package com.twitter.cr\_mixer.blender

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

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

import com.twitter.cr\_mixer.util.InterleaveUtil

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.util.Future

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

case class InterleaveBlender @Inject() (globalStats: StatsReceiver) {

private val name: String = this.getClass.getCanonicalName

private val stats: StatsReceiver = globalStats.scope(name)

/\*\*

\* Interleaves candidates, by taking 1 candidate from each Seq[Seq[InitialCandidate]] in sequence,

\* until we run out of candidates.

\*/

def blend(

inputCandidates: Seq[Seq[InitialCandidate]],

): Future[Seq[BlendedCandidate]] = {

val interleavedCandidates = InterleaveUtil.interleave(inputCandidates)

stats.stat("candidates").add(interleavedCandidates.size)

val blendedCandidates = BlendedCandidatesBuilder.build(inputCandidates, interleavedCandidates)

Future.value(blendedCandidates)

}

}