package com.twitter.cr\_mixer.param.decider

import com.twitter.decider.Decider

import com.twitter.decider.RandomRecipient

import com.twitter.decider.Recipient

import com.twitter.decider.SimpleRecipient

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

import javax.inject.Inject

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

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

decider.isAvailable(feature, recipient)

}

lazy val deciderGateBuilder = new DeciderGateBuilderWithIdHashing(decider)

/\*\*

\* 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)

}

/\*\*\*

\* Decide whether the decider is available for a specific id using SimpleRecipient(id).

\*/

def isAvailableForId(

id: Long,

deciderConstants: String

): Boolean = {

// Note: SimpleRecipient does expose a `val isUser = true` field which is not correct if the Id is not a user Id.

// However this field does not appear to be used anywhere in source.

decider.isAvailable(deciderConstants, Some(SimpleRecipient(id)))

}

}