Spring Data MongoDB : Query document

By mkyong (http://www.mkyong.com/author/mkyong/) | May 19, 2011 | Updated : April 8, 2013 | Viewed : 170,323 times +508 pv/w

Here we show you a few examples to query documents from MongoDB, by using Query, Criteria and along with some of the common operators.

Test Data

```
> db.users.find()
{ "_id" : ObjectId("id"), "ic" : "1001", "name" : "ant", "age" : 10 }
{ "_id" : ObjectId("id"), "ic" : "1002", "name" : "bird", "age" : 20 }
{ "_id" : ObjectId("id"), "ic" : "1003", "name" : "cat", "age" : 30 }
{ "_id" : ObjectId("id"), "ic" : "1004", "name" : "dog", "age" : 40 }
{ "_id" : ObjectId("id"), "ic" : "1005", "name" : "elephant", "age" : 50 }
{ "_id" : ObjectId("id"), "ic" : "1006", "name" : "frog", "age" : 60 }
```

P.S This example is tested under `mongo-java-driver-2.11.0.jar` and `spring-data-mongodb-1.2.0.RELEASE.jar`

1. BasicQuery example

If you are familiar with the core MongoDB console find() command, just put the "raw" query inside the `BasicQuery`.

```
BasicQuery query1 = new BasicQuery("{ age : { $lt : 40 }, name : 'cat' }");
User userTest1 = mongoOperation.findOne(query1, User.class);

System.out.println("query1 - " + query1.toString());
System.out.println("userTest1 - " + userTest1);
```

Output

```
query1 - Query: { "age" : { "$lt" : 40} , "name" : "cat"}, Fields: null, Sort: { }
userTest1 - User [id=id, ic=1003, name=cat, age=30]
```

2. findOne example

findOne will return the single document that matches the query, and you can a combine few criteria with `Criteria.and()` method. See example 4 for more details.

```
Query query2 = new Query();
query2.addCriteria(Criteria.where("name").is("dog").and("age").is(40));

User userTest2 = mongoOperation.findOne(query2, User.class);
System.out.println("query2 - " + query2.toString());
System.out.println("userTest2 - " + userTest2);
```

Output

```
query2 - Query: { "name" : "dog" , "age" : 40}, Fields: null, Sort: null
userTest2 - User [id=id, ic=1004, name=dog, age=40]
```

3. find and $inc example

Find and return a list of documents that match the query. This example also shows the use of `$inc` operator.

```
List<Integer> listOfAge = new ArrayList<Integer>();
listOfAge.add(10);
listOfAge.add(30);
listOfAge.add(40);

Query query3 = new Query();
query3.addCriteria(Criteria.where("age").in(listOfAge));

List<User> userTest3 = mongoOperation.find(query3, User.class);
System.out.println("query3 - " + query3.toString());

for (User user : userTest3) {
    System.out.println("userTest3 - " + user);
}
```

Output

```
query3 - Query: { "age" : { "$in" : [ 10 , 30 , 40]}}, Fields: null, Sort: null
userTest3 - User [id=id, ic=1001, name=ant, age=10]
userTest3 - User [id=id, ic=1003, name=cat, age=30]
userTest3 - User [id=id, ic=1004, name=dog, age=40]
```

4. find and $gt, $lt, $and example

Find and return a list of documents that match the query. This example also shows the use of `$gt` , `$lt` and `$and` operators.

```
Query query4 = new Query();
query4.addCriteria(Criteria.where("age").lt(40).and("age").gt(10));

List<User> userTest4 = mongoOperation.find(query4, User.class);
System.out.println("query4 - " + query4.toString());

for (User user : userTest4) {
    System.out.println("userTest4 - " + user);
}
```

Oppss, an error message is generated, the API doen’t works in this way :)

```
Due to limitations of the com.mongodb.BasicDBObject, you can't add a second 'age' expression
specified as 'age : { "$gt" : 10}'. Criteria already contains 'age : { "$lt" : 40}'.
```

