



Spring Data MongoDB : Insert document

By mkyong (<http://www.mkyong.com/author/mkyong/>) | May 18, 2011 | Updated : April 6, 2013 | Viewed : 76,102 times +266 pv/w

In Spring data MongoDB, you can use `save()`, `insert()` to save a or a list of objects into mongoDB database.

```
User user = new User("...");

//save user object into "user" collection / table
//class name will be used as collection name
mongoOperation.save(user);

//save user object into "tableA" collection
mongoOperation.save(user, "tableA");

//insert user object into "user" collection
//class name will be used as collection name
mongoOperation.insert(user);

//insert user object into "tableA" collection
mongoOperation.insert(user, "tableA");

//insert a list of user objects
mongoOperation.insert(listofUser);
```

By default, if you saved an object, and didn't specified any of the "collection name", the class name will be used as the collection name.

1. Save and Insert

Should you use the Save or Insert?

1. Save – It should rename to `saveOrUpdate()`, it performs `insert()` if “_id” is NOT exist or `update()` if “_id” is existed”.
2. Insert – Only insert, if “_id” is existed, an error is generated.

See example below

```
//get an existed data, and update it
User userD1 = mongoOperation.findOne(
    new Query(Criteria.where("age").is(64)), User.class);
userD1.setName("new name");
userD1.setAge(100);

//if you insert it, 'E11000 duplicate key error index' error is generated.
//mongoOperation.insert(userD1);

//instead you should use save.
mongoOperation.save(userD1);
```



2. Insert documents example

See a full example to show you how to save a or a list of “user” object into MongoDB.

SpringMongoConfig.java – Create a mongoTemplate bean in Spring container.

```
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;

    }

}
```

Uses @Document to define a “collection name” when you save this object. In this case, when “user” object saves, it will save into “users” collection.

User.java

```
package com.mkyong.user;

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;

    @DateTimeFormat(iso = ISO.DATE_TIME)
    private Date createdDate;

    //getter and setter methods
}
```

Full examples to show you different ways of inserting data, read code and comments for self-explanatory.

App.java

```
package com.mkyong.core;

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

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

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

public class App {

    public static void main(String[] args) {
        // For Annotation
        ApplicationContext ctx =
            new AnnotationConfigApplicationContext(SpringMongoConfig.class);
        MongoOperations mongoOperation =
            (MongoOperations) ctx.getBean("mongoTemplate");

        // Case1 - insert a user, put "tableA" as collection name
        System.out.println("Case 1...");
        User userA = new User("1000", "apple", 54, new Date());
        mongoOperation.save(userA, "tableA");

        // find
        Query findUserQuery = new Query();
        findUserQuery.addCriteria(Criteria.where("ic").is("1000"));
        User userA1 = mongoOperation.findOne(findUserQuery, User.class, "tableA");
        System.out.println(userA1);

        // Case2 - insert a user, put entity as collection name
        System.out.println("Case 2...");
        User userB = new User("2000", "orange", 64, new Date());
        mongoOperation.save(userB);

        // find
        User userB1 = mongoOperation.findOne(
            new Query(Criteria.where("age").is(64)), User.class);
        System.out.println(userB1);

        // Case3 - insert a list of users
        System.out.println("Case 3...");
        User userC = new User("3000", "metallica", 34, new Date());
        User userD = new User("4000", "metallica", 34, new Date());
        User userE = new User("5000", "metallica", 34, new Date());
        List<User> userList = new ArrayList<User>();
        userList.add(userC);
        userList.add(userD);
        userList.add(userE);
        mongoOperation.insert(userList, User.class);

        // find
        List<User> users = mongoOperation.find(
            new Query(Criteria.where("name").is("metallica")),
            User.class);

        for (User temp : users) {
            System.out.println(temp);
        }

        //save vs insert
        System.out.println("Case 4...");
        User userD1 = mongoOperation.findOne(
            new Query(Criteria.where("age").is(64)), User.class);
        userD1.setName("new name");
        userD1.setAge(100);

        //E11000 duplicate key error index, _id existed
        //mongoOperation.insert(userD1);
        mongoOperation.save(userD1);
        User userE1 = mongoOperation.findOne(
```

```

        new Query(Criteria.where("age").is(100)), User.class);
    System.out.println(userE1);
}

}

