package com.twitter.cr\_mixer.filter

import com.twitter.cr\_mixer.model.CandidateGeneratorQuery

import com.twitter.cr\_mixer.model.InitialCandidate

import com.twitter.util.Future

import javax.inject.Inject

import javax.inject.Singleton

/\*\*\*

\* Filters candidates that are replies

\*/

@Singleton

case class ReplyFilter @Inject() () extends FilterBase {

override def name: String = this.getClass.getCanonicalName

override type ConfigType = Boolean

override def filter(

candidates: Seq[Seq[InitialCandidate]],

config: ConfigType

): Future[Seq[Seq[InitialCandidate]]] = {

if (config) {

Future.value(

candidates.map { candidateSeq =>

candidateSeq.filterNot { candidate =>

candidate.tweetInfo.isReply.getOrElse(false)

}

}

)

} else {

Future.value(candidates)

}

}

override def requestToConfig[CGQueryType <: CandidateGeneratorQuery](

query: CGQueryType

): ConfigType = {

true

}

}