package com.twitter.unified\_user\_actions.service.module

import com.twitter.decider.Decider

import com.twitter.decider.RandomRecipient

import com.twitter.unified\_user\_actions.thriftscala.ActionType

import com.twitter.unified\_user\_actions.thriftscala.UnifiedUserAction

sealed trait DeciderUtils {

def shouldPublish(decider: Decider, uua: UnifiedUserAction, sinkTopic: String): Boolean

}

object DefaultDeciderUtils extends DeciderUtils {

override def shouldPublish(decider: Decider, uua: UnifiedUserAction, sinkTopic: String): Boolean =

decider.isAvailable(feature = s"Publish${uua.actionType}", Some(RandomRecipient))

}

object ClientEventDeciderUtils extends DeciderUtils {

override def shouldPublish(decider: Decider, uua: UnifiedUserAction, sinkTopic: String): Boolean =

decider.isAvailable(

feature = s"Publish${uua.actionType}",

Some(RandomRecipient)) && (uua.actionType match {

// for heavy impressions UUA only publishes to the "all" topic, not the engagementsOnly topic.

case ActionType.ClientTweetLingerImpression | ActionType.ClientTweetRenderImpression =>

sinkTopic == TopicsMapping().all

case \_ => true

})

}