# Java MongoDB : Update document

By mkyong (http://www.mkyong.com/author/mkyong/) | May 10, 2011 | Updated : April 2, 2013 | Viewed : 142,948 times +357 pv/w

In this tutorial, we show you how to use Java MongoDB API `collection.update()` to update documents.

## Test Data

Assume following data / documents are inserted.

```
{
    "hosting" : "hostA",
    "type" : "vps",
    "clients" : 1000
},
{
    "hosting" : "hostB",
    "type" : "dedicated server",
    "clients" : 100
},
{
    "hosting" : "hostC",
    "type" : "vps",
    "clients" : 900
}
```

```
{ "_id" : { "$oid" : "id"} , "hosting" : "hostA" , "type" : "vps" , "clients" : 1000}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostB" , "type" : "dedicated server" , "clients" : 100}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}
```

## 1. DBCollection.update() with $set

Find document where hosting = ‘hostB’, and update it’s clients values from 100 to 110.

```
BasicDBObject newDocument = new BasicDBObject();
newDocument.put("clients", 110);

BasicDBObject searchQuery = new BasicDBObject().append("hosting", "hostB");

collection.update(searchQuery, newDocument);
```

*Output*

```
{ "_id" : { "$oid" : "id"} , "hosting" : "hostA" , "type" : "vps" , "clients" : 1000}
{ "_id" : { "$oid" : "id"} , "clients" : 110}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}
```

**The document is replaced!?**
Wait, the entire “hostB” document is replaced with another new document, this is not what we want.

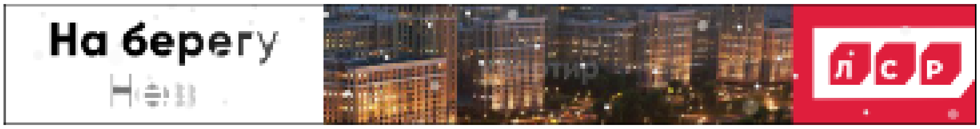
To update a particular value only, uses `$set` update modifier.

```
BasicDBObject newDocument = new BasicDBObject();
newDocument.append("$set", new BasicDBObject().append("clients", 110));

BasicDBObject searchQuery = new BasicDBObject().append("hosting", "hostB");

collection.update(searchQuery, newDocument);
```

*Output*

```
{ "_id" : { "$oid" : "id"} , "hosting" : "hostA" , "type" : "vps" , "clients" : 1000}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostB" , "type" : "dedicated server" , "clients" : 110}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}
```

**Note**
The MongoDB team should create another new API named `DBCollection.replace()`, many beginners are trapped in this `DBCollection.update()` API and replace the entire document accidentally. Again, to update a particular value, use `$set` .



## 2. DBCollection.update() with $inc

This example show the use of `$inc` modifier to increase a particular value. Find document where hosting = 'hostB', update it's 'clients' value by increasing the value from 100 to 199, (100 + 99) = 199.

```
BasicDBObject newDocument =
    new BasicDBObject().append("$inc",
    new BasicDBObject().append("total clients", 99));

collection.update(new BasicDBObject().append("hosting", "hostB"), newDocument);
```

*Output*

```
{ "_id" : { "$oid" : "id"} , "hosting" : "hostA" , "type" : "vps" , "clients" : 1000}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostB" , "type" : "dedicated server" , "clients" : 199}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}
```

## 3. DBCollection.update() with multi

This example show the use of `multi` parameter to update a set of matched documents. Find document where type = 'vps', update all the matched documents' 'clients' value to 888.

```
BasicDBObject updateQuery = new BasicDBObject();
updateQuery.append("$set",
    new BasicDBObject().append("clients", "888"));

BasicDBObject searchQuery = new BasicDBObject();
searchQuery.append("type", "vps");

collection.updateMulti(searchQuery, updateQuery);

//below statement set multi to true.
//collection.update(searchQuery, updateQuery, false, true);
```

*Output*

```
{ "_id" : { "$oid" : "id"} , "hosting" : "hostA" , "clients" : "888" , "type" : "vps"}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostB" , "type" : "dedicated server" , "clients" : 100}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "clients" : "888" , "type" : "vps"}
```

**Note**
If update without the `multi` set to true.

```
BasicDBObject updateQuery = new BasicDBObject();
updateQuery.append("$set",
    new BasicDBObject().append("clients", "888"));

BasicDBObject searchQuery = new BasicDBObject();
searchQuery.append("type", "vps");

collection.update(searchQuery, updateQuery);
```

You will noticed that only the first matched document is updated.

```
{"_id":{ "$oid" : "x"} , "hosting" : "hostA" , "clients" : "888" , "type" : "vps"}
{"_id":{ "$oid" : "x"} , "hosting" : "hostB" , "type" : "dedicated server" , "clients" : 100}
{"_id":{ "$oid" : "x"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}
```

To update a set of matched documents, you need to set “ `multi` ” to true.

