

# MongoDB学习笔记番外篇：JDBC方式操作MongoDB示例 - CSDN博客

## 1. pom.xml中依赖包设置

在项目中引入MongoDB JDBC 驱动mongo-java-driver-X.X.X.jar

[html]view plaincopy

```
1. <dependency>
2. <groupId>org.mongodbgroupId>
3. <artifactId>mongo-java-driverartifactId>
4. <version>3.4.0version>
5. </dependency>
```

## 2. 示例类

[java]view plaincopy

```
1. package memo.java.by.eric;
2. import java.util.ArrayList;
3. import java.util.List;
4. import org.bson.Document;
5. import com.mongodb.MongoClient;
6. import com.mongodb.client.FindIterable;
7. import com.mongodb.client.MongoCollection;
8. import com.mongodb.client.MongoCursor;
9. import com.mongodb.client.MongoDatabase;
10. import com.mongodb.client.model.Filters;
11. publicclass MongoDBJDBCDemo {
12. publicstaticvoid main(String args[]) {
13. try {
14. // 连接到 mongodb 服务
15. MongoClient mongoClient = new MongoClient("localhost", 27017);
16. // 连接到数据库hello
17. // 如果指定的数据库不存在，MongoDB会自动创建数据库
18. MongoDatabase mongoDatabase = mongoClient.getDatabase("hello");
19. System.out.println("连接到数据库hello");
20. // 创建集合
21. mongoDatabase.createCollection("students");
22. System.out.println("集合students创建成功");
23. // 选择集合
24. MongoCollection collection = mongoDatabase.getCollection("students");
25. System.out.println("集合students选择成功");
26. // 插入文档
27. /**
28. * 1. 创建文档 org.bson.Document 参数为key-value的格式 2. 创建文档集合List
29. * 3. 将文档集合插入数据库集合中 mongoCollection.insertMany(List)
30. * 插入单个文档可以用 mongoCollection.insertOne(Document)
31. */
32. Document document = new Document("name", "老司机").append("age", 31);
33. List documents = new ArrayList();
34. documents.add(document);
35. collection.insertMany(documents);
36. System.out.println("文档插入成功");
37. // 更新文档
38. // 将文档中age=31的文档修改为age=32
39. collection.updateMany(Filters.eq("age", 31), new Document("$set", new
Document("age", 32)));
```

```
40.         System.out.println("文档更新成功");
41. // 检索所有文档
42. /**
43.         * 1. 获取迭代器FindIterable 2. 获取游标MongoCursor 3.
44.         * 通过游标遍历检索出的文档集合
45.         */
46.         FindIterable findIterable = collection.find();
47.         MongoCursor mongoCursor = findIterable.iterator();
48. while (mongoCursor.hasNext()) {
49.         System.out.println(mongoCursor.next());
50.     }
51. // 删除符合条件的第一个文档
52.         collection.deleteOne(Filters.eq("name", "老司机"));
53. // 删除所有符合条件的文档
54.         collection.deleteMany(Filters.eq("name", "老司机"));
55. // 关闭连接
56.         mongoClient.close();
57.     } catch (Exception e) {
58.         System.err.println(e.getClass().getName() + ": " + e.getMessage());
59.     }
60. }
61. }
```