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

import com.twitter.product\_mixer.core.model.marshalling.response.urp.FollowedTopicsMeTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.FollowedTopicsOtherTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.ForYouExploreMixerTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.NoteworthyAccountsTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.NotInterestedTopicsMeTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.NuxForYouCategoryUserRecommendationsTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.NuxGeoCategoryUserRecommendationsTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.NuxPymkCategoryUserRecommendationsTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.NuxSingleInterestCategoryUserRecommendationsTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.NuxUserRecommendationsTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.ShoppingHomeTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.TimelineKey

import com.twitter.product\_mixer.core.model.marshalling.response.urp.TopicsLandingTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.TopicsPickerTimeline

import com.twitter.product\_mixer.core.model.marshalling.response.urp.TrendingExploreMixerTimeline

import com.twitter.strato.graphql.timelines.{thriftscala => graphql}

import javax.inject.Singleton

@Singleton

class TimelineKeyMarshaller {

def apply(timelineKey: TimelineKey): graphql.TimelineKey = timelineKey match {

case TopicsLandingTimeline(topicId) =>

graphql.TimelineKey.TopicTimeline(graphql.TopicId(topicId))

case NoteworthyAccountsTimeline(topicId) =>

graphql.TimelineKey.NoteworthyAccountsTimeline(graphql.TopicId(topicId))

case TopicsPickerTimeline(topicId) =>

graphql.TimelineKey.TopicsPickerTimeline(graphql.TopicId(topicId))

case FollowedTopicsMeTimeline() =>

graphql.TimelineKey.FollowedTopicsMeTimeline(graphql.Void())

case NotInterestedTopicsMeTimeline() =>

graphql.TimelineKey.NotInterestedTopicsMeTimeline(graphql.Void())

case FollowedTopicsOtherTimeline(userId) =>

graphql.TimelineKey.FollowedTopicsOtherTimeline(userId)

case NuxUserRecommendationsTimeline() =>

graphql.TimelineKey.NuxUserRecommendationsTimeline(graphql.Void())

case NuxForYouCategoryUserRecommendationsTimeline() =>

graphql.TimelineKey.NuxForYouCategoryUserRecommendationsTimeline(graphql.Void())

case NuxPymkCategoryUserRecommendationsTimeline() =>

graphql.TimelineKey.NuxPymkCategoryUserRecommendationsTimeline(graphql.Void())

case NuxGeoCategoryUserRecommendationsTimeline() =>

graphql.TimelineKey.NuxGeoCategoryUserRecommendationsTimeline(graphql.Void())

case NuxSingleInterestCategoryUserRecommendationsTimeline(topicId) =>

graphql.TimelineKey.NuxSingleInterestCategoryUserRecommendationsTimeline(

graphql.TopicId(topicId))

case ShoppingHomeTimeline() => graphql.TimelineKey.ShoppingHome(graphql.Void())

case ForYouExploreMixerTimeline() =>

graphql.TimelineKey.ForYouExploreMixerTimeline(graphql.Void())

case TrendingExploreMixerTimeline() =>

graphql.TimelineKey.TrendingExploreMixerTimeline(graphql.Void())

}

}