package com.twitter.ann.faiss

import com.twitter.ann.common.thriftscala.FaissRuntimeParam

import com.twitter.bijection.Injection

import scala.util.Failure

import scala.util.Success

import scala.util.Try

import com.twitter.ann.common.thriftscala.{RuntimeParams => ServiceRuntimeParams}

import com.twitter.search.common.file.AbstractFile

object FaissCommon {

val RuntimeParamsInjection: Injection[FaissParams, ServiceRuntimeParams] =

new Injection[FaissParams, ServiceRuntimeParams] {

override def apply(scalaParams: FaissParams): ServiceRuntimeParams = {

ServiceRuntimeParams.FaissParam(

FaissRuntimeParam(

scalaParams.nprobe,

scalaParams.quantizerEf,

scalaParams.quantizerKFactorRF,

scalaParams.quantizerNprobe,

scalaParams.ht)

)

}

override def invert(thriftParams: ServiceRuntimeParams): Try[FaissParams] =

thriftParams match {

case ServiceRuntimeParams.FaissParam(faissParam) =>

Success(

FaissParams(

faissParam.nprobe,

faissParam.quantizerEf,

faissParam.quantizerKfactorRf,

faissParam.quantizerNprobe,

faissParam.ht))

case p => Failure(new IllegalArgumentException(s"Expected FaissParams got $p"))

}

}

def isValidFaissIndex(path: AbstractFile): Boolean = {

path.isDirectory &&

path.hasSuccessFile &&

path.getChild("faiss.index").exists()

}

}