package com.twitter.cr\_mixer.featureswitch

import com.twitter.finagle.Filter

import javax.inject.Inject

import javax.inject.Singleton

import scala.collection.concurrent.TrieMap

import com.twitter.abdecider.Bucket

import com.twitter.finagle.Service

@Singleton

class SetImpressedBucketsLocalContextFilter @Inject() () extends Filter.TypeAgnostic {

override def toFilter[Req, Rep]: Filter[Req, Rep, Req, Rep] =

(request: Req, service: Service[Req, Rep]) => {

val concurrentTrieMap = TrieMap

.empty[Bucket, Boolean] // Trie map has no locks and O(1) inserts

CrMixerImpressedBuckets.localImpressedBucketsMap.let(concurrentTrieMap) {

service(request)

}

}

}