```

Output

```

Case 1...
User [id=id, ic=1000, name=apple, age=54, createdAt=Sat Apr 06 12:35:15 MYT 2013]
Case 2...
User [id=id, ic=2000, name=orange, age=64, createdAt=Sat Apr 06 12:59:19 MYT 2013]
Case 3...
User [id=id, ic=3000, name=metallica, age=34, createdAt=Sat Apr 06 12:59:19 MYT 2013]
User [id=id, ic=4000, name=metallica, age=34, createdAt=Sat Apr 06 12:59:19 MYT 2013]
User [id=id, ic=5000, name=metallica, age=34, createdAt=Sat Apr 06 12:59:19 MYT 2013]
Case 4...
User [id=id, ic=2000, name=new name, age=100, createdAt=Sat Apr 06 12:59:19 MYT 2013]

```



iHerb

3. Mongo Console

Review Mongo console, see what are inserted and created.

```

> mongo
MongoDB shell version: 2.2.3
connecting to: test

> show dbs
admin    0.203125GB
yourdb   0.203125GB

> use yourdb
switched to db yourdb
> show collections
system.indexes
tableA
users

> db.tableA.find()
{ "_id" : ObjectId("id"), "_class" : "com.mkyong.user.User",
  "ic" : "1000", "name" : "apple", "age" : 54, "createdAt" : ISODate("2013-04-06T05:04:06.384Z") }

> db.users.find()
{ "_id" : ObjectId("id"), "_class" : "com.mkyong.user.User",
  "ic" : "3000", "name" : "metallica", "age" : 34, "createdAt" : ISODate("2013-04-06T05:04:06.735Z") }
{ "_id" : ObjectId("id"), "_class" : "com.mkyong.user.User",
  "ic" : "4000", "name" : "metallica", "age" : 34, "createdAt" : ISODate("2013-04-06T05:04:06.735Z") }
{ "_id" : ObjectId("id"), "_class" : "com.mkyong.user.User",
  "ic" : "5000", "name" : "metallica", "age" : 34, "createdAt" : ISODate("2013-04-06T05:04:06.735Z") }
{ "_id" : ObjectId("id"), "_class" : "com.mkyong.user.User",
  "ic" : "2000", "name" : "new name", "age" : 100, "createdAt" : ISODate("2013-04-06T05:04:06.731Z") }

```

P.S To remove the extra _class column, read this article – Spring Data MongoDB Remove _class Column (http://www.mkyong.com/mongodb/spring-data-mongodb-remove-_class-column/).

Download Source Code

Download it – SpringData-MongoDB-Insert-Example.zip (<http://www.mkyong.com/wp-content/uploads/2011/05/SpringMongoDB-Insert-Example.zip>) (24KB)

References

1. Spring Data MongoDB – Saving, Updating, and Removing Documents (<http://static.springsource.org/spring-data/mongodb/docs/current/reference/html/mongo.core.html#mongo-template.save-update-remove>)
2. Spring Data MongoDB Hello World Example (<http://www.mkyong.com/mongodb/spring-data-mongodb-hello-world-example/>)

Tags : [insert](http://www.mkyong.com/tag/insert/) (<http://www.mkyong.com/tag/insert/>) [mongodb](http://www.mkyong.com/tag/mongodb/) (<http://www.mkyong.com/tag/mongodb/>) [spring-data](http://www.mkyong.com/tag/spring-data/) (<http://www.mkyong.com/tag/spring-data/>)

Share this article on

Twitter ([https://twitter.com/intent/tweet?text=Spring Data MongoDB : Insert document&url=http://www.mkyong.com/mongodb/spring-data-mongodb-insert-document/&via=mkyong](https://twitter.com/intent/tweet?text=Spring+Data+MongoDB+%3A+Insert+document&url=http://www.mkyong.com/mongodb/spring-data-mongodb-insert-document/&via=mkyong)) Facebook (<https://www.facebook.com/sharer/sharer.php?u=http://www.mkyong.com/mongodb/spring-data-mongodb-insert-document/>) Google+ (<https://plus.google.com/share?url=http://www.mkyong.com/mongodb/spring-data-mongodb-insert-document/>)

DO YOU REALLY KNOW YOUR MOBILES?

How many times does the average person unlock their phone a day?

