**查询**的**三种方法**：

三种分别是：

1. fielder
2. filter
3. and（query.criteria(“…”).算术方法（））

可以连续使用如下面的例子

例子：

Query<CouponInstance> query = datastore.find(CouponInstance.class);

//新建查询器query，还有一个方法等同于上面的语句：

// Query<CouponInstance> query = datastore.createQuery(CouponInstance.class);

// query.filter("batch\_no",BathNo).and(query.criteria("bindTime").notEqual(null));

//等价于下面的语句，BathNo是传递的参数

query.and(query.criteria("bindTime").notEqual(null))

.and(query.criteria("batch\_no").equal(BathNo));

//等价于

query.field("batch\_no ").equal(BathNo)). and(query.criteria("bindTime").notEqual(null));

其中criteria以及field后面可以跟查询的后续算术方法例如contains(),equals(),notEquals()..lessThanOrEq（）

**转化成list** ：

query.asList().size();//这个就可以用来数有几个查询结果

**转化为**集合中的元素**对象：**

query.get();//获取集合中的第一个元素

**对查询结果集合中的数据进行筛选条数**

List<CouponInstance> couponInstance = query.asList(**new** FindOptions().skip(beforeCount).limit(pageSize));

**修改update写法：**

Query<Entry> query =db.createQuery(Entry.class).field("id").equal(-1991880012);//定义查询

UpdateOperations<Entry> updateOptions=db.createUpdateOperations(Entry.class).set("name","hejingyuan"); //可在代码末尾继续写set设置 其他的字段。

db.update(query, updateOptions);//update方法完成更新 参数是查询对象加上更新对象

package com.github.carter659.spring03.controller;

import com.github.carter659.spring03.pojo.Entry;

import com.github.carter659.spring03.repository.DBHelper;

import org.mongodb.morphia.Datastore;

import org.mongodb.morphia.query.Query;

import org.mongodb.morphia.query.UpdateOperations;

import java.util.List;

/\*\*

\* Unit test for simple App.

\*/

public class MongoDBTest

{

private static DBHelper dbHelper=new DBHelper();//连接类对象

private static Entry en=new Entry();//实体

private static Datastore db=dbHelper.ds;//连接类返回的连接 datastore对象

public static void queryAll(){

Query<Entry> query=db.createQuery(Entry.class);//定义查询对象

List<Entry> list=query.asList();//查询返回集合 asList()

for(Entry entry:list){//遍历查询结果集list

System.out.println(entry);

System.out.println(entry.getId());

}

}

public static void add(){//增加

int num=(int)System.currentTimeMillis();

for(int i=0;i<10;i++){

Entry en=new Entry();

en.setId(num++);

en.setName("hjy");

db.save(en);//用datastore对象的save方法

}

}

public static void update(){

//将name==hjy的修改为name=11111

// Query<Entry> query =

// db.createQuery(Entry.class).field("name").equal("hjy");

// UpdateOperations<Entry> uo =

// db.createUpdateOperations(Entry.class).set("name",

// "11111");

// db.update(query, uo);

// System.out.println("OK");

//将id为-1991880012的文档修改为name=hejingyuan

Query<Entry> query =

db.createQuery(Entry.class).field("id").equal(-1991880012);//定义查询field是id value是-1991880012

UpdateOperations<Entry> uo =

db.createUpdateOperations(Entry.class).set("name", "hejingyuan");//定义更新对将name设置成“hejingyuan”

db.update(query, uo);//update方法完成更新 参数是查询对象加上更新对象

System.out.println("OK");

}

public static void main(String args[]){

add();

queryAll();

update();

}

}

package com.github.carter659.spring03.repository;

import com.mongodb.Mongo;

import com.mongodb.MongoClient;

import org.mongodb.morphia.Datastore;

import org.mongodb.morphia.Morphia;

/\*\*

\* Created by hejy on 2016/3/18.

\*/

public class DBHelper {

public Datastore ds;

public DBHelper(){

//初始化数据

init();

}

public void init(){

try{

Mongo mongo = new Mongo("localhost");

MongoClient mongoClient=new MongoClient("localhost");

ds = new Morphia().createDatastore(mongoClient, "mydb");

}

catch(Exception e){

e.printStackTrace();

}

}

}

还要一个实体类