package com.twitter.home\_mixer.product.for\_you.side\_effect

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.home\_mixer.model.HomeFeatures.InNetworkFeature

import com.twitter.home\_mixer.model.HomeFeatures.InReplyToTweetIdFeature

import com.twitter.home\_mixer.model.HomeFeatures.SuggestTypeFeature

import com.twitter.home\_mixer.product.for\_you.param.ForYouParam.ExperimentStatsParam

import com.twitter.home\_mixer.util.CandidatesUtil

import com.twitter.product\_mixer.component\_library.model.candidate.TweetCandidate

import com.twitter.product\_mixer.core.functional\_component.side\_effect.PipelineResultSideEffect

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

import com.twitter.product\_mixer.core.model.common.presentation.CandidateWithDetails

import com.twitter.product\_mixer.core.model.common.presentation.ItemCandidateWithDetails

import com.twitter.product\_mixer.core.model.marshalling.response.urt.Timeline

import com.twitter.product\_mixer.core.pipeline.PipelineQuery

import com.twitter.stitch.Stitch

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

class ServedStatsSideEffect @Inject() (statsReceiver: StatsReceiver)

extends PipelineResultSideEffect[PipelineQuery, Timeline] {

override val identifier: SideEffectIdentifier = SideEffectIdentifier("ServedStats")

private val baseStatsReceiver = statsReceiver.scope(identifier.toString)

private val suggestTypeStatsReceiver = baseStatsReceiver.scope("SuggestType")

private val responseSizeStatsReceiver = baseStatsReceiver.scope("ResponseSize")

private val contentBalanceStatsReceiver = baseStatsReceiver.scope("ContentBalance")

private val inNetworkStatsReceiver = contentBalanceStatsReceiver.scope("InNetwork")

private val outOfNetworkStatsReceiver = contentBalanceStatsReceiver.scope("OutOfNetwork")

private val replyStatsReceiver = contentBalanceStatsReceiver.scope("Reply")

private val originalStatsReceiver = contentBalanceStatsReceiver.scope("Original")

private val emptyStatsReceiver = responseSizeStatsReceiver.scope("Empty")

private val lessThan5StatsReceiver = responseSizeStatsReceiver.scope("LessThan5")

private val lessThan10StatsReceiver = responseSizeStatsReceiver.scope("LessThan10")

override def apply(

inputs: PipelineResultSideEffect.Inputs[PipelineQuery, Timeline]

): Stitch[Unit] = {

val tweetCandidates = CandidatesUtil

.getItemCandidates(inputs.selectedCandidates).filter(\_.isCandidateType[TweetCandidate]())

val expBucket = inputs.query.params(ExperimentStatsParam)

recordSuggestTypeStats(tweetCandidates, expBucket)

recordContentBalanceStats(tweetCandidates, expBucket)

recordResponseSizeStats(tweetCandidates, expBucket)

Stitch.Unit

}

def recordSuggestTypeStats(

candidates: Seq[ItemCandidateWithDetails],

expBucket: String

): Unit = {

candidates.groupBy(getSuggestType).foreach {

case (suggestType, suggestTypeCandidates) =>

suggestTypeStatsReceiver

.scope(expBucket).counter(suggestType).incr(suggestTypeCandidates.size)

}

}

def recordContentBalanceStats(

candidates: Seq[ItemCandidateWithDetails],

expBucket: String

): Unit = {

val (in, oon) = candidates.partition(\_.features.getOrElse(InNetworkFeature, true))

inNetworkStatsReceiver.counter(expBucket).incr(in.size)

outOfNetworkStatsReceiver.counter(expBucket).incr(oon.size)

val (reply, original) =

candidates.partition(\_.features.getOrElse(InReplyToTweetIdFeature, None).isDefined)

replyStatsReceiver.counter(expBucket).incr(reply.size)

originalStatsReceiver.counter(expBucket).incr(original.size)

}

def recordResponseSizeStats(

candidates: Seq[ItemCandidateWithDetails],

expBucket: String

): Unit = {

if (candidates.size == 0) emptyStatsReceiver.counter(expBucket).incr()

if (candidates.size < 5) lessThan5StatsReceiver.counter(expBucket).incr()

if (candidates.size < 10) lessThan10StatsReceiver.counter(expBucket).incr()

}

private def getSuggestType(candidate: CandidateWithDetails): String =

candidate.features.getOrElse(SuggestTypeFeature, None).map(\_.name).getOrElse("None")

}