package com.twitter.follow\_recommendations.flows.content\_recommender\_flow

import com.twitter.core\_workflows.user\_model.thriftscala.UserState

import com.twitter.follow\_recommendations.common.models.DebugOptions

import com.twitter.follow\_recommendations.common.models.DisplayLocation

import com.twitter.follow\_recommendations.common.models.GeohashAndCountryCode

import com.twitter.follow\_recommendations.common.models.HasDebugOptions

import com.twitter.follow\_recommendations.common.models.HasDisplayLocation

import com.twitter.follow\_recommendations.common.models.HasExcludedUserIds

import com.twitter.follow\_recommendations.common.models.HasGeohashAndCountryCode

import com.twitter.follow\_recommendations.common.models.HasInvalidRelationshipUserIds

import com.twitter.follow\_recommendations.common.models.HasRecentFollowedByUserIds

import com.twitter.follow\_recommendations.common.models.HasRecentFollowedUserIds

import com.twitter.follow\_recommendations.common.models.HasUserState

import com.twitter.product\_mixer.core.model.marshalling.request.ClientContext

import com.twitter.product\_mixer.core.model.marshalling.request.HasClientContext

import com.twitter.timelines.configapi.HasParams

import com.twitter.timelines.configapi.Params

case class ContentRecommenderRequest(

override val params: Params,

override val clientContext: ClientContext,

inputExcludeUserIds: Seq[Long],

override val recentFollowedUserIds: Option[Seq[Long]],

override val recentFollowedByUserIds: Option[Seq[Long]],

override val invalidRelationshipUserIds: Option[Set[Long]],

override val displayLocation: DisplayLocation,

maxResults: Option[Int] = None,

override val debugOptions: Option[DebugOptions] = None,

override val geohashAndCountryCode: Option[GeohashAndCountryCode] = None,

override val userState: Option[UserState] = None)

extends HasParams

with HasClientContext

with HasDisplayLocation

with HasDebugOptions

with HasRecentFollowedUserIds

with HasRecentFollowedByUserIds

with HasInvalidRelationshipUserIds

with HasExcludedUserIds

with HasUserState

with HasGeohashAndCountryCode {

override val excludedUserIds: Seq[Long] = {

inputExcludeUserIds ++ clientContext.userId.toSeq

}

}