package com.twitter.follow\_recommendations.models

import com.twitter.follow\_recommendations.common.models.DebugOptions

import com.twitter.follow\_recommendations.common.models.DebugOptions.fromDebugParamsThrift

import com.twitter.follow\_recommendations.logging.{thriftscala => offline}

import com.twitter.follow\_recommendations.{thriftscala => t}

import com.twitter.timelines.configapi.{FeatureValue => ConfigApiFeatureValue}

case class DebugParams(

featureOverrides: Option[Map[String, ConfigApiFeatureValue]],

debugOptions: Option[DebugOptions])

object DebugParams {

def fromThrift(thrift: t.DebugParams): DebugParams = DebugParams(

featureOverrides = thrift.featureOverrides.map { map =>

map.mapValues(FeatureValue.fromThrift).toMap

},

debugOptions = Some(

fromDebugParamsThrift(thrift)

)

)

def toOfflineThrift(model: DebugParams): offline.OfflineDebugParams =

offline.OfflineDebugParams(randomizationSeed = model.debugOptions.flatMap(\_.randomizationSeed))

}

trait HasFrsDebugParams {

def frsDebugParams: Option[DebugParams]

}