package com.twitter.visibility.generators

import com.twitter.visibility.common.actions.InterstitialReason

import com.twitter.visibility.common.actions.LocalizedMessage

import com.twitter.visibility.common.actions.MessageLink

import com.twitter.visibility.results.richtext.InterstitialReasonToRichText

import com.twitter.visibility.results.richtext.InterstitialReasonToRichText.InterstitialCopy

import com.twitter.visibility.results.richtext.InterstitialReasonToRichText.InterstitialLink

import com.twitter.visibility.results.translation.LearnMoreLink

import com.twitter.visibility.results.translation.Resource

import com.twitter.visibility.results.translation.Translator

object InterstitialReasonToLocalizedMessage {

def apply(

reason: InterstitialReason,

languageTag: String,

): Option[LocalizedMessage] = {

InterstitialReasonToRichText.reasonToCopy(reason).map { copy =>

val text = Translator.translate(

copy.resource,

languageTag

)

localizeWithCopyAndText(copy, languageTag, text)

}

}

private def localizeWithCopyAndText(

copy: InterstitialCopy,

languageTag: String,

text: String

): LocalizedMessage = {

val learnMore = Translator.translate(LearnMoreLink, languageTag)

val learnMoreLinkOpt =

copy.link.map { link =>

MessageLink(key = Resource.LearnMorePlaceholder, displayText = learnMore, uri = link)

}

val additionalLinks = copy.additionalLinks.map {

case InterstitialLink(placeholder, copyResource, link) =>

val copyText = Translator.translate(copyResource, languageTag)

MessageLink(key = placeholder, displayText = copyText, uri = link)

}

val links = learnMoreLinkOpt.toSeq ++ additionalLinks

LocalizedMessage(message = text, language = languageTag, links = links)

}

}