You can’t use `Criteria.and()` to add multiple criteria into the same field, to fix it, use `Criteria.andOperator()`, see updated example :

```
Query query4 = new Query();
query4.addCriteria(
    Criteria.where("age").exists(true)
    .andOperator(
        Criteria.where("age").gt(10),
                Criteria.where("age").lt(40)
    )
);

List<User> userTest4 = mongoOperation.find(query4, User.class);
System.out.println("query4 - " + query4.toString());

for (User user : userTest4) {
    System.out.println("userTest4 - " + user);
}
```

Output

```
query4 - Query: { "age" : { "$lt" : 40} , "$and" : [ { "age" : { "$gt" : 10}}]}, Fields: null, Sort: null
userTest4 - User [id=51627a0a3004cc5c0af72964, ic=1002, name=bird, age=20]
userTest4 - User [id=51627a0a3004cc5c0af72965, ic=1003, name=cat, age=30]
```

5. find and sorting example

Find and sort the result.

```
Query query5 = new Query();
query5.addCriteria(Criteria.where("age").gte(30));
query5.with(new Sort(Sort.Direction.DESC, "age"));

List<User> userTest5 = mongoOperation.find(query5, User.class);
System.out.println("query5 - " + query5.toString());

for (User user : userTest5) {
    System.out.println("userTest5 - " + user);
}
```

Output

```
query5 - Query: { "age" : { "$gte" : 30}}, Fields: null, Sort: { "age" : -1}
userTest5 - User [id=id, ic=1006, name=frog, age=60]
userTest5 - User [id=id, ic=1005, name=elephant, age=50]
userTest5 - User [id=id, ic=1004, name=dog, age=40]
userTest5 - User [id=id, ic=1003, name=cat, age=30]
```

6. find and $regex example

Find by regular expression pattern.

```
Query query6 = new Query();
query6.addCriteria(Criteria.where("name").regex("D.*G", "i"));

List<User> userTest6 = mongoOperation.find(query6, User.class);
System.out.println("query6 - " + query6.toString());

for (User user : userTest6) {
    System.out.println("userTest6 - " + user);
}
```

Output

```
query6 - Query: { "name" : { "$regex" : "D.*G" , "$options" : "i"}}, Fields: null, Sort: null
userTest6 - User [id=id, ic=1004, name=dog, age=40]
```

7. Full Example

A full example to combine everything from example 1 to 6.

SpringMongoConfig.java

```
package com.mkyong.config;

import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.data.mongodb.core.MongoTemplate;

import com.mongodb.MongoClient;

/**
 * Spring MongoDB configuration file
 *
 */
@Configuration
public class SpringMongoConfig{

    public @Bean
    MongoTemplate mongoTemplate() throws Exception {

        MongoTemplate mongoTemplate =
            new MongoTemplate(new MongoClient("127.0.0.1"),"yourdb");
        return mongoTemplate;

    }

}
```

User.java

```
package com.mkyong.model;

import java.util.Date;
import org.springframework.data.annotation.Id;
import org.springframework.data.mongodb.core.index.Indexed;
import org.springframework.data.mongodb.core.mapping.Document;
import org.springframework.format.annotation.DateTimeFormat;
import org.springframework.format.annotation.DateTimeFormat.ISO;

@Document(collection = "users")
public class User {

    @Id
    private String id;

    @Indexed
    private String ic;

    private String name;

    private int age;

    //getter, setter and constructor methods

}
```

