package com.twitter.follow\_recommendations.common.base

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.follow\_recommendations.common.base.EnrichedCandidateSource.toEnriched

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

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

// a helper structure to register and select candidate sources based on identifiers

trait CandidateSourceRegistry[Target, Candidate] {

val statsReceiver: StatsReceiver

def sources: Set[CandidateSource[Target, Candidate]]

final lazy val candidateSources: Map[

CandidateSourceIdentifier,

CandidateSource[Target, Candidate]

] = {

val map = sources.map { c =>

c.identifier -> c.observe(statsReceiver)

}.toMap

if (map.size != sources.size) {

throw new IllegalArgumentException("Duplicate Candidate Source Identifiers")

}

map

}

def select(

identifiers: Set[CandidateSourceIdentifier]

): Set[CandidateSource[Target, Candidate]] = {

// fails loud if the candidate source is not registered

identifiers.map(candidateSources(\_))

}

}