15

▼

A 20

B 110

C 55

D 201

Your Score 0 Question 1/10

Get mortgage help

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

- | | |
|------|---|
| 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=0) |
| 96k | Spring Data MongoDB : Update document (/mongodb/spring-data-mongodb-update-document/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=1) |
| 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=2) |
| 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=3) |
| 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=4) |
| 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=5) |
| 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=6) |
| 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=7) |
| 0 | JPA Insert + Oracle Sequences example (/jpa/jpa-insert-oracle-sequences-example/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=8) |

Popular Posts

	501k
Hibernate - fetching strategies examples (/hibernate/hibernate-fetching-strategies-examples/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=1)	
	897k
Struts 2 Tutorial (/tutorials/struts-2-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=2)	
	1.6m
How to send HTTP request GET/POST in Java (/java/how-to-send-http-request-getpost-in-java/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=3)	
	764k
Hibernate – One-to-Many example (Annotation) (/hibernate/hibernate-one-to-many-relationship-example-annotation/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=4)	
	740k
Maven + Spring + Hibernate + MySql Example (/spring/maven-spring-hibernate-mysql-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=5)	
	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=6)	
	1.4m
JSON.simple example - Read and write JSON (/java/json-simple-example-read-and-write-json/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=7)	
	1.3m
Java object sorting example (Comparable and Comparator) (/java/java-object-sorting-example-comparable-and-comparator/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=8)	
	516k
JAX-WS : wsimport tool example (/webservices/jax-ws/jax-ws-wsimport-tool-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=9)	
	903k
Hibernate Query examples (HQL) (/hibernate/hibernate-query-examples-hql/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=10)	
	526k
Hibernate - One-to-Many example (XML Mapping) (/hibernate/hibernate-one-to-many-relationship-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=11)	
	1.8m
Java Properties file examples (/java/java-properties-file-examples/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=12)	
	642k
Spring + JDBC example (/spring/maven-spring-jdbc-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=13)	
	823k
Spring MVC hello world example (/spring-mvc/spring-mvc-hello-world-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=14)	
	1.4m
JAX-RS Tutorial (/tutorials/jax-rs-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=15)	
	956k
log4j.properties example (/logging/log4j-log4j-properties-examples/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=16)	
	825k
Spring JdbcTemplate Querying examples (/spring/spring-jdbctemplate-querying-examples/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=17)	
	545k
How to access JSON object in JavaScript (/javascript/how-to-access-json-object-in-javascript/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=18)	
	532k
Java MongoDB Tutorial (/tutorials/java-mongodb-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=19)	
	504k
How to install Maven on Ubuntu (/maven/how-to-install-maven-in-ubuntu/?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](#)



???

hello, your tutorial is so great..I have a question. When we use the method findAll? do we have to define a class before? If so how to determine its properties? I am Chinese, English is not well. thanks in advance.

Guest



🕒 2 months 5 days ago



Mohan Reddy

date is saving 8 hrs less then current time, how to fix this one?

Guest



🕒 7 months 16 days ago



Carol

Thanks for sharing

Guest



🕒 1 year 6 months ago



Lingaraj Nayak

Hi Mykong I am new to mongoDB and JAVA. I had gone through some your blogs and I understood that my objective of inserting a pdf in gridFS method is well suitable in Spring framework. Can you tell me the step by step procedure how to setup spring in eclipse kepler 4.3 and there by the example how I insert the pdf and then view the pdf. Thank you in advance.

Guest



🕒 2 years 5 months ago



Arvind Das

Hi, In my case , making a separate MongoConfiguration class and extending it with AbstractMongoConfiguration did not solve the problem. It always gets connected to test db of mongo. what possibly might have gone wrong ? Any ideas!

Guest



🕒 2 years 9 months ago



Sharvari

Really found useful to me!

Guest



🕒 3 years 11 months ago



Abhishek

Hi,
Thanks for the quick tutorial. Really helped me in getting started with Mongo+Spring

Guest



🕒 4 years 2 months ago



ajit


```
User userD1 = mongoOperation.findOne(
    new Query(Criteria.where("age").is(64)), User.class);
```

Why do we use User.class?
Any help will be appreciated!
thanks in Advance :)

Guest




🕒 4 years 6 months ago



Java APIs for Developers


Office formats and Native Java
Code Free support and free
evaluation

aspose.com



Успейте сыграть


Vtemnote игра в которую хочется сыграть еще. 99,9% Гостей возвращаются снова. Проверь!
6+




Mkyong
55,036 likes

Like Page Sign Up

Be the first of your friends to like this





DART-6UL

Variscite

- ▶ NXP i.MX 6UL / 6ULL up to 900MHz Cortex-A7
- ▶ Dual LAN, USB, CAN
- ▶ Low power < 5mA
- ▶ Small size 25x50mm
- ▶ Certified WiFi/BT 802.11 ac/b/g/n

\$24