

# Node.js



## Set up a simple application using Node.js and MongoDB

- First, create a directory where your application will live.

```
mkdir myproject
cd myproject
```

- Create the initial structure for your new project

```
npm init -y
```

- Next, install the driver dependency.

```
npm install mongodb --save
```

## Connect to MongoDB

Connect using the MongoClient to a running mongod instance by specifying the MongoDB uri.

```
const { MongoClient } = require("mongodb");
const dbName = 'test'; const uri = 'mongodb://Alice:Moon1234@127.0.0.1:27017/' + dbName ;
const client = new MongoClient(uri);
async function run() {
  try {
    const database = client.db(dbName);
    const movies = database.collection("movies");
    // Query for a movie that has the title 'The Room'
    const query = { title: "Back to the Future" };
    const options = {
      // sort matched documents in descending order by rating
      sort: { "imdb.rating": -1 },
      // Include only the `title` and `imdb` fields in the returned document
      projection: { _id: 0, title: 1, imdb: 1 },
    };
    const movie = await movies.findOne(query, options);
    // since this method returns the matched document, not a cursor, print it directly
    console.log(movie);
  } finally { await client.close(); } }
run().catch(console.dir);
```

## Insert a Document

```
const { MongoClient } = require("mongodb");
const dbName = 'test';
const uri = 'mongodb://Alice:Moon1234@127.0.0.1:27017/' + dbName ;

const client = new MongoClient(uri);
async function run() {
  try {
    const database = client.db(dbName);
    const movies = database.collection('movies');

    const doc = {
      title: "Back to the Future",
      year: "1985",
    };

    const result_insert = await movies.insertOne(doc);
    console.log(result_insert);

  } finally {
    // Ensures that the client will close when you finish/error
    await client.close(); } }
run().catch(console.dir);
```

# Find All Documents

```
...
async function run() {
  try {
    const database = client.db(dbName);
    const movies = database.collection("movies");
    // query for movies that have a runtime less than 15 minutes
    //const query = { runtime: { $lt: 15 } };
    const query = { };
    const options = {
      // sort returned documents in ascending order by title (A->Z)
      sort: { title: 1 },
      // Include only the `title` and `imdb` fields in each returned document
      projection: { _id: 0, title: 1, imdb: 1 },
    };
    const cursor = movies.find(query, options);
    // print a message if no documents were found
    if ((await movies.countDocuments(query)) === 0) {
      console.log("No documents found!");
    }
    // replace console.dir with your callback to access individual elements
    await cursor.forEach(console.dir);
  } finally { await client.close(); }
}
run().catch(console.dir);
```

Taller Web

33

# MongoDB (old version)

- Next slides are from a previous mongodb database and driver.
- Asynchronous code based on callbacks
- See [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language\\_Overview#asynchronous\\_programming](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_Overview#asynchronous_programming) for an overview of Asynchronous programming

Taller Web

34

# Connect to MongoDB (old version)

## Declare MongoClient variable and other variables

```
const MongoClient = require('mongodb').MongoClient;
const assert = require('assert');
```

Connect using the MongoClient to a running mongod instance by specifying the MongoDB uri.

```
// Connection URL
const dbName = 'test';
const url = 'mongodb://Alice:Moon1234@localhost:27017/' + dbName ;

// Database Name

// Create a new MongoClient
const client = new MongoClient(url);

// Use connect method to connect to the Server
client.connect(function(err) {
  assert.equal(null, err);
  console.log("Connected successfully to server");

  const db = client.db(dbName);

  client.close(); });
```

Ta

# Insert a Document (old version)

```
const insertDocuments = function(db, callback) {
  // Get the documents collection
  const collection = db.collection('documents');
  // Insert some documents
  collection.insertMany([
    {a : 1}, {a : 2}, {a : 3}
  ], function(err, result) {
    assert.equal(err, null);
    assert.equal(3, result.result.n);
    assert.equal(3, result.ops.length);
    console.log("Inserted 3 documents into the collection");
    callback(result);
  });
}

// Use connect method to connect to the server
client.connect(function(err) {
  assert.equal(null, err);
  console.log("Connected successfully to server");

  const db = client.db(dbName);

  insertDocuments(db, function() {
    client.close();
  });
});
```

Taller Web

36

# Find All Documents (old version)

```
const findDocuments = function(db, callback) {
  // Get the documents collection
  const collection = db.collection('documents');
  // Find some documents
  collection.find({}).toArray(function(err, docs) {
    assert.equal(err, null);
    console.log("Found the following records");
    console.log(docs);
    callback(docs);
  });
}

// Use connect method to connect to the server
client.connect(function(err) {
  assert.equal(null, err);
  console.log("Connected correctly to server");

  const db = client.db(dbName);

  insertDocuments(db, function() {
    findDocuments(db, function() {
      client.close();
    });
  });
});
```

Taller web

37

# MongoDB NodeJS

- <https://www.mongodb.com/docs/drivers/node/current/quick-start/connect-to-mongodb/> (current v 5.2)
- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language\\_Overview](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_Overview)
- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language\\_Overview#asynchronous\\_programming](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_Overview#asynchronous_programming)
- Tarea: Ejercitar operaciones CRUD desde Node.js en el documento propio

Taller Web

38