## 4. Full Example

Full example by combining above code snippets.

```
package com.mkyong.core;

import java.net.UnknownHostException;
import com.mongodb.BasicDBObject;
import com.mongodb.DB;
import com.mongodb.DBCollection;
import com.mongodb.DBCursor;
import com.mongodb.Mongo;
import com.mongodb.MongoException;

/**
 * Java MongoDB update document
 *
 * @author mkyong
 *
 */

public class UpdateApp {

    public static void printAllDocuments(DBCollection collection) {
        DBCursor cursor = collection.find();
        while (cursor.hasNext()) {
            System.out.println(cursor.next());
        }
    }

    public static void removeAllDocuments(DBCollection collection) {
        collection.remove(new BasicDBObject());
    }

    public static void insertDummyDocuments(DBCollection collection) {
        BasicDBObject document = new BasicDBObject();
        document.put("hosting", "hostA");
        document.put("type", "vps");
        document.put("clients", 1000);

        BasicDBObject document2 = new BasicDBObject();
        document2.put("hosting", "hostB");
        document2.put("type", "dedicated server");
        document2.put("clients", 100);

        BasicDBObject document3 = new BasicDBObject();
        document3.put("hosting", "hostC");
        document3.put("type", "vps");
        document3.put("clients", 900);

        collection.insert(document);
        collection.insert(document2);
        collection.insert(document3);
    }

    public static void main(String[] args) {

    try {

      Mongo mongo = new Mongo("localhost", 27017);
      DB db = mongo.getDB("yourdb");

      // get a single collection
      DBCollection collection = db.getCollection("dummyColl");

      System.out.println("Testing 1...no $set");

      insertDummyDocuments(collection);

      // find hosting = hostB, and update the clients to 110
      BasicDBObject newDocument = new BasicDBObject();
      newDocument.put("clients", 110);

      BasicDBObject searchQuery = new BasicDBObject().append("hosting", "hostB");

      collection.update(searchQuery, newDocument);

      printAllDocuments(collection);
      removeAllDocuments(collection);
      System.out.println("\nTesting 1...with $set");

      insertDummyDocuments(collection);

      BasicDBObject updateDocument = new BasicDBObject();
      updateDocument.append("$set", new BasicDBObject().append("clients", 110));

      BasicDBObject searchQuery2 = new BasicDBObject().append("hosting", "hostB");

      collection.update(searchQuery2, updateDocument);

      printAllDocuments(collection);
      removeAllDocuments(collection);

      System.out.println("\nTesting 2... with $inc");
      insertDummyDocuments(collection);
      // find hosting = hostB and increase it's "clients" value by 99
      BasicDBObject newDocument2 = new BasicDBObject().append("$inc",
        new BasicDBObject().append("clients", 99));

      collection.update(new BasicDBObject().append("hosting", "hostB"), newDocument2);

      printAllDocuments(collection);
      removeAllDocuments(collection);

      System.out.println("\nTesting 3... with $multi");

      insertDummyDocuments(collection);
      // find type = vps , update all matched documents , clients value to 888
      BasicDBObject updateQuery = new BasicDBObject();
      updateQuery.append("$set", new BasicDBObject().append("clients", "888"));

      BasicDBObject searchQuery3 = new BasicDBObject();
      searchQuery3.append("type", "vps");

      collection.updateMulti(searchQuery3, updateQuery);
      // collection.update(searchQuery3, updateQuery, false, true);

      printAllDocuments(collection);
      removeAllDocuments(collection);

      System.out.println("Done");

    } catch (UnknownHostException e) {
        e.printStackTrace();
    } catch (MongoException e) {
        e.printStackTrace();
    }

    }
}
```

