package com.twitter.home\_mixer.functional\_component.gate

import com.twitter.home\_mixer.model.request.DeviceContext.RequestContext

import com.twitter.home\_mixer.model.request.HasDeviceContext

import com.twitter.product\_mixer.core.functional\_component.gate.Gate

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

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

import com.twitter.stitch.Stitch

/\*\*

\* Gate that fetches the request context from the device context and

\* continues if the request context matches \*any\* of the specified ones.

\*/

case class RequestContextGate(requestContexts: Seq[RequestContext.Value])

extends Gate[PipelineQuery with HasDeviceContext] {

override val identifier: GateIdentifier = GateIdentifier("RequestContext")

override def shouldContinue(query: PipelineQuery with HasDeviceContext): Stitch[Boolean] =

Stitch.value(

requestContexts.exists(query.deviceContext.flatMap(\_.requestContextValue).contains))

}