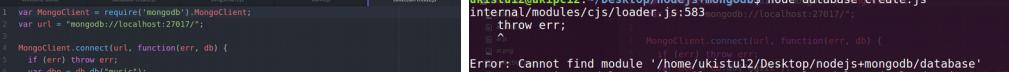


Create a Database called music.

Create a collection called songdetails.



The screenshot shows the Atom code editor with a file named `collection.create.js` open. The code creates a MongoDB client, connects to the local host at port 27017, and creates a collection named `songdetails`. The terminal window shows the command `node database.create.js` being run, which outputs an error message about the module not being found, followed by a deprecation warning from Node.js version 5.2.0 regarding the Server Discovery and Monitoring engine.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  dbo.createCollection("songdetails", function(err, res) {
    if (err) throw err;
    console.log("Collection created!");
    db.close();
  });
});
```

```
ukistul2@ukipc12:~/Desktop/nodejs+mongodb$ node database.create.js
Error: Cannot find module '/home/ukistul2/Desktop/nodejs+mongodb/database'
  at Function.Module._resolveFilename (internal/modules/cjs/loader.js:581:15)
  at Function.Module._load (internal/modules/cjs/loader.js:507:29)
  at Function.Module.require (internal/modules/cjs/loader.js:742:12)
  at startup (internal/bootstrap/node.js:279:19)
  at bootstrapNodeJSCore (internal/bootstrap/node.js:696:3)
(node:5620) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useNewUrlParser: true } to the MongoClient constructor.
Database created!
ukistul2@ukipc12:~/Desktop/nodejs+mongodb$ node collectioncreate.js
(node:5755) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discovery and Monitoring engine, pass option { useNewUrlParser: true } to the MongoClient constructor.
Collection created!
```

Create the above 5 song documents.

The screenshot shows a dual-monitor setup. The left monitor displays the Atom code editor with a file named 'songinsert.js' open. The right monitor displays a terminal window with the command 'nodejs songinsert.js' running, outputting error messages related to MongoDB connection issues.

```
Atom File Edit View Selection Find Packages Help
File Edit View Selection Find Packages Help
songinsert.js --Desktop/nodejs+mongodb - Atom
Project Welcome Guide songinsert.js database.create.js singername.js
mongodbs
  A.R.and
  A.R.song.js
  A.Rang
  A.Rang
  collection.create.js
  database.create.js
  delete.js
  delete.Ang
  insert.js
  singername.js
  singername.png
  singername.png
  singername.js
  singername.js

1 var MongoClient = require('mongodb').MongoClient;
2 var url = "mongodb://localhost:27017/";
3
4 MongoClient.connect(url, function(err, db) {
5   if (err) throw err;
6   var music = [
7     {
8       SongName: "Thaniye thanthan thanhiye", film: "Rytham", musicDirector: "A.R.Rahman", singer: "Sivaraj"
9       SongName: "Evano oruvan", film: "Alaiaputhney", musicDirector: "A.R.Rahman", singer: "Swarna"
10      SongName: "Roja poonthadam", film: "Kannukulil nilavu", musicDirector: "Ilaiyaraaja", singer: "Hariharan"
11      SongName: "Vennilave Vennilave", film: "Minisarakanavu", musicDirector: "A.R.Rahman", singer: "Hariharan"
12      SongName: "Sollamathoddu sellum kannal", film: "Dheena", musicDirector: "Yuvan Shankar Raja"
13    }
14  ];
15  dbo.collection("songdetails").insertMany(music, function(err, res) {
16    if (err) throw err;
17    console.log("Number of documents inserted: " + res.insertedCount);
18    db.close();
19  });
20 });
21
```

```
ustikul12@ukip12:~/Desktop/nodejs+mongodb$ node songinsert.js
throw err;
^
Error: cannot find module '/home/ustikul12/Desktop/nodejs+mongodb/database'
  at Function.Module._resolveFilename (internal/modules/cjs/loader.js:581:15)
  at Function.Module._load (internal/modules/cjs/loader.js:507:25)
  at Function.Module.runMain (internal/modules/cjs/loader.js:742:12)
  at startup (internal/bootstrap/node.js:279:19)
  at bootstrapNodeJSCore (internal/bootstrap/node.js:696:3)
ustikul12@ukip12:~/Desktop/nodejs+mongodb$ node dbcreate.js
(node:1255) DeprecationWarning: current Service Discovery and Monitoring engine is deprecated, an
d will be removed in a future version. To use the new Server Discover and Monitoring engine, pas
s option { useNewUrlParser: true } to the MongoClient constructor.
Collection created!
ustikul12@ukip12:~/Desktop/nodejs+mongodb$ node collection.create.js
(node:1255) DeprecationWarning: current Service Discovery and Monitoring engine is deprecated, an
d will be removed in a future version. To use the new Server Discover and Monitoring engine, pas
s option { useNewUrlParser: true } to the MongoClient constructor.
Collection created!
ustikul12@ukip12:~/Desktop/nodejs+mongodb$ node singername.js
(node:1255) DeprecationWarning: current Service Discovery and Monitoring engine is deprecated, an
d will be removed in a future version. To use the new Server Discover and Monitoring engine, pas
s option { useNewUrlParser: true } to the MongoClient constructor.
Number of documents inserted: 5
ustikul12@ukip12:~/Desktop/nodejs+mongodb$
```

List all documents created.

