package com.twitter.representationscorer.common

import com.twitter.decider.Decider

import com.twitter.decider.RandomRecipient

import com.twitter.decider.Recipient

import com.twitter.simclusters\_v2.common.DeciderGateBuilderWithIdHashing

import javax.inject.Inject

import javax.inject.Singleton

@Singleton

case class RepresentationScorerDecider @Inject() (decider: Decider) {

val deciderGateBuilder = new DeciderGateBuilderWithIdHashing(decider)

def isAvailable(feature: String, recipient: Option[Recipient]): Boolean = {

decider.isAvailable(feature, recipient)

}

/\*\*

\* When useRandomRecipient is set to false, the decider is either completely on or off.

\* When useRandomRecipient is set to true, the decider is on for the specified % of traffic.

\*/

def isAvailable(feature: String, useRandomRecipient: Boolean = true): Boolean = {

if (useRandomRecipient) isAvailable(feature, Some(RandomRecipient))

else isAvailable(feature, None)

}

}