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

import com.twitter.onboarding.task.service.thriftscala.PromptType

import com.twitter.onboarding.task.service.{thriftscala => flip}

import com.twitter.product\_mixer.core.functional\_component.transformer.CandidatePipelineQueryTransformer

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

object FlipQueryTransformer

extends CandidatePipelineQueryTransformer[

PipelineQuery with HasFlipInjectionParams,

flip.GetInjectionsRequest

] {

val SUPPORTED\_PROMPT\_TYPES: Set[PromptType] = Set(

PromptType.InlinePrompt,

PromptType.FullCover,

PromptType.HalfCover,

PromptType.TileCarousel,

PromptType.RelevancePrompt)

override def transform(

query: PipelineQuery with HasFlipInjectionParams

): flip.GetInjectionsRequest = {

val clientContext = flip.ClientContext(

userId = query.clientContext.userId,

guestId = query.clientContext.guestId,

clientApplicationId = query.clientContext.appId,

deviceId = query.clientContext.deviceId,

countryCode = query.clientContext.countryCode,

languageCode = query.clientContext.languageCode,

userAgent = query.clientContext.userAgent,

guestIdMarketing = query.clientContext.guestIdMarketing,

guestIdAds = query.clientContext.guestIdAds,

isInternalOrTwoffice = query.clientContext.isTwoffice,

ipAddress = query.clientContext.ipAddress

)

val displayContext: flip.DisplayContext =

flip.DisplayContext(

displayLocation = query.displayLocation,

timelineId = query.clientContext.userId

)

val requestTargetingContext: flip.RequestTargetingContext =

flip.RequestTargetingContext(

rankingDisablerWithLatestControlsAvaliable =

query.rankingDisablerWithLatestControlsAvailable,

reactivePromptContext = None,

isEmptyState = query.isEmptyState,

isFirstRequestAfterSignup = query.isFirstRequestAfterSignup,

isEndOfTimeline = query.isEndOfTimeline

)

flip.GetInjectionsRequest(

clientContext = clientContext,

displayContext = displayContext,

requestTargetingContext = Some(requestTargetingContext),

userRoles = query.clientContext.userRoles,

timelineContext = None,

supportedPromptTypes = Some(SUPPORTED\_PROMPT\_TYPES)

)

}

}