package com.twitter.follow\_recommendations.common.candidate\_sources.top\_organic\_follows\_accounts

import com.twitter.timelines.configapi.FSBoundedParam

import com.twitter.timelines.configapi.FSEnumSeqParam

import com.twitter.timelines.configapi.FSParam

object TopOrganicFollowsAccountsParams {

// whether or not to fetch TopOrganicFollowsAccounts candidate sources

case object CandidateSourceEnabled

extends FSParam[Boolean]("top\_organic\_follows\_accounts\_candidate\_source\_enabled", false)

/\*\*

\* Contains the logic key for account filtering and ranking. Currently we have 3 main logic keys

\* - new\_organic\_follows: filtering top organically followed accounts followed by new users

\* - non\_new\_organic\_follows: filtering top organically followed accounts followed by non new users

\* - organic\_follows: filtering top organically followed accounts followed by all users

\* Mapping of the Logic Id to Logic key is done via @enum AccountsFilteringAndRankingLogic

\*/

case object AccountsFilteringAndRankingLogics

extends FSEnumSeqParam[AccountsFilteringAndRankingLogicId.type](

name = "top\_organic\_follows\_accounts\_filtering\_and\_ranking\_logic\_ids",

default = Seq(AccountsFilteringAndRankingLogicId.OrganicFollows),

enum = AccountsFilteringAndRankingLogicId)

case object CandidateSourceWeight

extends FSBoundedParam[Double](

"top\_organic\_follows\_accounts\_candidate\_source\_weight",

default = 1200,

min = 0.001,

max = 2000)

}