Output

```
Testing 1...no $set
{ "_id" : { "$oid" : "id"} , "hosting" : "hostA" , "type" : "vps" , "clients" : 1000}
{ "_id" : { "$oid" : "id"} , "clients" : 110}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}

Testing 1...with $set
{ "_id" : { "$oid" : "id"} , "hosting" : "hostA" , "type" : "vps" , "clients" : 1000}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostB" , "type" : "dedicated server" , "clients" : 110}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}

Testing 2... with $inc
{ "_id" : { "$oid" : "id"} , "hosting" : "hostA" , "type" : "vps" , "clients" : 1000}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostB" , "type" : "dedicated server" , "clients" : 199}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}

Testing 3... with $multi
{ "_id" : { "$oid" : "id"} , "hosting" : "hostB" , "type" : "dedicated server" , "clients" : 100}
{ "_id" : { "$oid" : "id"} , "hosting" : "hostC" , "type" : "vps" , "clients" : 900}
{ "_id" : { "$oid" : "id"} , "clients" : "888" , "hosting" : "hostA" , "type" : "vps"}
Done
```

## References

1. How to do updating in MongoDB (http://docs.mongodb.org/manual/applications/update/)
2. $set update modifier (http://docs.mongodb.org/manual/reference/operator/set/#op._S_set)
3. $inc update modifier (http://docs.mongodb.org/manual/reference/operator/inc/#op._S_inc)
4. Java MongoDB APIs , DBCollection JavaDoc (http://api.mongodb.org/java/current/com/mongodb/DBCollection.html)

Tags : mongodb (http://www.mkyong.com/tag/mongodb/) update (http://www.mkyong.com/tag/update/)

## Share this article on

Twitter (https://twitter.com/intent/tweet?text=Java MongoDB : Update document&url=http://www.mkyong.com/mongodb/java-mongodb-update-document/&via=mkyong) Facebook (https://www.facebook.com/sharer/sharer.php?u=http://www.mkyong.com/mongodb/java-mongodb-update-document/) Google+ (https://plus.google.com/share?url=http://www.mkyong.com/mongodb/java-mongodb-update-document/)

SMOKEY
BRING IT IN FOR A BIRTHDAY BEAR HUG
HAPPY BIRTHDAY 70 YEARS SINCE 1944
SMOKEYBEAR.COM
Ad Council

## 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

| | |
|---|---|
| 94k | Java MongoDB : Save image example (/mongodb/java-mongodb-save-image-example/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=0) |
| 139k | Java MongoDB : Convert JSON data to DBObject (/mongodb/java-mongodb-convert-json-data-to-dbobject/?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) |
| 171k | MongoDB import and export example (/mongodb/mongodb-import-and-export-example/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=4) |
| 9k | MongoDB - Remove a field from array (/mongodb/mongodb-remove-a-field-from-array-documents/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=5) |
| 38k | MongoDB - How to remove a field from document (/mongodb/mongodb-how-to-remove-a-field-from-document/?utm_source=mkyong&utm_medium=author&utm_campaign=related-post&utm_content=6) |
| 3k | MongoDB - Update to upper case (/mongodb/mongodb-update-to-upper-case/?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

509k

SunCertPathBuilderException: unable to find valid certification path to requested target (/webservices/jax-ws/suncertpathbuilderexception-unable-to-find-valid-certification-path-to-requested-target/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=1)

635k

How to loop / iterate a List in Java (/java/how-do-loop-iterate-a-list-in-java/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=2)

616k

Java - Read a file from resources folder (/java/java-read-a-file-from-resources-folder/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=3)

1.2m

JAXB hello world example (/java/jaxb-hello-world-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=4)

528k

Android radio buttons example (/android/android-radio-buttons-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=5)

1.8m

Java Properties file examples (/java/java-properties-file-examples/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=6)

740k

Maven + Spring + Hibernate + MySql Example (/spring/maven-spring-hibernate-mysql-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=7)

1.4m

JAX-RS Tutorial (/tutorials/jax-rs-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=8)

801k

Java 8 forEach examples (/java8/java-8-foreach-examples/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=9)

966k

Maven Tutorial (/tutorials/maven-tutorials/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=10)

535k

How to loop a Map in Java (/java/how-to-loop-a-map-in-java/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=11)

615k

Java - Convert String to int (/java/java-convert-string-to-int/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=12)

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=13)

1.3m

How to write to file in Java - BufferedWriter (/java/how-to-write-to-file-in-java-bufferedwriter-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=14)

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=15)

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=16)

633k

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

584k

How to Set $JAVA_HOME environment variable on Mac OS X (/java/how-to-set-java_home-environment-variable-on-mac-os-x/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=18)

702k

Spring 3 MVC and JSON example (/spring-mvc/spring-3-mvc-and-json-example/?utm_source=mkyong&utm_medium=author&utm_campaign=top-pv&utm_content=19)

758k

RESTful Java client with java.net.URL (/webservices/jax-rs/restfull-java-client-with-java-net-url/?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

Grégory Hrycaj

Guest

Hello and thank you, what if we want to update two or more fields at once ?

