package com.twitter.follow\_recommendations.common.candidate\_sources.stp

import com.twitter.finagle.stats.StatsReceiver

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

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

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

import com.twitter.hermit.model.Algorithm

import com.twitter.product\_mixer.core.functional\_component.candidate\_source.CandidateSource

import com.twitter.product\_mixer.core.model.common.identifier.CandidateSourceIdentifier

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

import com.twitter.stitch.Stitch

import com.twitter.timelines.configapi.HasParams

import com.twitter.util.logging.Logging

import com.twitter.wtf.scalding.jobs.strong\_tie\_prediction.STPFeatureGenerator

import com.twitter.wtf.scalding.jobs.strong\_tie\_prediction.STPRecord

abstract class BaseOnlineSTPSource(

stpGraphBuilder: STPGraphBuilder,

baseStatsReceiver: StatsReceiver)

extends CandidateSource[

HasClientContext with HasParams with HasRecentFollowedUserIds,

CandidateUser

]

with Logging {

protected val statsReceiver: StatsReceiver = baseStatsReceiver.scope("online\_stp")

override val identifier: CandidateSourceIdentifier = BaseOnlineSTPSource.Identifier

def getCandidates(

records: Seq[STPRecord],

request: HasClientContext with HasParams with HasRecentFollowedUserIds

): Stitch[Seq[CandidateUser]]

override def apply(

request: HasClientContext with HasParams with HasRecentFollowedUserIds

): Stitch[Seq[CandidateUser]] =

request.getOptionalUserId

.map { userId =>

stpGraphBuilder(request)

.flatMap { graph: STPGraph =>

logger.debug(graph)

val records = STPFeatureGenerator.constructFeatures(

userId,

graph.firstDegreeEdgeInfoList,

graph.secondDegreeEdgeInfoList)

getCandidates(records, request)

}

}.getOrElse(Stitch.Nil)

}

object BaseOnlineSTPSource {

val Identifier: CandidateSourceIdentifier = CandidateSourceIdentifier(

Algorithm.OnlineStrongTiePredictionRecNoCaching.toString)

}