package com.twitter.product\_mixer.component\_library.decorator.urt.builder.flexible\_injection\_pipeline

import com.twitter.onboarding.injections.thriftscala.Injection

import com.twitter.onboarding.injections.{thriftscala => onboardingthrift}

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module.AutomaticUniqueModuleId

import com.twitter.product\_mixer.component\_library.decorator.urt.builder.timeline\_module.ModuleIdGeneration

import com.twitter.product\_mixer.component\_library.model.candidate.BasePromptCandidate

import com.twitter.product\_mixer.component\_library.pipeline.candidate.flexible\_injection\_pipeline.transformer.FlipPromptInjectionsFeature

import com.twitter.product\_mixer.core.functional\_component.decorator.urt.builder.timeline\_module.BaseTimelineModuleBuilder

import com.twitter.product\_mixer.core.functional\_component.marshaller.TransportMarshaller

import com.twitter.product\_mixer.core.model.common.CandidateWithFeatures

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

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

import com.twitter.product\_mixer.core.model.marshalling.response.urt.timeline\_module.Carousel

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

case class FlipPromptUrtModuleBuilder[-Query <: PipelineQuery](

moduleIdGeneration: ModuleIdGeneration = AutomaticUniqueModuleId())

extends BaseTimelineModuleBuilder[Query, BasePromptCandidate[Any]] {

override def apply(

query: Query,

candidates: Seq[CandidateWithFeatures[BasePromptCandidate[Any]]]

): TimelineModule = {

val firstCandidate = candidates.head

val injection = firstCandidate.features.get(FlipPromptInjectionsFeature)

injection match {

case Injection.TilesCarousel(candidate) =>

TimelineModule(

id = moduleIdGeneration.moduleId,

sortIndex = None,

entryNamespace = EntryNamespace("flip-timeline-module"),

clientEventInfo =

Some(OnboardingInjectionConversions.convertClientEventInfo(candidate.clientEventInfo)),

feedbackActionInfo =

candidate.feedbackInfo.map(OnboardingInjectionConversions.convertFeedbackInfo),

isPinned = Some(candidate.isPinnedEntry),

// Items are automatically set in the domain marshaller phase

items = Seq.empty,

displayType = Carousel,

header = candidate.header.map(TilesCarouselConversions.convertModuleHeader),

footer = None,

metadata = None,

showMoreBehavior = None

)

case \_ => throw new UnsupportedFlipPromptInModuleException(injection)

}

}

}

class UnsupportedFlipPromptInModuleException(injection: onboardingthrift.Injection)

extends UnsupportedOperationException(

"Unsupported timeline item in a Flip prompt module " + TransportMarshaller.getSimpleName(

injection.getClass))