



FREE eBook

LEARNING mongodb-java

Free unaffiliated eBook created from
Stack Overflow contributors.

#mongodb-
java

Table of Contents

About	1
Chapter 1: Getting started with mongodb-java	2
Remarks.....	2
Versions.....	2
Examples.....	2
Installation or Setup.....	3
Credits	6

About

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [mongodb-java](#)

It is an unofficial and free mongodb-java ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official mongodb-java.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapter 1: Getting started with mongodb-java

Remarks

This section provides an overview of what mongodb-java is, and why a developer might want to use it.

It should also mention any large subjects within mongodb-java, and link out to the related topics. Since the Documentation for mongodb-java is new, you may need to create initial versions of those related topics.

Versions

Version	ReleaseDate
3.4.2	2017-01-26
3.4.1	2016-12-19
3.4.0	2016-11-27
3.3.0	2016-07-18
3.2.2	2016-02-16
3.2.1	2016-01-13
3.2.0	2015-12-07
3.1.1	2015-11-12
3.1.0	2015-10-07
3.0.4	2014-09-14
2.14.3	2016-07-18
2.14.2	2016-03-31
2.14.1	2016-01-13
2.14.0	2015-12-07
2.13.3	2015-09-08

Examples

Installation or Setup

Prerequisite

A running MongoDB server, more details on that [here](#).

Setting Up a Java Project

The MongoDB Java Driver is supplied as a JAR file and can be included in a project just like any other JAR. For example:

- **Maven:** add an entry to the `<dependencies>` section of your `pom.xml`.

```
<dependency>
  <groupId>org.mongodb</groupId>
  <artifactId>mongo-java-driver</artifactId>
  <version>3.4.2</version>
</dependency>
```

- **Gradle:** add the following dependency to your `build.gradle`

```
compile 'org.mongodb:mongo-java-driver:3.4.2'
```

- **Manual Inclusion:** download the JAR file from [here](#) and add it to your project's classpath.

Note: the version shown in the examples above is the latest stable version at the time of writing but you can choose any version available [here](#).

Using the MongoDB Java Driver

Let's create a `MongoClient`, the simplest invocation is:

```
MongoClient mongoClient = new MongoClient(new ServerAddress("localhost", 27017));
```

This presumes that the MongoDB server is running on localhost at the default port and that it is unsecured. There are numerous variants of this for authenticated access, custom connection options, connecting to replicaset and sharded clusters all covered in detail in the [Mongo DB Java Driver docs](#).

Now that we're connected let's create a database and a collection. MongoDB implicitly creates a collection when the first document is saved into it so let's do that:

```
MongoDatabase database = mongoClient.getDatabase("starter");
MongoCollection<Document> collection = database.getCollection("superheroes");
Document document = new Document("name", "Superman");
collection.insertOne(document);
```

Note: you can [explicitly create a collection](#) but the above usage is more common and certainly fits better with a 'getting started' topic.

Now let's see if we can find the document we created:

```
FindIterable<Document> documents = purchases.find(Filters.eq("name", "Superman"));
for (Document found : documents) {
    System.out.println(found.toJson());
}
```

The above code will output the following (the `_id` field is MongoDB's auto generated document key):

```
Found document: { "_id" : { "$oid" : "5981bd586d47c203904a9cf9" }, "name" :
"Superman" }
```

Now that we've verified that we successfully created a document let's delete it:

```
DeleteResult deleteResult = purchases.deleteOne(Filters.eq("name", "Superman"));
System.out.println(String.format("Deleted %s document(s)", deleteResult.getDeletedCount()));
```

The above code will output the following:

```
Deleted 1 document(s)
```

Complete Example

Putting the various pieces together we get:

```
import com.google.common.collect.Lists;
import com.mongodb.MongoClient;
import com.mongodb.MongoCredential;
import com.mongodb.ServerAddress;
import com.mongodb.client.FindIterable;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.model.Filters;
import com.mongodb.client.result.DeleteResult;
import org.bson.Document;

import java.util.List;

public class IntroducingTheMongoDBJavaDriver {

    public static void main(String... args) throws Exception {
        MongoClient mongoClient = new MongoClient(new ServerAddress("localhost", 27017));

        MongoCollection<Document> superheroes =
mongoClient.getDatabase("starter").getCollection("superheroes");

        Document document = new Document("name", "Superman");
        superheroes.insertOne(document);

        FindIterable<Document> documents = superheroes.find(Filters.eq("name", "Superman"));
        for (Document found : documents) {
            System.out.println(String.format("Found document: %s", found.toJson()));
        }

        DeleteResult deleteResult = superheroes.deleteOne(Filters.eq("name", "Superman"));
        System.out.println(String.format("Deleted %s document(s)",
deleteResult.getDeletedCount()));
    }
}
```

```
    long count = superheroes.count();  
    System.out.println(String.format("Number of superheroes: %s", count));  
  }  
}
```

Read [Getting started with mongodb-java](https://riptutorial.com/mongodb-java/topic/9735/getting-started-with-mongodb-java) online: <https://riptutorial.com/mongodb-java/topic/9735/getting-started-with-mongodb-java>

Credits

S. No	Chapters	Contributors
1	Getting started with mongodb-java	Community , glytching , Menuka Ishan