package com.twitter.follow\_recommendations.common.feature\_hydration.sources

import com.google.inject.Inject

import com.google.inject.Provides

import com.google.inject.Singleton

import com.twitter.follow\_recommendations.common.feature\_hydration.adapters.PreFetchedFeatureAdapter

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

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

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

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

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

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

import com.twitter.ml.api.DataRecord

import com.twitter.ml.api.FeatureContext

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

import com.twitter.stitch.Stitch

import com.twitter.timelines.configapi.HasParams

@Provides

@Singleton

class PreFetchedFeatureSource @Inject() () extends FeatureSource {

override def id: FeatureSourceId = FeatureSourceId.PreFetchedFeatureSourceId

override def featureContext: FeatureContext = PreFetchedFeatureAdapter.getFeatureContext

override def hydrateFeatures(

target: HasClientContext

with HasPreFetchedFeature

with HasParams

with HasSimilarToContext

with HasDisplayLocation,

candidates: Seq[CandidateUser]

): Stitch[Map[CandidateUser, DataRecord]] = {

Stitch.value(candidates.map { candidate =>

candidate -> PreFetchedFeatureAdapter.adaptToDataRecord((target, candidate))

}.toMap)

}

}