package com.twitter.cr\_mixer.module.similarity\_engine

import com.google.inject.Provides

import com.twitter.cr\_mixer.config.TimeoutConfig

import com.twitter.cr\_mixer.param.decider.CrMixerDecider

import com.twitter.cr\_mixer.param.decider.DeciderConstants

import com.twitter.cr\_mixer.similarity\_engine.EarlybirdModelBasedSimilarityEngine

import com.twitter.cr\_mixer.similarity\_engine.EarlybirdRecencyBasedSimilarityEngine

import com.twitter.cr\_mixer.similarity\_engine.EarlybirdSimilarityEngine

import com.twitter.cr\_mixer.similarity\_engine.EarlybirdTensorflowBasedSimilarityEngine

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine.DeciderConfig

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine.GatingConfig

import com.twitter.cr\_mixer.similarity\_engine.SimilarityEngine.SimilarityEngineConfig

import com.twitter.cr\_mixer.thriftscala.SimilarityEngineType

import com.twitter.finagle.stats.StatsReceiver

import com.twitter.inject.TwitterModule

import javax.inject.Singleton

object EarlybirdSimilarityEngineModule extends TwitterModule {

@Provides

@Singleton

def providesRecencyBasedEarlybirdSimilarityEngine(

earlybirdRecencyBasedSimilarityEngine: EarlybirdRecencyBasedSimilarityEngine,

timeoutConfig: TimeoutConfig,

decider: CrMixerDecider,

statsReceiver: StatsReceiver

): EarlybirdSimilarityEngine[

EarlybirdRecencyBasedSimilarityEngine.EarlybirdRecencyBasedSearchQuery,

EarlybirdRecencyBasedSimilarityEngine

] = {

new EarlybirdSimilarityEngine[

EarlybirdRecencyBasedSimilarityEngine.EarlybirdRecencyBasedSearchQuery,

EarlybirdRecencyBasedSimilarityEngine

](

implementingStore = earlybirdRecencyBasedSimilarityEngine,

identifier = SimilarityEngineType.EarlybirdRecencyBasedSimilarityEngine,

globalStats =

statsReceiver.scope(SimilarityEngineType.EarlybirdRecencyBasedSimilarityEngine.name),

engineConfig = SimilarityEngineConfig(

timeout = timeoutConfig.earlybirdSimilarityEngineTimeout,

gatingConfig = GatingConfig(

deciderConfig = Some(

DeciderConfig(

decider = decider,

deciderString = DeciderConstants.enableEarlybirdTrafficDeciderKey

)),

enableFeatureSwitch = None

)

)

)

}

@Provides

@Singleton

def providesModelBasedEarlybirdSimilarityEngine(

earlybirdModelBasedSimilarityEngine: EarlybirdModelBasedSimilarityEngine,

timeoutConfig: TimeoutConfig,

decider: CrMixerDecider,

statsReceiver: StatsReceiver

): EarlybirdSimilarityEngine[

EarlybirdModelBasedSimilarityEngine.EarlybirdModelBasedSearchQuery,

EarlybirdModelBasedSimilarityEngine

] = {

new EarlybirdSimilarityEngine[

EarlybirdModelBasedSimilarityEngine.EarlybirdModelBasedSearchQuery,

EarlybirdModelBasedSimilarityEngine

](

implementingStore = earlybirdModelBasedSimilarityEngine,

identifier = SimilarityEngineType.EarlybirdModelBasedSimilarityEngine,

globalStats =

statsReceiver.scope(SimilarityEngineType.EarlybirdModelBasedSimilarityEngine.name),

engineConfig = SimilarityEngineConfig(

timeout = timeoutConfig.earlybirdSimilarityEngineTimeout,

gatingConfig = GatingConfig(

deciderConfig = Some(

DeciderConfig(

decider = decider,

deciderString = DeciderConstants.enableEarlybirdTrafficDeciderKey

)),

enableFeatureSwitch = None

)

)

)

}

@Provides

@Singleton

def providesTensorflowBasedEarlybirdSimilarityEngine(

earlybirdTensorflowBasedSimilarityEngine: EarlybirdTensorflowBasedSimilarityEngine,

timeoutConfig: TimeoutConfig,

decider: CrMixerDecider,

statsReceiver: StatsReceiver

): EarlybirdSimilarityEngine[

EarlybirdTensorflowBasedSimilarityEngine.EarlybirdTensorflowBasedSearchQuery,

EarlybirdTensorflowBasedSimilarityEngine

] = {

new EarlybirdSimilarityEngine[

EarlybirdTensorflowBasedSimilarityEngine.EarlybirdTensorflowBasedSearchQuery,

EarlybirdTensorflowBasedSimilarityEngine

](

implementingStore = earlybirdTensorflowBasedSimilarityEngine,

identifier = SimilarityEngineType.EarlybirdTensorflowBasedSimilarityEngine,

globalStats =

statsReceiver.scope(SimilarityEngineType.EarlybirdTensorflowBasedSimilarityEngine.name),

engineConfig = SimilarityEngineConfig(

timeout = timeoutConfig.earlybirdSimilarityEngineTimeout,

gatingConfig = GatingConfig(

deciderConfig = Some(

DeciderConfig(

decider = decider,

deciderString = DeciderConstants.enableEarlybirdTrafficDeciderKey

)),

enableFeatureSwitch = None

)

)

)

}

}