QueryApp.java

```
package com.mkyong.core;

import java.util.ArrayList;
import java.util.List;

import org.springframework.context.ApplicationContext;
import org.springframework.context.annotation.AnnotationConfigApplicationContext;
import org.springframework.data.domain.Sort;
import org.springframework.data.mongodb.core.MongoOperations;
import org.springframework.data.mongodb.core.query.BasicQuery;
import org.springframework.data.mongodb.core.query.Criteria;
import org.springframework.data.mongodb.core.query.Query;

import com.mkyong.config.SpringMongoConfig;
import com.mkyong.model.User;

/**
 * Query example
 *
 * @author mkyong
 *
 */
public class QueryApp {

    public static void main(String[] args) {

        ApplicationContext ctx =
                         new AnnotationConfigApplicationContext(SpringMongoConfig.class);
        MongoOperations mongoOperation =
                         (MongoOperations) ctx.getBean("mongoTemplate");

        // insert 6 users for testing
        List<User> users = new ArrayList<User>();

        User user1 = new User("1001", "ant", 10);
        User user2 = new User("1002", "bird", 20);
        User user3 = new User("1003", "cat", 30);
        User user4 = new User("1004", "dog", 40);
        User user5 = new User("1005", "elephant",50);
        User user6 = new User("1006", "frog", 60);
        users.add(user1);
        users.add(user2);
        users.add(user3);
        users.add(user4);
        users.add(user5);
        users.add(user6);
        mongoOperation.insert(users, User.class);

        System.out.println("Case 1 - find with BasicQuery example");

        BasicQuery query1 = new BasicQuery("{ age : { $lt : 40 }, name : 'cat' }");
        User userTest1 = mongoOperation.findOne(query1, User.class);

        System.out.println("query1 - " + query1.toString());
        System.out.println("userTest1 - " + userTest1);

        System.out.println("\nCase 2 - find example");

        Query query2 = new Query();
        query2.addCriteria(Criteria.where("name").is("dog").and("age").is(40));

        User userTest2 = mongoOperation.findOne(query2, User.class);
        System.out.println("query2 - " + query2.toString());
        System.out.println("userTest2 - " + userTest2);

        System.out.println("\nCase 3 - find list $inc example");

        List<Integer> listOfAge = new ArrayList<Integer>();
        listOfAge.add(10);
        listOfAge.add(30);
        listOfAge.add(40);

        Query query3 = new Query();
        query3.addCriteria(Criteria.where("age").in(listOfAge));

        List<User> userTest3 = mongoOperation.find(query3, User.class);
        System.out.println("query3 - " + query3.toString());

        for (User user : userTest3) {
            System.out.println("userTest3 - " + user);
        }

        System.out.println("\nCase 4 - find list $and $lt, $gt example");

        Query query4 = new Query();

        // it hits error
        // query4.addCriteria(Criteria.where("age").lt(40).and("age").gt(10));

        query4.addCriteria(
                   Criteria.where("age").exists(true).andOperator(
                 Criteria.where("age").gt(10),
                         Criteria.where("age").lt(40)
                )
                );

        List<User> userTest4 = mongoOperation.find(query4, User.class);
        System.out.println("query4 - " + query4.toString());

        for (User user : userTest4) {
            System.out.println("userTest4 - " + user);
        }

        System.out.println("\nCase 5 - find list and sorting example");
        Query query5 = new Query();
        query5.addCriteria(Criteria.where("age").gte(30));
        query5.with(new Sort(Sort.Direction.DESC, "age"));

        List<User> userTest5 = mongoOperation.find(query5, User.class);
        System.out.println("query5 - " + query5.toString());

        for (User user : userTest5) {
            System.out.println("userTest5 - " + user);
        }

        System.out.println("\nCase 6 - find by regex example");
        Query query6 = new Query();
        query6.addCriteria(Criteria.where("name").regex("D.*G", "i"));

        List<User> userTest6 = mongoOperation.find(query6, User.class);
        System.out.println("query6 - " + query6.toString());

        for (User user : userTest6) {
            System.out.println("userTest6 - " + user);
        }

        mongoOperation.dropCollection(User.class);

    }

}
```

