package com.twitter.product\_mixer.component\_library.candidate\_source.cr\_mixer

import com.twitter.cr\_mixer.{thriftscala => t}

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

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

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

/\*\*

\* Returns out-of-network Tweet recommendations by using user recommendations

\* from FollowRecommendationService as an input seed-set to Earlybird

\*/

@Singleton

class CrMixerFrsBasedTweetRecommendationsCandidateSource @Inject() (

crMixerClient: t.CrMixer.MethodPerEndpoint)

extends CandidateSource[t.FrsTweetRequest, t.FrsTweet] {

override val identifier: CandidateSourceIdentifier =

CandidateSourceIdentifier("CrMixerFrsBasedTweetRecommendations")

override def apply(request: t.FrsTweetRequest): Stitch[Seq[t.FrsTweet]] = Stitch

.callFuture(crMixerClient.getFrsBasedTweetRecommendations(request))

.map(\_.tweets)

}