package com.twitter.visibility.models

import com.twitter.spam.rtf.{thriftscala => s}

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

case class SafetyLabel(

score: Option[Double] = None,

applicableUsers: Set[Long] = Set.empty,

source: Option[LabelSource] = None,

modelMetadata: Option[TweetModelMetadata] = None,

createdAtMsec: Option[Long] = None,

expiresAtMsec: Option[Long] = None,

labelMetadata: Option[SafetyLabelMetadata] = None,

applicableCountries: Option[Seq[String]] = None)

object SafetyLabel {

def fromThrift(safetyLabel: s.SafetyLabel): SafetyLabel = {

SafetyLabel(

score = safetyLabel.score,

applicableUsers = safetyLabel.applicableUsers

.map { perspectivalUsers =>

(perspectivalUsers map {

\_.userId

}).toSet

}.getOrElse(Set.empty),

source = safetyLabel.source.flatMap(LabelSource.fromString),

modelMetadata = safetyLabel.modelMetadata.flatMap(TweetModelMetadata.fromThrift),

createdAtMsec = safetyLabel.createdAtMsec,

expiresAtMsec = safetyLabel.expiresAtMsec,

labelMetadata = safetyLabel.labelMetadata.map(SafetyLabelMetadata.fromThrift(\_)),

applicableCountries = safetyLabel.applicableCountries

)

}

def toThrift(safetyLabel: SafetyLabel): s.SafetyLabel = {

s.SafetyLabel(

score = safetyLabel.score,

applicableUsers = if (safetyLabel.applicableUsers.nonEmpty) {

Some(safetyLabel.applicableUsers.toSeq.map {

s.PerspectivalUser(\_)

})

} else {

None

},

source = safetyLabel.source.map(\_.name),

modelMetadata = safetyLabel.modelMetadata.map(TweetModelMetadata.toThrift),

createdAtMsec = safetyLabel.createdAtMsec,

expiresAtMsec = safetyLabel.expiresAtMsec,

labelMetadata = safetyLabel.labelMetadata.map(\_.toThrift),

applicableCountries = safetyLabel.applicableCountries

)

}

}

trait SafetyLabelWithType[EntitySafetyLabelType <: SafetyLabelType] {

val safetyLabelType: EntitySafetyLabelType

val safetyLabel: SafetyLabel

}

case class MediaSafetyLabel(

override val safetyLabelType: MediaSafetyLabelType,

override val safetyLabel: SafetyLabel)

extends SafetyLabelWithType[MediaSafetyLabelType] {

def fromThrift(

thriftType: store.MediaSafetyLabelType,

thriftLabel: s.SafetyLabel

): MediaSafetyLabel = {

MediaSafetyLabel(

MediaSafetyLabelType.fromThrift(thriftType),

SafetyLabel.fromThrift(thriftLabel)

)

}

}

case class SpaceSafetyLabel(

override val safetyLabelType: SpaceSafetyLabelType,

override val safetyLabel: SafetyLabel)

extends SafetyLabelWithType[SpaceSafetyLabelType] {

def fromThrift(

thriftType: store.SpaceSafetyLabelType,

thriftLabel: s.SafetyLabel

): SpaceSafetyLabel = {

SpaceSafetyLabel(

SpaceSafetyLabelType.fromThrift(thriftType),

SafetyLabel.fromThrift(thriftLabel)

)

}

}