package com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.metadata

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.icon.HorizonIconMarshaller

import com.twitter.product\_mixer.core.functional\_component.marshaller.response.urt.metadata.FeedbackActionMarshaller.generateKey

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

import com.twitter.timelines.render.{thriftscala => urt}

import javax.inject.Inject

import javax.inject.Singleton

object FeedbackActionMarshaller {

def generateKey(feedbackAction: urt.FeedbackAction): String = {

feedbackAction.hashCode.toString

}

}

@Singleton

class FeedbackActionMarshaller @Inject() (

childFeedbackActionMarshaller: ChildFeedbackActionMarshaller,

feedbackTypeMarshaller: FeedbackTypeMarshaller,

confirmationDisplayTypeMarshaller: ConfirmationDisplayTypeMarshaller,

clientEventInfoMarshaller: ClientEventInfoMarshaller,

horizonIconMarshaller: HorizonIconMarshaller,

richFeedbackBehaviorMarshaller: RichFeedbackBehaviorMarshaller) {

def apply(feedbackAction: FeedbackAction): urt.FeedbackAction = {

val childKeys = feedbackAction.childFeedbackActions

.map(\_.map { childFeedbackAction =>

val urtChildFeedbackAction = childFeedbackActionMarshaller(childFeedbackAction)

generateKey(urtChildFeedbackAction)

})

urt.FeedbackAction(

feedbackType = feedbackTypeMarshaller(feedbackAction.feedbackType),

prompt = feedbackAction.prompt,

confirmation = feedbackAction.confirmation,

childKeys = childKeys,

feedbackUrl = feedbackAction.feedbackUrl,

hasUndoAction = feedbackAction.hasUndoAction,

confirmationDisplayType =

feedbackAction.confirmationDisplayType.map(confirmationDisplayTypeMarshaller(\_)),

clientEventInfo = feedbackAction.clientEventInfo.map(clientEventInfoMarshaller(\_)),

icon = feedbackAction.icon.map(horizonIconMarshaller(\_)),

richBehavior = feedbackAction.richBehavior.map(richFeedbackBehaviorMarshaller(\_)),

subprompt = feedbackAction.subprompt,

encodedFeedbackRequest = feedbackAction.encodedFeedbackRequest

)

}

}