```
ukistu12@ukipc12:~/Desktop/nodejs/mongodb$ mongo
MongoDB shell version: 3.2.10
connecting to: test
> show dbs
admin 0.000GB
local 0.000GB
student 0.000GB
tutorialtautos 0.000GB
uk1 0.000GB
uk14 0.000GB
uk16 0.000GB
> use music
switched to db music
> db.songsdetails.find()
[{"_id": ObjectId("5e2a6fB46ecabfc1b3ec6137c"), "SongName": "Usuru narampula", "film": "iruthi hchuttu", "musicDirector": "A.R.Rahman", "singer": "Syreya"}, {"_id": ObjectId("5e1446d5682920bb075a6d"), "SongName": "Usuru narampula", "film": "iruthi hchuttu", "musicDirector": "A.R.Rahman", "singer": "Syreya"}, {"_id": ObjectId("5e14a8e4d75c17e15e13ca"), "SongName": "Thaniy thananth thaniye", "film": "Rytham", "musicDirector": "A.R.Rahman", "singer": "Sankaradevan"}, {"_id": ObjectId("5e14a8e4d75c17e15e13cb"), "SongName": "Evano oruvan", "film": "Alaiapauthey", "musicDirector": "A.R.Rahman", "singer": "Swarnalatha"}, {"_id": ObjectId("5e14a8e4d75c17e15e13cc"), "SongName": "Roja poonthodam", "film": "Kannukkul nilavu", "musicDirector": "Alaiyaraaja", "singer": "Unnikrishnan,Anurathasriram"}, {"_id": ObjectId("5e14a8e4d75c17e15e13cd"), "SongName": "Kannithiavu", "film": "Minsarakonavayal", "musicDirector": "A.R.Rahman", "singer": "Harkaran,Sadhansarkam"}, {"_id": ObjectId("5e14a8e4d75c17e15e13ce"), "SongName": "Soltamalithoddellum kannikal", "film": "Dheena", "musicDirector": "Yuvan Sarathraja", "singer": "Harikaran"}]
```

List A.R.Rahman's songs.

```
akshit@akshit-OptiPlex-5090:~/Desktop/nodejs+mongodb$ node create.js
{
  "_id": "5e14a665629208b0b75a6d",
  "SongName": "Kehne Pele",
  "musicDirector": "A. R. Rahman",
  "singer": "Arijit Singh"
}
akshit@akshit-OptiPlex-5090:~/Desktop/nodejs+mongodb$ node collection.create.js
{
  "_id": "5e14a665629208b0b75a6d",
  "SongName": "Kehne Pele",
  "musicDirector": "A. R. Rahman",
  "singer": "Arijit Singh"
}
akshit@akshit-OptiPlex-5090:~/Desktop/nodejs+mongodb$ node database.create.js
{
  "_id": "5e14a888ea475171e51c3aa",
  "SongName": "Eagle Wings",
  "file": "EagleWings.mp3",
  "musicDirector": "A. R. Rahman",
  "singer": "Arijit Singh"
}
akshit@akshit-OptiPlex-5090:~/Desktop/nodejs+mongodb$ node delete.js
{
  "_id": "5e14a888ea475171e51c3ab",
  "SongName": "Eagle Wings",
  "file": "EagleWings.mp3",
  "musicDirector": "A. R. Rahman",
  "singer": "Arijit Singh"
}
akshit@akshit-OptiPlex-5090:~/Desktop/nodejs+mongodb$ ls
delete.png
```

Delete the song which you don't like.

The terminal window shows the command `nodejs>mongod --atlas` running. The code editor displays a file named `delete.js` which contains a script to delete documents from a MongoDB database. The code uses the `Mongoose` library to connect to the database and delete a document where the songname is "Roja poonthoddam". The terminal also lists several other files and images in the project directory.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var myquery = { Songname: "Roja poonthoddam" };
  dbo.collection("songdetails").deleteOne(myquery, function(err, obj) {
    if (err) throw err;
    console.log("1 document deleted");
    db.close();
  });
});
});
```

Add new song which is your favourite.



