package com.twitter.home\_mixer.module

import com.google.inject.Provides

import com.twitter.conversions.DurationOps.\_

import com.twitter.eventbus.client.EventBusPublisher

import com.twitter.eventbus.client.EventBusPublisherBuilder

import com.twitter.finagle.mtls.authentication.ServiceIdentifier

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import com.twitter.timelines.config.ConfigUtils

import com.twitter.timelines.config.Env

import com.twitter.timelines.impressionstore.thriftscala.PublishedImpressionList

import javax.inject.Singleton

object ClientSentImpressionsPublisherModule extends TwitterModule with ConfigUtils {

private val serviceName = "home-mixer"

@Singleton

@Provides

def providesClientSentImpressionsPublisher(

serviceIdentifier: ServiceIdentifier,

statsReceiver: StatsReceiver

): EventBusPublisher[PublishedImpressionList] = {

val env = serviceIdentifier.environment.toLowerCase match {

case "prod" => Env.prod

case "staging" => Env.staging

case "local" => Env.local

case \_ => Env.devel

}

val streamName = env match {

case Env.prod => "timelinemixer\_client\_sent\_impressions\_prod"

case \_ => "timelinemixer\_client\_sent\_impressions\_devel"

}

EventBusPublisherBuilder()

.clientId(clientIdWithScopeOpt(serviceName, env))

.serviceIdentifier(serviceIdentifier)

.streamName(streamName)

.statsReceiver(statsReceiver.scope("eventbus"))

.thriftStruct(PublishedImpressionList)

.tcpConnectTimeout(20.milliseconds)

.connectTimeout(100.milliseconds)

.requestTimeout(1.second)

.publishTimeout(1.second)

.build()

}

}