**import** com.mongodb.BasicDBObject;

**import** com.mongodb.DB;

**import** com.mongodb.DBCollection;

**import** com.mongodb.Mongo;

**import** com.mongodb.WriteConcern;

**import** com.mongodb.gridfs.GridFS;

**import** com.mongodb.gridfs.GridFSInputFile;

**import** com.mongodb.gridfs.GridFSDBFile;

**public** **class** appletImage **extends** Applet {

Image img;

MediaTracker tr;

File temp;

**public** **void** init(){

**try** {

Mongo mongo = **new** Mongo("localhost", 27017);

DB db = mongo.getDB("ImageData");

DBCollection collection = db.getCollection("downloads\_meta");

File file = **new** File("/home/student/DSC01899.JPG");

//save the a file

GridFS gridfs = **new** GridFS(db, "downloads");

GridFSInputFile gfsFile = gridfs.createFile(file);

gfsFile.setFilename("destfile");

gfsFile.save();

//read the file

GridFSDBFile outFile = gridfs.findOne(gfsFile .getFilename());

System.*out*.println(outFile);

//write output to temp file

temp = File.*createTempFile*("delme", ".tmp");

System.*out*.println(temp.getPath());

outFile.writeTo(temp);

}

**catch** (IOException e) {

e.printStackTrace();

}

}

**public** **void** paint(Graphics g) {

tr = **new** MediaTracker(**this**);

img = getImage(getCodeBase(), temp.getPath());

tr.addImage(img,0);

g.drawImage(img,50,50, 100,100,**this**);

}

}

**Complete Example Code**

Below is the complete for building above short examples. Feel free to play with it.

|  |
| --- |
| package examples.mongodb.crud;    import java.io.File;  import java.io.IOException;    import com.mongodb.DB;  import com.mongodb.DBCursor;  import com.mongodb.MongoClient;  import com.mongodb.gridfs.GridFS;  import com.mongodb.gridfs.GridFSDBFile;  import com.mongodb.gridfs.GridFSInputFile;    public class MongoDBBinaryExample  {      public static void main(String[] args) throws IOException      {          MongoClient mongo = new MongoClient("localhost", 27017);          DB db = mongo.getDB("howtodoinjava");          //Save a image in DB          saveImageIntoMongoDB(db);          //Get a image from DB          getSingleImageExample(db);          //Get all images from DB          listAllImages(db);          saveToFileSystem(db);          //Delete images from DB          deleteImageFromMongoDB(db);            //Verifying if image was deleted or not          getSingleImageExample(db);      }        private static void saveImageIntoMongoDB(DB db) throws IOException {          String dbFileName = "DemoImage";          File imageFile = new File("c:\\DemoImage.png");          GridFS gfsPhoto = new GridFS(db, "photo");          GridFSInputFile gfsFile = gfsPhoto.createFile(imageFile);          gfsFile.setFilename(dbFileName);          gfsFile.save();      }        private static void getSingleImageExample(DB db) {          String newFileName = "c:/DemoImage";          GridFS gfsPhoto = new GridFS(db, "photo");          GridFSDBFile imageForOutput = gfsPhoto.findOne(newFileName);          System.out.println(imageForOutput);      }          private static void listAllImages(DB db) {          GridFS gfsPhoto = new GridFS(db, "photo");          DBCursor cursor = gfsPhoto.getFileList();          while (cursor.hasNext()) {              System.out.println(cursor.next());          }      }        private static void saveToFileSystem(DB db) throws IOException {          String dbFileName = "DemoImage";          GridFS gfsPhoto = new GridFS(db, "photo");          GridFSDBFile imageForOutput = gfsPhoto.findOne(dbFileName);          imageForOutput.writeTo("c:/DemoImageNew.png");      }        private static void deleteImageFromMongoDB(DB db) {          String dbFileName = "DemoImage";          GridFS gfsPhoto = new GridFS(db, "photo");          gfsPhoto.remove(gfsPhoto.findOne(dbFileName));      }  }      Output:    { "\_id" : { "$oid" : "53cff8d736414e8af4a4f0b8"} , "chunkSize" : 262144 , "length" : 138855 , "md5" : "b75f77c16c3ac6472365c06cde15d0da" , "filename" : "DemoImage" , "contentType" :  null  , "uploadDate" : { "$date" : "2014-07-23T18:03:03.403Z"} , "aliases" :  null }  { "\_id" : { "$oid" : "53cff8d736414e8af4a4f0b8"} , "chunkSize" : 262144 , "length" : 138855 , "md5" : "b75f77c16c3ac6472365c06cde15d0da" , "filename" : "DemoImage" , "contentType" :  null  , "uploadDate" : { "$date" : "2014-07-23T18:03:03.403Z"} , "aliases" :  null }    null |

