package com.twitter.ann.annoy

import com.twitter.ann.annoy.AnnoyCommon.\_

import com.twitter.ann.common.\_

import com.twitter.ann.file\_store.ReadableIndexIdFileStore

import com.twitter.bijection.Injection

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

import com.twitter.util.FuturePool

private[annoy] object TypedAnnoyQueryIndexWithFile {

private[annoy] def apply[T, D <: Distance[D]](

dimension: Int,

metric: Metric[D],

injection: Injection[T, Array[Byte]],

futurePool: FuturePool,

directory: AbstractFile

): Queryable[T, AnnoyRuntimeParams, D] = {

val deserializer =

new TypedAnnoyQueryIndexWithFile(dimension, metric, futurePool, injection)

deserializer.fromDirectory(directory)

}

}

private[this] class TypedAnnoyQueryIndexWithFile[T, D <: Distance[D]](

dimension: Int,

metric: Metric[D],

futurePool: FuturePool,

injection: Injection[T, Array[Byte]])

extends QueryableDeserialization[

T,

AnnoyRuntimeParams,

D,

Queryable[T, AnnoyRuntimeParams, D]

] {

override def fromDirectory(directory: AbstractFile): Queryable[T, AnnoyRuntimeParams, D] = {

val index = RawAnnoyQueryIndex(dimension, metric, futurePool, directory)

val indexIdFile = directory.getChild(IndexIdMappingFileName)

val readableFileStore = ReadableIndexIdFileStore(indexIdFile, injection)

IndexTransformer.transformQueryable(index, readableFileStore)

}

}