package com.twitter.home\_mixer.functional\_component.decorator.urt.builder

import com.twitter.home\_mixer.model.HomeFeatures.AuthorIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.IsRetweetFeature

import com.twitter.home\_mixer.model.HomeFeatures.ScreenNamesFeature

import com.twitter.home\_mixer.model.HomeFeatures.SuggestTypeFeature

import com.twitter.home\_mixer.product.following.model.HomeMixerExternalStrings

import com.twitter.product\_mixer.core.feature.featuremap.FeatureMap

import com.twitter.product\_mixer.core.model.marshalling.response.urt.metadata.ChildFeedbackAction

import com.twitter.product\_mixer.core.product.guice.scope.ProductScoped

import com.twitter.stringcenter.client.StringCenter

import com.twitter.timelines.service.{thriftscala => t}

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

case class RetweeterChildFeedbackActionBuilder @Inject() (

@ProductScoped stringCenter: StringCenter,

externalStrings: HomeMixerExternalStrings) {

def apply(candidateFeatures: FeatureMap): Option[ChildFeedbackAction] = {

val isRetweet = candidateFeatures.getOrElse(IsRetweetFeature, false)

if (isRetweet) {

candidateFeatures.getOrElse(AuthorIdFeature, None).flatMap { retweeterId =>

FeedbackUtil.buildUserSeeFewerChildFeedbackAction(

userId = retweeterId,

namesByUserId = candidateFeatures.getOrElse(ScreenNamesFeature, Map.empty[Long, String]),

promptExternalString = externalStrings.showFewerRetweetsString,

confirmationExternalString = externalStrings.showFewerRetweetsConfirmationString,

engagementType = t.FeedbackEngagementType.Retweet,

stringCenter = stringCenter,

injectionType = candidateFeatures.getOrElse(SuggestTypeFeature, None)

)

}

} else None

}

}