

s.SpaceSafetyLabelType.Reserved23 -> Deprecated,

s.SpaceSafetyLabelType.Reserved24 -> Deprecated,

s.SpaceSafetyLabelType.Reserved25 -> Deprecated,

)

private lazy val modelToThriftMap: Map[SpaceSafetyLabelType, s.SpaceSafetyLabelType] =

(for ((k, v) <- thriftToModelMap) yield (v, k)) ++ Map(

Deprecated -> s.SpaceSafetyLabelType.EnumUnknownSpaceSafetyLabelType(DeprecatedEnumValue),

)

case object DoNotAmplify extends SpaceSafetyLabelType

case object CoordinatedHarmfulActivityHighRecall extends SpaceSafetyLabelType

case object UntrustedUrl extends SpaceSafetyLabelType

case object MisleadingHighRecall extends SpaceSafetyLabelType

case object NsfwHighPrecision extends SpaceSafetyLabelType

case object NsfwHighRecall extends SpaceSafetyLabelType

case object CivicIntegrityMisinfo extends SpaceSafetyLabelType

case object MedicalMisinfo extends SpaceSafetyLabelType

case object GenericMisinfo extends SpaceSafetyLabelType

case object DmcaWithheld extends SpaceSafetyLabelType

case object HatefulHighRecall extends SpaceSafetyLabelType

case object ViolenceHighRecall extends SpaceSafetyLabelType

case object HighToxicityModelScore extends SpaceSafetyLabelType

case object Deprecated extends SpaceSafetyLabelType

case object Unknown extends SpaceSafetyLabelType

def fromThrift(safetyLabelType: s.SpaceSafetyLabelType): SpaceSafetyLabelType =

thriftToModelMap.get(safetyLabelType) match {

case Some(spaceSafetyLabelType) => spaceSafetyLabelType

case \_ =>

safetyLabelType match {

case s.SpaceSafetyLabelType.EnumUnknownSpaceSafetyLabelType(DeprecatedEnumValue) =>

Deprecated

case \_ =>

Unknown

}

}

def toThrift(safetyLabelType: SpaceSafetyLabelType): s.SpaceSafetyLabelType = {

modelToThriftMap

.get(safetyLabelType).getOrElse(UnknownThriftSafetyLabelType)

}

}