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

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

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

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

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

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

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

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

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

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

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

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

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

object UrpTransportMarshallerBuilder {

// Convenience constructor for services not using dependency injection and unit tests. If using

// dependency injection, instead @Inject an instance of UrpTransportMarshaller to construct.

val timelineKeyMarshaller = new TimelineKeyMarshaller

val timelineScribeConfigMarshaller = new TimelineScribeConfigMarshaller

val urlMarshaller = new UrlMarshaller(new UrlTypeMarshaller, new UrtEndpointOptionsMarshaller)

val clientEventInfoMarshaller = new ClientEventInfoMarshaller(

new ClientEventDetailsMarshaller(

new ConversationDetailsMarshaller(new ConversationSectionMarshaller),

new TimelinesDetailsMarshaller,

new ArticleDetailsMarshaller,

new LiveEventDetailsMarshaller,

new CommerceDetailsMarshaller)

)

val segmentedTimelineMarshaller =

new SegmentedTimelineMarshaller(timelineKeyMarshaller, timelineScribeConfigMarshaller)

val segmentedTimelinesMarshaller = new SegmentedTimelinesMarshaller(segmentedTimelineMarshaller)

val pageBodyMarshaller: PageBodyMarshaller = new PageBodyMarshaller(

timelineKeyMarshaller,

segmentedTimelinesMarshaller

)

val topicPageHeaderFacepileMarshaller = new TopicPageHeaderFacepileMarshaller(urlMarshaller)

val topicPageHeaderDisplayTypeMarshaller = new TopicPageHeaderDisplayTypeMarshaller

val topicPageHeaderMarshaller = new TopicPageHeaderMarshaller(

topicPageHeaderFacepileMarshaller,

clientEventInfoMarshaller,

topicPageHeaderDisplayTypeMarshaller

)

val pageHeaderMarshaller: PageHeaderMarshaller = new PageHeaderMarshaller(

topicPageHeaderMarshaller)

val topicPageNavBarMarshaller = new TopicPageNavBarMarshaller(clientEventInfoMarshaller)

val titleNavBarMarshaller = new TitleNavBarMarshaller(clientEventInfoMarshaller)

val pageNavBarMarshaller: PageNavBarMarshaller = new PageNavBarMarshaller(

topicPageNavBarMarshaller,

titleNavBarMarshaller

)

val marshaller: UrpTransportMarshaller =

new UrpTransportMarshaller(

pageBodyMarshaller = pageBodyMarshaller,

timelineScribeConfigMarshaller = timelineScribeConfigMarshaller,

pageHeaderMarshaller = pageHeaderMarshaller,

pageNavBarMarshaller = pageNavBarMarshaller

)

}