**Full Example**

Full example to work with image, via **Java MongoDB GridFS API**. See comments for explanation.

package com.mkyong.core;

import java.io.File;

import java.io.IOException;

import java.net.UnknownHostException;

import com.mongodb.DB;

import com.mongodb.DBCollection;

import com.mongodb.DBCursor;

import com.mongodb.Mongo;

import com.mongodb.MongoException;

import com.mongodb.gridfs.GridFS;

import com.mongodb.gridfs.GridFSDBFile;

import com.mongodb.gridfs.GridFSInputFile;

/\*\*

\* Java MongoDB : Save image example

\*

\*/

public class SaveImageApp {

public static void main(String[] args) {

try {

Mongo mongo = new Mongo("localhost", 27017);

DB db = mongo.getDB("imagedb");

DBCollection collection = db.getCollection("dummyColl");

String newFileName = "mkyong-java-image";

File imageFile = new File("c:\\JavaWebHosting.png");

// create a "photo" namespace

GridFS gfsPhoto = new GridFS(db, "photo");

// get image file from local drive

GridFSInputFile gfsFile = gfsPhoto.createFile(imageFile);

// set a new filename for identify purpose

gfsFile.setFilename(newFileName);

// save the image file into mongoDB

gfsFile.save();

// print the result

DBCursor cursor = gfsPhoto.getFileList();

while (cursor.hasNext()) {

System.out.println(cursor.next());

}

// get image file by it's filename

GridFSDBFile imageForOutput = gfsPhoto.findOne(newFileName);

// save it into a new image file

imageForOutput.writeTo("c:\\JavaWebHostingNew.png");

// remove the image file from mongoDB

gfsPhoto.remove(gfsPhoto.findOne(newFileName));

System.out.println("Done");

} catch (UnknownHostException e) {

e.printStackTrace();

} catch (MongoException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

\

package com.technodyne.core;

import java.io.File;

import java.io.IOException;

import java.net.UnknownHostException;

import com.mongodb.DB;

import com.mongodb.DBCollection;

import com.mongodb.DBCursor;

import com.mongodb.Mongo;

import com.mongodb.MongoException;

import com.mongodb.gridfs.GridFS;

import com.mongodb.gridfs.GridFSDBFile;

import com.mongodb.gridfs.GridFSInputFile;

public class SaveImage {

public static void main(String[]args){

try {

Mongo mongo = new Mongo("127.0.0.1",27017);

DB db = mongo.getDB("imagedb");

DBCollection collection = db.getCollection("dummyImageCollection");

String newFileName = "technodyne-java-image";

File imageFile = new File("C:UsersSUPUNDesktopimages.jpg");

GridFS gfsPhoto = new GridFS(db,"photo");

try {

GridFSInputFile gfsFile = gfsPhoto.createFile(imageFile);

gfsFile.setFilename(newFileName);

gfsFile.save();

DBCursor cursor = gfsPhoto.getFileList();

while(cursor.hasNext()){

System.out.println(cursor.next());

}

GridFSDBFile imageForOutput = gfsPhoto.findOne(newFileName);

imageForOutput.writeTo("C:UsersSUPUNDesktopjava-monngo-image.jpg");

gfsPhoto.remove(gfsPhoto.findOne(newFileName));

System.out.println("Done");

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

} catch (UnknownHostException e) {

// TODO: handle exception

e.printStackTrace();

}

catch(MongoException e){

e.printStackTrace();

}

}

}

import pymongo

import gridfs

if \_\_name\_\_ == '\_\_main\_\_':

# read in the image.

    filename = "path to input file"

    datafile = open(filename,"r");

    thedata = datafile.read()

# connect to database

    connection = pymongo.Connection("localhost",27017);

    database = connection['example']

# create a new gridfs object.

    fs = gridfs.GridFS(database)

# store the data in the database. Returns the id of the file in gridFS

    stored = fs.put(thedata, filename="testimage")

# retrieve what was just stored.

    outputdata =fs.get(stored).read()

# create an output file and store the image in the output file

    outfilename = "path to output file"

    output= open(outfilename,"w")

    output.write(outputdata)

# close the output file

    output.close()

# for experimental code restore to known state and close connection

    fs.delete(stored)

    connection.drop\_database('example');

#    print(connection.database\_names())

    connection.close()