👍 0 👎 ↩ REPLY

⏱ 8 months 5 days ago

Carol

Guest

Thank you, nice post!

👍 0 👎 ↩ REPLY

⏱ 1 year 6 months ago

Lyubka Genova

Guest

Thank you!

👍 0 👎 ↩ REPLY

⏱ 1 year 6 months ago

siddhant Gupta

Guest

Hi Mkyong, your posts are always great…

I think 3rd line of code in 2nd point “2. DBCollection.update() with $inc,”, i.e.

new BasicDBObject().append(“total clients”, 99));

should be updated to:

new BasicDBObject().append(“clients”, 99));

👍 0 👎 ↩ REPLY

⏱ 2 years 7 days ago

SD

Guest

Very simple and great post. thanks!

👍 0 👎 ↩ REPLY

⏱ 4 years 8 months ago

Nima

Guest

Thanks for the helpful post

👍 0 👎 ↩ REPLY

⏱ 4 years 9 months ago

Sri

Guest

Just love this post. Simple but covers everything.

👍 0 👎 ↩ REPLY

⏱ 4 years 9 months ago

Java + MongoDB hello world example (http://www.mkyong.com/mongodb/java-mongodb-hello-world-example/)

[…] Refer to this Java MongoDB update example. […]

👍 0 👎 ↩ REPLY

⏱ 4 years 9 months ago

Juan Toro Marty

Guest

I cant make it work.

```
boolean upsert = true;
                boolean multi = true;
                Mongo m = new Mongo(); // Conectamos a una instalacion en la maquina por
                                                                // default.
                DB bd = m.getDB("proyectos");
                DBCollection encargados = bd.getCollection("encargados");
                BasicDBObject doc = new BasicDBObject();
                doc.put("rut", this.rut);
                doc.put("nombre", this.nombre);
                doc.put("apellidos", this.apellidos);

                encargados.update(new BasicDBObject().append("rut", this.rut), doc, true, true);
```

It simple didnt update. Can you help me please. Thanks.

👍 0 👎 REPLY 5 years 2 months ago

Guest

Juan Toro Marty

I resolved it. The problem is that i need to set the multi flag to false. Multi only works with atomic updates ($)

👍 0 👎 REPLY 5 years 2 months ago

Guest

Sujeevan

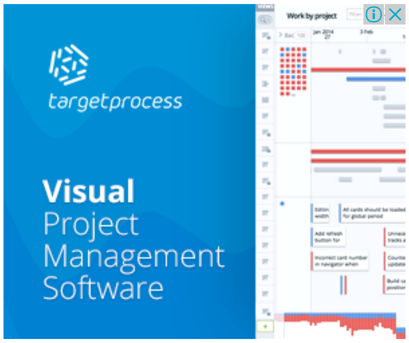
Hi,
What is the usage of 'total' in append("total clients", 99)? Is it a typo error?

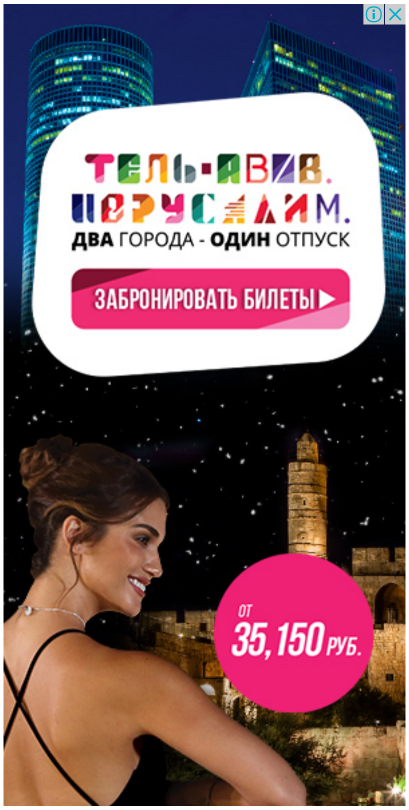
👍 0 👎 REPLY 5 years 7 months ago

Spring Data MongoDB : Update document (http://www.mkyong.com/mongodb/spring-data-mongodb-update-document/)

[…] Java MongoDB update example/ […]

👍 0 👎 REPLY 6 years 7 months ago




ТЕЛЬ-АВИВ.
ИЕРУСАЛИМ.
ДВА ГОРОДА - ОДИН ОТПУСК
ЗАБРОНИРОВАТЬ БИЛЕТЫ ▶
ОТ
35,150 РУБ.