Output

```
Case 1 - find with BasicQuery example
query1 - Query: { "age" : { "$lt" : 40} , "name" : "cat"}, Fields: null, Sort: { }
userTest1 - User [id=id, ic=1003, name=cat, age=30]

Case 2 - find example
query2 - Query: { "name" : "dog" , "age" : 40}, Fields: null, Sort: null
userTest2 - User [id=id, ic=1004, name=dog, age=40]

Case 3 - find list $inc example
query3 - Query: { "age" : { "$in" : [ 10 , 30 , 40]}}, Fields: null, Sort: null
userTest3 - User [id=id, ic=1001, name=ant, age=10]
userTest3 - User [id=id, ic=1003, name=cat, age=30]
userTest3 - User [id=id, ic=1004, name=dog, age=40]

Case 4 - find list $and $lt, $gt example
query4 - Query: { "age" : { "$lt" : 40} , "$and" : [ { "age" : { "$gt" : 10}}]}, Fields: null, Sort: null
userTest4 - User [id=id, ic=1002, name=bird, age=20]
userTest4 - User [id=id, ic=1003, name=cat, age=30]

Case 5 - find list and sorting example
query5 - Query: { "age" : { "$gte" : 30}}, Fields: null, Sort: { "age" : -1}
userTest5 - User [id=id, ic=1006, name=frog, age=60]
userTest5 - User [id=id, ic=1005, name=elephant, age=50]
userTest5 - User [id=id, ic=1004, name=dog, age=40]
userTest5 - User [id=id, ic=1003, name=cat, age=30]

Case 6 - find by regex example
query6 - Query: { "name" : { "$regex" : "D.*G" , "$options" : "i"}}, Fields: null, Sort: null
userTest6 - User [id=id, ic=1004, name=dog, age=40]
```

Download Source Code

Download it – SpringMongoDB-Query-Example.zip (http://www.mkyong.com/wp-content/uploads/2011/05/SpringMongoDB-Query-Example.zip) (25 KB)

References

1. Spring data Mongodb – Query documents (http://static.springsource.org/spring-data/mongodb/docs/current/reference/html/mongo.core.html#mongo.query)
2. Java MongoDB query example/ (http://www.mkyong.com/mongodb/java-mongodb-query-document/)

Tags : mongodb (http://www.mkyong.com/tag/mongodb/) query (http://www.mkyong.com/tag/query/) spring-data (http://www.mkyong.com/tag/spring-data/)

Share this article on

Twitter (https://twitter.com/intent/tweet?text=Spring Data MongoDB : Query document&url=http://www.mkyong.com/mongodb/spring-data-mongodb-query-document/&via=mkyong) Facebook (https://www.facebook.com/sharer/sharer.php?u=http://www.mkyong.com/mongodb/spring-data-mongodb-query-document/) Google+ (https://plus.google.com/share?url=http://www.mkyong.com/mongodb/spring-data-mongodb-query-document/)

About the Author

mkyong

Founder of Mkyong.com (http://mkyong.com), love Java and open source stuff. Follow him on Twitter (https://twitter.com/mkyong), or befriend him on Facebook (http://www.facebook.com/java.tutorial) or Google Plus (https://plus.google.com/110948163568945735692?rel=author). If you like my tutorials, consider make a donation to these charities (http://www.mkyong.com/blog/donate-to-charity/).

Related Posts

76k	Spring Data MongoDB : Insert document (/mongodb/spring-data-mongodb-insert-document/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=0)
337k	Spring Data MongoDB hello world example (/mongodb/spring-data-mongodb-hello-world-example/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=1)
96k	Spring Data MongoDB : Update document (/mongodb/spring-data-mongodb-update-document/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=2)
47k	Spring Data MongoDB : Save binary file, GridFS example (/mongodb/spring-data-mongodb-save-binary-file-gridfs-example/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=3)
24k	Spring Data MongoDB remove _class column (/mongodb/spring-data-mongodb-remove-_class-column/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=4)
38k	Spring Data MongoDB : get last modified records (date sorting) (/mongodb/spring-data-mongodb-get-last-modified-records-date-sorting/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=5)
25k	Spring Data MongoDB - Select fields to return (/mongodb/spring-data-mongodb-select-fields-to-return/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=6)
58k	Spring Data MongoDB - Aggregation Grouping Example (/mongodb/spring-data-mongodb-aggregation-grouping-example/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=7)
63k	Spring Boot + Spring Data MongoDB example (/spring-boot/spring-boot-spring-data-mongodb-example/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=8)

