package com.twitter.follow\_recommendations.flows.post\_nux\_ml

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

import com.twitter.follow\_recommendations.common.feature\_hydration.common.HasPreFetchedFeature

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

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 PostNuxMlRequest(

override val params: Params,

override val clientContext: ClientContext,

override val similarToUserIds: Seq[Long],

inputExcludeUserIds: Seq[Long],

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

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

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

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

override val displayLocation: DisplayLocation,

maxResults: Option[Int] = None,

override val debugOptions: Option[DebugOptions] = None,

override val wtfImpressions: Option[Seq[WtfImpression]],

override val uttInterestIds: Option[Seq[Long]] = None,

override val customInterests: Option[Seq[String]] = None,

override val geohashAndCountryCode: Option[GeohashAndCountryCode] = None,

inputPreviouslyRecommendedUserIds: Option[Set[Long]] = None,

inputPreviouslyFollowedUserIds: Option[Set[Long]] = None,

override val isSoftUser: Boolean = false,

override val userState: Option[UserState] = None,

override val qualityFactor: Option[Double] = None)

extends HasParams

with HasSimilarToContext

with HasClientContext

with HasExcludedUserIds

with HasDisplayLocation

with HasDebugOptions

with HasGeohashAndCountryCode

with HasPreFetchedFeature

with HasDismissedUserIds

with HasInterestIds

with HasPreviousRecommendationsContext

with HasIsSoftUser

with HasUserState

with HasInvalidRelationshipUserIds

with HasQualityFactor {

override val excludedUserIds: Seq[Long] = {

inputExcludeUserIds ++ clientContext.userId.toSeq ++ similarToUserIds

}

override val previouslyRecommendedUserIDs: Set[Long] =

inputPreviouslyRecommendedUserIds.getOrElse(Set.empty)

override val previouslyFollowedUserIds: Set[Long] =

inputPreviouslyFollowedUserIds.getOrElse(Set.empty)

}