package com.twitter.tweetypie.util

import com.twitter.takedown.util.TakedownReasons

import com.twitter.takedown.util.TakedownReasons.CountryCode

import com.twitter.tseng.withholding.thriftscala.TakedownReason

import com.twitter.tseng.withholding.thriftscala.UnspecifiedReason

import com.twitter.tweetypie.thriftscala.Tweet

/\*\*

\* Contains tweetypie-specific utils for working with TakedownReasons.

\*/

object Takedowns {

type CountryCode = String

/\*\*

\* Take a list of [[TakedownReason]] and return values to be saved on the [[Tweet]] in fields

\* tweetypieOnlyTakedownCountryCode and tweetypieOnlyTakedownReason.

\*

\* - tweetypieOnlyTakedownCountryCode contains the country\_code of all UnspecifiedReasons

\* - tweetypieOnlyTakedownReason contains all other reasons

\*/

def partitionReasons(reasons: Seq[TakedownReason]): (Seq[String], Seq[TakedownReason]) = {

val (unspecifiedReasons, specifiedReasons) = reasons.partition {

case TakedownReason.UnspecifiedReason(UnspecifiedReason(\_)) => true

case \_ => false

}

val unspecifiedCountryCodes = unspecifiedReasons.collect(TakedownReasons.reasonToCountryCode)

(unspecifiedCountryCodes, specifiedReasons)

}

def fromTweet(t: Tweet): Takedowns =

Takedowns(

Seq

.concat(

t.tweetypieOnlyTakedownCountryCodes

.getOrElse(Nil).map(TakedownReasons.countryCodeToReason),

t.tweetypieOnlyTakedownReasons.getOrElse(Nil)

).toSet

)

}

/\*\*

\* This class is used to ensure the caller has access to both the full list of reasons as well

\* as the backwards-compatible list of country codes.

\*/

case class Takedowns(reasons: Set[TakedownReason]) {

def countryCodes: Set[CountryCode] = reasons.collect(TakedownReasons.reasonToCountryCode)

}