JAVA 之 mongodb 增删改查

1. 准备
2. 了解mongodb及其常用语言

可以根据这个网站学习和了解下mongodb的日常操作语句

http://www.w3cschool.cc/mongodb/mongodb-tutorial.html

（2）下载mongo-java-driver JAR包

<http://central.maven.org/maven2/org/mongodb/mongo-java-driver/>

[3.0.0-rc0/](http://central.maven.org/maven2/org/mongodb/mongo-java-driver/3.0.0-rc0/)

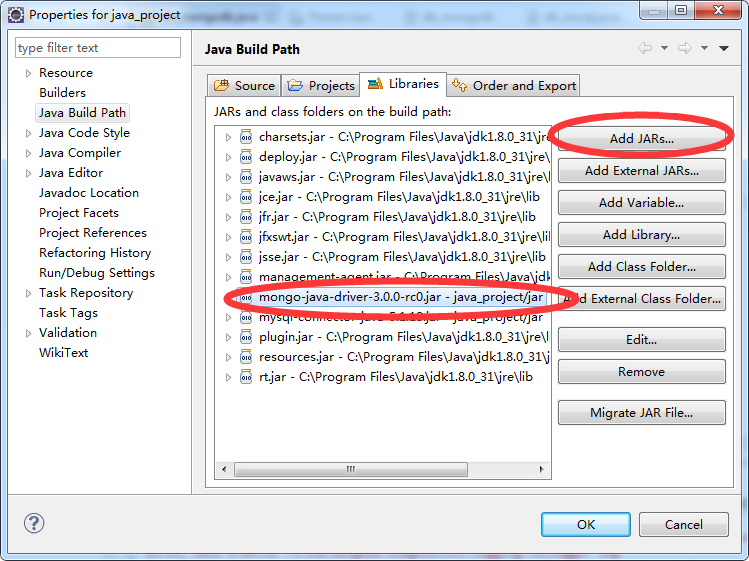
1. 添加jar包至工程目录下

（1）首先在工程文件夹下新建一个名字为jar（名称随意）的文件夹，然后复制刚才第一步下载的jar包至该文件夹下。

（2）添加对这个jar包的引用

右击工程—选择Properies

然后如图添加对jar的 引用



1. db\_mongodb类源代码，代码有注释和示例。

**import** java.util.Arrays;

**import** com.mongodb.BasicDBObject;

**import** com.mongodb.DB;

**import** com.mongodb.DBCollection;

**import** com.mongodb.DBCursor;

**import** com.mongodb.DBObject;

**import** com.mongodb.MongoClient;

**import** com.mongodb.MongoCredential;

**import** com.mongodb.ServerAddress;

**import** com.mongodb.WriteConcern;

**public** **class** db\_mongodb {

MongoClient mongoClient = **null**;

DB db = **null**;

/\*\*\*

\* 初始化数据库连接

\*

\* **@host** mongodb IP

\* **@port** int 端口号

\* **@user** mongodb用户名,如果没有赋值给null

\* **@password** mongodb密码,如果没有复制null

\* **@database连接的数据库**

\*

\*

\*\*\*/

**public** db\_mongodb(String host, **int** port, String user, String password,

String database) {

// **TODO** Auto-generated constructor stub

**if** (host == **null** || port == 0 || database == **null**) {

System.*out*

.println("please check user or port or database is inserted");

}

**if** (user == **null** && password == **null**) {

mongoClient = **new** MongoClient(host, port);

} **else** {

ServerAddress serverAddress = **new** ServerAddress(host, port);

**char**[] passwordArray = password.toCharArray();

MongoCredential credential = MongoCredential

.*createMongoCRCredential*(user, database, passwordArray);

mongoClient = **new** MongoClient(serverAddress,

Arrays.*asList*(credential));

}

mongoClient.setWriteConcern(WriteConcern.*JOURNALED*);

db = mongoClient.~~getDB~~(database);

}

/\*\*\*

\* 插入操作

\*

\* **@colletion** 待操作表名

\* **@doc待插入数据**

\*

\*\*\*/

**public** **void** db\_insert(String collection, BasicDBObject doc) {

DBCollection coll = db.getCollection(collection);

coll.insert(doc);

}

/\*\*\*

\* 查询所有操作

\*

\* **@colletion** 待操作表名

\* **@doc待查询数据** 如果为null查询第一条

\*\*\*/

**public** DBObject db\_findOne(String collection, BasicDBObject doc) {

DBCollection coll = db.getCollection(collection);

