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

import com.twitter.conversions.DurationOps.\_

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

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

import com.twitter.stringcenter.client.StringCenter

import com.twitter.stringcenter.client.core.ExternalString

import com.twitter.timelines.common.{thriftscala => tlc}

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

import com.twitter.timelineservice.model.FeedbackInfo

import com.twitter.timelineservice.model.FeedbackMetadata

import com.twitter.timelineservice.suggests.{thriftscala => st}

import com.twitter.timelineservice.{thriftscala => tlst}

object FeedbackUtil {

val FeedbackTtl = 30.days

def buildUserSeeFewerChildFeedbackAction(

userId: Long,

namesByUserId: Map[Long, String],

promptExternalString: ExternalString,

confirmationExternalString: ExternalString,

engagementType: t.FeedbackEngagementType,

stringCenter: StringCenter,

injectionType: Option[st.SuggestType]

): Option[ChildFeedbackAction] = {

namesByUserId.get(userId).map { userScreenName =>

val prompt = stringCenter.prepare(

promptExternalString,

Map("user" -> userScreenName)

)

val confirmation = stringCenter.prepare(

confirmationExternalString,

Map("user" -> userScreenName)

)

val feedbackMetadata = FeedbackMetadata(

engagementType = Some(engagementType),

entityIds = Seq(tlc.FeedbackEntity.UserId(userId)),

ttl = Some(FeedbackTtl))

val feedbackUrl = FeedbackInfo.feedbackUrl(

feedbackType = tlst.FeedbackType.SeeFewer,

feedbackMetadata = feedbackMetadata,

injectionType = injectionType

)

ChildFeedbackAction(

feedbackType = SeeFewer,

prompt = Some(prompt),

confirmation = Some(confirmation),

feedbackUrl = Some(feedbackUrl),

hasUndoAction = Some(true),

confirmationDisplayType = None,

clientEventInfo = None,

icon = None,

richBehavior = None,

subprompt = None

)

}

}

}