Popular Posts

2m

Spring MVC Tutorial (/tutorials/spring-mvc-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=1)

995k

Java Date and Calendar examples (/java/java-date-and-calendar-examples/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=2)

536k

10 Java Regular Expression Examples You Should Know (/regular-expressions/10-java-regular-expression-examples-you-should-know/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=3)

906k

How to create a Web Application Project with Maven (/maven/how-to-create-a-web-application-project-with-maven/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=4)

758k

How to read XML file in Java - (SAX Parser) (/java/how-to-read-xml-file-in-java-sax-parser/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=5)

539k

Android GridView example (/android/android-gridview-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=6)

651k

Spring Security hello world example (/spring-security/spring-security-hello-world-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=7)

956k

log4j.properties example (/logging/log4j-log4j-properties-examples/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=8)

1.1m

Android Tutorial (/tutorials/android-tutorial/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=9)

783k

JSF 2.0 hello world example (/jsf2/jsf-2-0-hello-world-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=10)

533k

Android TableLayout example (/android/android-tablelayout-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=11)

897k

Struts 2 Tutorial (/tutorials/struts-2-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=12)

740k

Maven + Spring + Hibernate + MySql Example (/spring/maven-spring-hibernate-mysql-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=13)

1m

JavaMail API - Sending email via Gmail SMTP example (/java/javamail-api-sending-email-via-gmail-smtp-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=14)

801k

Android custom dialog example (/android/android-custom-dialog-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=15)

2.3m

Spring Tutorial (/tutorials/spring-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=16)

682k

Android ListView example (/android/android-listview-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=17)

751k

how to convert Byte[] array to String in Java (/java/how-do-convert-byte-array-to-string-in-java/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=18)

812k

Java XML Tutorial (/tutorials/java-xml-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=19)

531k

How to export data to CSV file - Java (/java/how-to-export-data-to-csv-file-java/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=20)

Leave a Reply

Join the discussion

Sort by: newest | oldest | most voted

shridevi

Guest

if suppose record not found then what will be the result of find()? will it return empty list?

👍 0 👎 ↩ REPLY

5 months 5 days ago

sumit

Guest

Please correct me if I am wrong. It's a very useful website but thus this article specifies misplaced example for $inc operator?

👍 0 👎 ↩ REPLY

8 months 18 days ago

Carol

Guest

well explained

👍 0 👎 ↩ REPLY

1 year 6 months ago

PA

Guest

What we can do if we're not aware about field name or I need to search particular string in all fields of collection?

👍 0 👎 ↩ REPLY

2 years 2 months ago

Roma

Guest

Thanks! Mkyong you are the BesT!

👍 0 👎 ↩ REPLY

3 years 9 months ago

Arunkumar Gopalan

Guest

```
> db.ships.find({},{name:true,_id:false})
{ “name” : “USS Enterprise-D” }
{ “name” : “USS Prometheus” }
{ “name” : “USS Defiant” }
{ “name” : “IKS Buruk” }
{ “name” : “IKS Somraw” }
{ “name” : “Scimitar” }
{ “name” : “Narada” }
```

The above command how to do with spring? I need urgent

👍 0 👎 REPLY 3 years 10 months ago

ben narendren

Guest

You Rock, mkyong

👍 0 👎 REPLY 4 years 3 months ago

Ngan

Guest

Hi, thanks for a great tutorial.

Would you know how to set a max limit number of results returning back?
In Oracle we’d use setMaxResults(number) on the Query object.