The screenshot shows the Atom code editor with the following file structure:

- Project: nodejs-mongodb
- File Explorer:
 - nodejs-mongodb
 - AKR.png
 - ar song1.png
 - atd1.png
 - collection create.js
 - database create.js
 - delele.js
 - delele.png
 - insert.js
 - insert1.js
 - singers.png
 - singers1.png
 - singersame.js
 - song.png
 - songame.js

The `insert.js` file contains the following code:

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var myobj = { SongName:"Usuru narampula", film:"iruvarichchutu", musicDirector:"A.R.Rahman" };
  db.collection("songdetails").insertOne(myobj), function(err, res) {
    if (err) throw err;
    console.log("1 document inserted");
    db.close();
  });
});
```

```
ukistu12@ukipci12:~/Desktop/nodejs$ mongod
ukistu12@ukipci12:~/Desktop/nodejs$ mongod
[...]
at Function._load (internal/modules/cjs/loader.js:507:25)
at Function.Module._compile (internal/modules/cjs/loader.js:742:12)
at Module._extensions..js (internal/modules/cjs/loader.js:789:19)
at Object.require.extensions.(anonymous function) [as .js] (internal/bootstrap/node_modules/mongodb/lib/_bootstrap/bootstrap_node.js:696:3)
ukistu12@ukipci12:~/Desktop/nodejs$ mongod --node db:create.js
(node:5620) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, an
d will be removed in a future version. To use the new Server Discover and Monitoring engine, pas
s option { useUnifiedTopology: true } to the MongoClient constructor.
Database created!
ukistu12@ukipci12:~/Desktop/nodejs$ mongod --node collectionCreate.js
(node:5755) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, an
d will be removed in a future version. To use the new Server Discover and Monitoring engine, pas
s option { useUnifiedTopology: true } to the MongoClient constructor.
Collection created!
ukistu12@ukipci12:~/Desktop/nodejs$ mongodbs
(node:5867) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, an
d will be removed in a future version. To use the new Server Discover and Monitoring engine, pas
s option { useUnifiedTopology: true } to the MongoClient constructor.
MongoDB shell version: 4.2.0
ukistu12@ukipci12:~/Desktop/nodejs$ mongodbs node songinsert.js
(node:6276) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, an
d will be removed in a future version. To use the new Server Discover and Monitoring engine, pas
s option { useUnifiedTopology: true } to the MongoClient constructor.
1 document inserted
ukistu12@ukipci12:~/Desktop/nodejs$ mongodbs
```

List Songs sung by Hariharan from Minsara kanavu film.



The screenshot shows the Atom code editor with a file named `main.js` open. The code uses the `MongoClient` module to connect to a local MongoDB instance at port 27017. It then finds documents in the `songsdetails` collection where the genre is "Minsarakavu". The results are printed to the console.

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  var query = { $in: ["Minsarakavu"] };
  dbo.collection("songsdetails").find(query).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

10) List out the singers' names in your document.



The screenshot shows the Atom code editor interface. The title bar displays "Atom" and the current file path as "singersname.js — ~/Desktop/nodejs/mongodb — Atom". The left sidebar lists files in the project: "singersname.js" (highlighted), "database.create.js", "collection.create.js", "insert.js", "delete.js", "update.png", "A.R.png", "song.json", "ar.json", "ar.js", and "singers.png". The main editor area contains the following JavaScript code:

```
var MongoClient = require('mongodb').MongoClient;
var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {
  if (err) throw err;
  var dbo = db.db("music");
  dbo.collection("songdetails").find({}, { projection: { _id: 0, singer: 1 } }).toArray(function(err, result) {
    if (err) throw err;
    console.log(result);
    db.close();
  });
});
```

```
Terminal File Edit View Search Terminal Tabs Help
ukistul2@ukipc12:~/Desktop/nodejs+mongodb$ mongod
ukistul2@ukipc12:~/Desktop/nodejs+mongodb$ node singersname.js
{
  singer: "Sankaradevayan",
  Client: require('mongodb').MongoClient;
  { _id: 5e314a88ea475c171e51c3ab, mongoDB://localhost:27017/ },
  SongName: "Evano oruvan",
  film: "Alainauauthey",
  musicDirector: "A.R.Rahman"
} throw err;
singer: "Swarnalatha",
  db.db("music");
{ _id: 5e314a88ea475c171e51c3ad, mongoDB://localhost:27017/ },
  projection: { _id: 0, singer: 1 } }).toArray(function(err, result) {
SongName: "Vennilave Vennilave",
  throw err;
film: "Minsarakavu",
  console.log(result);
  musicDirector: "A.R.Rahman"
} ) {
  singer: "Harkaran,Sadhanasarkkam" }
}
ukistul2@ukipc12:~/Desktop/nodejs+mongodb$ node singersname.js
(node:4834) DeprecationWarning: current Server Discovery and Monitoring engine is deprecated, and will be removed in a future version. To use the new Server Discover and Monitoring engine, pass option { useUnifiedTopology: true } to the MongoClient constructor.
[ { singer: 'Syreya' },
  { singer: 'Syreya' },
  { singer: 'Sankaradevayan' },
  { singer: 'Swarnalatha' },
  { singer: 'Unnikirishnan,Anurathasirim' },
  { singer: 'Harkaran,Sadhanasarkkam' },
  { singer: 'Harikaran' } ]
ukistul2@ukipc12:~/Desktop/nodejs+mongodb$
```