

Using Mongodb

In Node, Java, and Spring

Using Mongodb in Node

Using mongodb javascript driver

Official mongodb driver

<https://www.npmjs.com/package/mongodb>

<http://mongodb.github.io/node-mongodb-native/>

In this section we cover examples in:

https://www.w3schools.com/nodejs/nodejs_mongodb.asp

Create npm project and install mongodb driver package

```
$ npm init
```

```
$ npm install mongodb
```

Also install babel-preset-es2015 so that we can write ES6 code

```
$ npm install babel-preset-es2015 --save-dev
```

NOTE: Before running mongodb examples make sure mongodb server is running

Connect to mongodb

Steps to connect:

1. Import mongodb driver's MongoClient Class
2. Define database URL
3. Call MongoClient.connect() function and pass it DB URL and callback method
4. Once connected close database connection

```
import MongoClient from "mongodb";
```

```
const URL = "mongodb://localhost:27017/mydb";
```

```

MongoClient.connect(URL, (error, db) => {
  if (error) throw error;
  console.log(`Connected to ${URL}`);
  db.close();
});

```

Running code

Running code is same as running any other node ES6 application

E.g. if your file name is **app01.js** then give this command
 \$ babel-node --presets es2015 app01.js

NOTE: make sure **babel-cli** global utility is installed
 \$ npm install babel-cli -g

Insert

```

import MongoClient from "mongodb";

const URL = "mongodb://localhost:27017/mydb";

// Insert
MongoClient.connect(URL, (error, db) => {
  if (error) throw error;
  let myPerson = {
    name: "Sheraz",
    salary: 200
  };
  db.collection("person").insert(myPerson, (error, result) => {
    if (error) throw error;
    console.log("inserted record", result);
    db.close();
  });
});

```

Find

```

import MongoClient from "mongodb";

const URL = "mongodb://localhost:27017/mydb";

```

```
// Find
MongoClient.connect(URL, (error, db) => {
  if (error) throw error;
  let query = {name: "Sheraz", salary: {$gt: 50}};
  let cursor = db.collection("person").find(query);

  cursor.toArray((error, resultArray) => {
    if (error) throw error;
    console.log(resultArray);
    cursor.close();
    db.close();
  });
});
```

Update

```
import MongoClient from "mongodb";

const URL = "mongodb://localhost:27017/mydb";

// Update
MongoClient.connect(URL, (error, db) => {
  if (error) throw error;
  let query = {name: "Sheraz"};
  let newPerson = {$set: {salary: 300}}
  db.collection("person").updateMany(query, newPerson, (error,
result) => {
    if (error) throw error;
    console.log("Updated records", result.result);
    db.close();
  });
});
```

Delete

```
import MongoClient from "mongodb";

const URL = "mongodb://localhost:27017/mydb";

// Delete
MongoClient.connect(URL, (error, db) => {
  if (error) throw error;
  let query = {name: "Sheraz"};
  db.collection("person").deleteMany(query, (error, result) => {
    if (error) throw error;
    console.log("Deleted records", result.result);
  });
});
```

```
        db.close();
    });
});
```

Drop Collection

```
import MongoClient from "mongodb";

const URL = "mongodb://localhost:27017/mydb";

// Drop Collection
MongoClient.connect(URL, (error, db) => {
    if (error) throw error;
    db.collection("person").drop((error, result) => {
        if (error) throw error;
        console.log("Dropped Collection", result);
        db.close();
    });
});
```

Using Mongodb in Node

Using Monk

<https://automattic.github.io/monk/>

<https://github.com/Automattic/monk>

<https://www.npmjs.com/package/monk>

Monk api is written over mongodb driver.

It makes mongodb interactions easier and reduces code.

The whole monk api is uses ES6's promises

Install Monk

```
$ npm install monk
```

Connect to mongodb

Steps to connect:

1. Import monk function
2. Call monk function to create mongo connection. Returned connection is a promise
3. Handle promise's success, failure and exception conditions
4. Close connection once all database operations are complete

NOTE: success data is the same connection that is returned by

```
import monk from "monk";
```

```
let db = monk("localhost:27017/mydb");
```

```
db.then((successDb) => {  
  console.log(`  
    Successfully connected  
    Both db are reference to same.  
    (db === successDb) = ${db === successDb}`);  
  successDb.close();  
}, (error) => {  
  console.log("Error Occured");  
}).catch((exception) => {  
  console.log("Failed to connect", exception);  
});
```

Insert

```
import monk from "monk";
```

```
let db = monk("localhost:27017/mydb");  
let personCollection = db.get("person");
```

```
const personObject = {  
  name: "Sheraz",  
  salary: 250  
};
```

```
let insertPromise = personCollection.insert(personObject);
```

```
insertPromise.then((insertedDocument) => {  
  console.log("Inserted document", insertedDocument);  
}).catch((error) => {  
  console.error("Failed to insert", error);  
}).then(() => db.close());
```

Find

```
import monk from "monk";
```

```
let db = monk("localhost:27017/mydb");  
let personCollection = db.get("person");
```

```
let query = {salary: {$lt: 1000}};

let findPromise = personCollection.find(query);

findPromise.then((foundDocuments) => {
  console.log("Found Documents", foundDocuments);
}).catch((error) => {
  console.error("Failed", error);
}).then(() => db.close());
```

Update

```
import monk from "monk";

let db = monk("localhost:27017/mydb");
let personCollection = db.get("person");

const newObject = {$set: {salary: 500}};
const query = {name: "Sheraz"}

let updatePromise = personCollection.update(query, newObject, {multi: true});

updatePromise.then((updateResult) => {
  console.log("Updated Documents", updateResult);
}).catch((error) => {
  console.error("Failed", error);
}).then(() => db.close());
```

Delete

```
import monk from "monk";

let db = monk("localhost:27017/mydb");
let personCollection = db.get("person");

const query = {name: "Sheraz"}

let deletePromise = personCollection.remove(query);

deletePromise.then((deleteResult) => {
  console.log("Deleted Documents", deleteResult.result);
}).catch((error) => {
  console.error("Failed", error);
}).then(() => db.close());
```