**if** (doc == **null**) {

**return** coll.findOne();

} **else** {

**return** coll.findOne(doc);

}

}

/\*\*\*

\*

\* 查询所有操作

\*

\* **@colletion** 待操作表名

\* **@doc待查询数据** 如果为null查询所有

\*

\*\*\*/

**public** DBCursor db\_findAll(String collection, BasicDBObject doc) {

DBCollection coll = db.getCollection(collection);

**if** (doc == **null**) {

**return** coll.find();

} **else** {

**return** coll.find(doc);

}

}

/\*\*\*\*\*

\* 更新操作

\*

\* **@colletion** 待操作表名

\* **@querydoc** 更新条件

\* **@updatedoc** 更新数据

\* **@multi** true:更新所有，false:更新第一条

\*

\*

\*\*\*\*/

**public** **void** db\_update(String collection, BasicDBObject querydoc,

BasicDBObject updatedoc, **boolean** multi) {

DBCollection coll = db.getCollection(collection);

**if** (multi == **false**) {

coll.update(querydoc, updatedoc);

} **else** {

coll.updateMulti(querydoc, updatedoc);

}

}

/\*\*\*\*\*

\* 删除操作

\*

\* **@colletion** 待操作表名

\* **@querydoc** 待删除数据

\*

\*\*\*\*/

**public** **void** db\_remove(String collection, BasicDBObject querydoc) {

DBCollection coll = db.getCollection(collection);

coll.remove(querydoc);

}

/\*\*\*\*\*

\* 更改目前的连接数据库为@newdatabase

\*

\*\*\*\*/

**public** **void** change\_database(String newdatabase) {

db = mongoClient.~~getDB~~(newdatabase);

}

/\*\*\*\*\*

\* 删除操作

\*

\* **@colletion** 待操作表名

\* **@doc** 待统计的doc

\*

\*\*\*\*/

**public** **long** db\_count(String collection, BasicDBObject doc) {

DBCollection coll = db.getCollection(collection);

**if** (doc == **null**) {

**return** coll.getCount();

}

**else** {

**return** coll.getCount(doc);

}

}

**public** DBObject db\_group\_by(String collection, BasicDBObject key,

BasicDBObject cond) {

BasicDBObject initial = **new** BasicDBObject("count", 0);

String reduce = "function(obj, prev){prev.count++;}";

DBCollection coll = db.getCollection(collection);

DBObject resultDbObject = coll.group(key, cond, initial, reduce);

**return** resultDbObject;

}

**public** Iterable<String> get\_collectionnames() {

Iterable<String> abcString = db.getCollectionNames();

**return** abcString;

}

**public** **void** show(DBCursor cursor) {

**try** {

**while** (cursor.hasNext()) {

DBObject temp = cursor.next();

System.*out*.println(temp);

}

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

} **finally** {

cursor.close();

}

}

**public** **static** **void** main(String[] args) {

String host = "10.0.0.191";

**int** port = 27017;

String database = "crawl\_data";

db\_mongodb oDb\_mongodb = **new** db\_mongodb(host, port, **null**, **null**,

database);

BasicDBObject doc = **new** BasicDBObject("name", "MogoDB")

.append("type", "database").append("count", 1)

.append("info", **new** BasicDBObject("x", 203).append("y", 102));

// test insert

// oDb\_mongodb.db\_insert(collection, doc);

// DBObject dbObject = oDb\_mongodb.db\_findOne(collection, doc);

// System.out.println(dbObject);

String collection = "testCollection";

// for (int i = 0; i < 100; i++) {

// oDb\_mongodb.db\_insert(collection, new BasicDBObject("i", i));

// }

System.*out*.println(oDb\_mongodb.db\_count(collection, **null**));

// test update

BasicDBObject querydoc = **new** BasicDBObject("i", 9);

BasicDBObject updatedoc = **new** BasicDBObject("$set", **new** BasicDBObject(

"name", "rong").append("age", 100));

oDb\_mongodb.db\_update(collection, querydoc, updatedoc, **false**);

// test query

BasicDBObject queryBasicDBObject = **new** BasicDBObject("i",

**new** BasicDBObject("$lt", 20));

DBCursor cursor = oDb\_mongodb

.db\_findAll(collection, queryBasicDBObject);

oDb\_mongodb.show(cursor);

// test remove

BasicDBObject removedoc = **new** BasicDBObject("i", 13);

oDb\_mongodb.db\_remove(collection, removedoc);

}

}

详细可参见mongodb的官方文档

http://docs.mongodb.org/ecosystem/tutorial/getting-started-with-java-driver/