Thanks!
Ngan

👍 0 👎 REPLY 4 years 6 months ago

shravan

Guest

query.with(new PageRequest(0, maxNumQuestions));

👍 0 👎 REPLY 2 years 4 months ago

Elisabeth T

Guest

I’m trying to create a dynamic query as this: Criteria.where(key1).is(value1 .and(key2).is(value2) If I use it the query works but if I try: new Criteria().andOperator(Criteria.where(key1).is(value1),Criteria.where(key2).is(value2)) It doesn’t works. I would want to use the second solutions because the number of keys and values aren’t fixed. Can someone tell me what’s the difference beetween these solutions or where is the error? Thanks too much. down vote favorite I’m trying to create a dynamic mongodb query using spring criteria. My query is : Criteria.where(key1).is(value1 .and(key2).is(value2) If I use it the query works but if I try: new Criteria().andOperator(Criteria.where(key1).is(value1),Criteria.where(key2).is(value2)) It doesn’t works. I would… Read more »

👍 0 👎 REPLY 5 years 3 months ago

mkyong (http://www.mkyong.com)

(http://www.mkyong.com/author/mkyong/)

Author

try `query.toString` to print out the query send to MongoDB, and see the different. I agreed that the and() and andOperator() are quite confusing, hope Spring data’ team will make it more make sense in future release.

And, also refer to example 4, it shows the use of $and operator. Hope help.

👍 0 👎 REPLY 4 years 8 months ago

Kael

Guest

how to config so that I can let the project print mongo query when using mongorepository of spring data mongodb

👍 0 👎 REPLY 2 years 8 months ago

maleo (http://www.prowebx.net)

Guest

Great write ups! it helps me resolve a long term issue on our system

👍 0 👎 ↩ REPLY

5 years 3 months ago

Marco Berri (http://tecnicume.blogspot.com)

Guest

Sorry, I did not understand, for sorting?

Thank you.

👍 0 👎 ↩ REPLY

5 years 10 months ago

mkyong (http://www.mkyong.com)

(http://www.mkyong.com/author/mkyong/)

Author

Refer to example 5 – find and sorting.

👍 0 👎 ↩ REPLY

4 years 8 months ago

Juhi Bhatia

Guest

Hi, I have just started up on spring-data-mongodb and am using version 1.0.0.M2 and mongo-java-driver 2.6.3. Your tutorial was very useful to start with spring-data. I am stuck on using sort() and findOne() together. My POJO to be inserted in mongodb is like this: public MongoTag(String tagName, String value, DataType dataType, long timestamp) { this.tagName = tagName; this.value = value; this.dataType = dataType; this.timestamp = timestamp; } now i need to retrieve say tagName=”water temperature” that has been written in the mongodb latest. So i need to sort on timestamp. I am using the following approach: Query query = new… Read more »

👍 0 👎 ↩ REPLY

6 years 3 months ago

mkyong (http://www.mkyong.com)

(http://www.mkyong.com/author/mkyong/)

Author

Try this ~

```
Query query = new Query();
query.addCriteria(Criteria.where("tagName").is('water temperature'));
query.with(new Sort(Sort.Direction.DESC, "timestamp"));
```

Refer to example 5 above, find and sorting example.

👍 0 👎 ↩ REPLY

4 years 8 months ago

dimann889s

Guest

how to use ‘*’ or ‘?’ as it use in sql. If I use
> db.book.find(“{‘*’:’*???*’}”).pretty()
{
“_id” : ObjectId(“51b61eee6b14af5c5e731924”),
“_class” : “com.library.model.Book”,
“title” : “???????? ????: ??????? ????”,
“authors” : “????????”,
“publish” : “??????”,
“category” : “?????”,
“volumeOfPages” : 123,
“dateOfReceipt” : ISODate(“2013-06-08T19:19:03.904Z”),
“countInStore” : 1,
“countOnHands” : 0
} or
> db.book.find(“{$regex:’???’, $option:’i’}”).pretty()
{
“_id” : ObjectId(“51b61eee6b14af5c5e731924”),
“_class” : “com.library.model.Book”,
“title” : “???????? ????: ??????? ????”,
“authors” : “????????”,
“publish” : “??????”,
“category” : “?????”,

“volumeOfPages” : 123,
“dateOfReceipt” : ISODate(“2013-06-08T19:19:03.904Z”),
“countInStore” : 1,
“countOnHands” : 0
}

it ok, if I use it java
BasicQuery query = new BasicQuery(“{$regex: ” + term + “, $option:’i’}”,);
return mongoTemplate.find(query, Book.class);

it threw java.lang.CastException what regexp not suitable for DBObject

👍 0 👎 ↩ REPLY 4 years 6 months ago

Areg

Guest

Were you able to find answer to this ?

👍 0 👎 ↩ REPLY 5 years 3 months ago

Java MongoDB Tutorial (http://www.mkyong.com/tutorials/java-mongodb-tutorials/)

[…] save(), updateFirst() and updateMulti() to update existing domain object from mongoDB database. Spring Data MongoDB : Query document Example to use Spring data findOne(), find() and getCollection() to get / query documents from […]

👍 0 👎 ↩ REPLY 6 years 6 months ago

Ricardo M Lino

Guest

Alguém já teve problema de recursão no mongo? Vou tentar explicar com um exemplo. public class Usuario { private GroupoDeUsuario grupo; ... //ggas } public class GrupoDeUsuario { private Usuario criador; ... //ggas } O problema é que na hora que ele vai salvar o Usuario, ele tenta salvar primeiro o Grupo e quando vai salvar o Grupo tenta salvar primeiro o usuario… e assim infinitamente… Alguem sabe como posso fazer isso usando o MongoDB?

👍 -1 👎 ↩ REPLY 5 years 2 months ago

mkyong (http://www.mkyong.com)

(http://www.mkyong.com/author/mkyong/)

Author

Portuguese? Translate your message from Google translate. The relationship is not really make sense to me, mind to give me a use case?

Btw, In Spring data, you can use `@Transient` to tell Spring to ignore the field.

```
import org.springframework.data.annotation.Transient;

public class GrupoDeUsuario {

   @Transient
   private Usuario criador;
    ...
    //ggas
}
```

In above example, when GrupoDeUsuario is saved, Usuario will be ignored.

👍 0 👎 ↩ REPLY 4 years 8 months ago

HUAWEI
honor
ЗИМНИЙ ЦЕНОПАД!
Снижение цен на смартфоны и планшеты!
ООО «Техкомпания Хуавэй», ОГРН 1027739023212, Москва, ул. Крылатская, д.17, к.2
shop.huawei.ru

HUAWEI
honor
ЗИМНИЙ ЦЕНОПАД!
Снижение цен на смартфоны и планшеты!
ООО «Техкомпания Хуавэй», ОГРН 1027739023212, Москва, ул. Крылатская, д.17, к.2
shop.huawei.ru

Rising Posts (100k-500k pv)

+214.9k	Spring Boot Hello World Example - JSP (/spring-boot/spring-boot-hello-world-example-jsp/?utm_source=mkyong&utm_medium=sidebar&utm_campaign=100k-500k&utm_content=0)
+154.4k	Spring Boot Hello World Example - Thymeleaf (/spring-boot/spring-boot-hello-world-example-thymeleaf/?utm_source=mkyong&utm_medium=sidebar&utm_campaign=100k-500k&utm_content=1)
+4.6k	Spring Boot Tutorials (/tutorials/spring-boot-tutorials/?utm_source=mkyong&utm_medium=sidebar&utm_campaign=100k-500k&utm_content=2)
+4.5k	Java 8 Streams filter examples (/java8/java-8-streams-filter-examples/?utm_source=mkyong&utm_medium=sidebar&utm_campaign=100k-500k&utm_content=3)
+3.4k	Java 8 Tutorials (/tutorials/java-8-tutorials/?utm_source=mkyong&utm_medium=sidebar&utm_campaign=100k-500k&utm_content=4)