

mongodb

TS

6.2.0

•

Public

•

Published 14 days ago

Readme

Code

Beta

3 Dependencies

12,285 Dependents

427 Versions

MongoDB Node.js Driver

The official **MongoDB** driver for Node.js.

Upgrading to version 6? Take a look at our **upgrade guide here!**

Quick Links

Site	Link
Documentation	<a href="http://www.mongodb.com/docs/drivers/node">www.mongodb.com/docs/drivers/node</a>
API Docs	<a href="http://mongodb.github.io/node-mongodb-native">mongodb.github.io/node-mongodb-native</a>
npm package	<a href="http://www.npmjs.com/package/mongodb">www.npmjs.com/package/mongodb</a>
MongoDB	<a href="http://www.mongodb.com">www.mongodb.com</a>
MongoDB University	<a href="http://learn.mongodb.com">learn.mongodb.com</a>
MongoDB Developer Center	<a href="http://www.mongodb.com/developer">www.mongodb.com/developer</a>
Stack Overflow	<a href="http://stackoverflow.com">stackoverflow.com</a>
Source Code	<a href="http://github.com/mongodb/node-mongodb-native">github.com/mongodb/node-mongodb-native</a>
Upgrade to v6	<a href="#">etc/notes/CHANGES_6.0.0.md</a>
Contributing	<a href="#">CONTRIBUTING.md</a>
Changelog	<a href="#">HISTORY.md</a>

Bugs / Feature Requests

Think you’ve found a bug? Want to see a new feature in node-mongodb-native ? Please open a case in our issue management tool, JIRA:

- Create an account and login [jira.mongodb.org](http://jira.mongodb.org).
- Navigate to the NODE project [jira.mongodb.org/browse/NODE](http://jira.mongodb.org/browse/NODE).
- Click **Create Issue** - Please provide as much information as possible about the issue type and how to reproduce it.

Bug reports in JIRA for all driver projects (i.e. NODE, PYTHON, CSHARP, JAVA) and the Core Server (i.e. SERVER) project are **public**.

Support / Feedback

For issues with, questions about, or feedback for the Node.js driver, please look into our [support channels](#). Please do not email any of the driver developers directly with issues or questions - you're more likely to get an answer on the [MongoDB Community Forums](#).

### Change Log

Change history can be found in [HISTORY.md](#) .

### Compatibility

For server and runtime version compatibility matrices, please refer to the following links:

- [MongoDB](#)
- [NodeJS](#)

### Component Support Matrix

The following table describes add-on component version compatibility for the Node.js driver. Only packages with versions in these supported ranges are stable when used in combination.

Component	mongodb@3.x	mongodb@4.x	mongodb@5.x	mongodb@6.x
<a href="#">bson</a>	^1.0.0	^4.0.0	^5.0.0	^6.0.0
<a href="#">bson-ext</a>	^1.0.0    ^2.0.0	^4.0.0	N/A	N/A
<a href="#">kerberos</a>	^1.0.0	^1.0.0    ^2.0.0	^1.0.0    ^2.0.0	^2.0.1
<a href="#">mongodb-client-encryption</a>	^1.0.0	^1.0.0    ^2.0.0	^2.3.0	^6.0.0
<a href="#">mongodb-legacy</a>	N/A	^4.0.0	^5.0.0	^6.0.0
<a href="#">@mongodb-js/zstd</a>	N/A	^1.0.0	^1.0.0	^1.1.0

### Typescript Version

We recommend using the latest version of typescript, however we currently ensure the driver's public types compile against `typescript@4.1.6` . This is the lowest typescript version guaranteed to work with our driver: older versions may or may not work - use at your own risk. Since typescript **does not restrict breaking changes to major versions** we consider this support best effort. If you run into any unexpected compiler failures against our supported TypeScript versions please let us know by filing an issue on our [JIRA](#).

## Installation

The recommended way to get started using the Node.js 5.x driver is by using the `npm` (Node Package Manager) to install the dependency in your project.

After you've created your own project using `npm init` , you can run:

```
npm install mongodb
# or ...
yarn add mongodb
```

This will download the MongoDB driver and add a dependency entry in your `package.json` file.

If you are a Typescript user, you will need the Node.js type definitions to use the driver's definitions:

```
npm install -D @types/node
```

## Driver Extensions

The MongoDB driver can optionally be enhanced by the following feature packages:

Maintained by MongoDB:

- Zstd network compression - [@mongodb-js/zstd](#)
- MongoDB field level and queryable encryption - [mongodb-client-encryption](#)

- GSSAPI / SSPI / Kerberos authentication - [kerberos](#)

Some of these packages include native C++ extensions. Consult the [trouble shooting guide here](#) if you run into compilation issues.

Third party:

- Snappy network compression - [snappy](#)
- AWS authentication - [@aws-sdk/credential-providers](#)

## Quick Start

This guide will show you how to set up a simple application using Node.js and MongoDB. Its scope is only how to set up the driver and perform the simple CRUD operations. For more in-depth coverage, see the [official documentation](#).

### Create the `package.json` file

First, create a directory where your application will live.

```
mkdir myProject
cd myProject
```

Enter the following command and answer the questions to create the initial structure for your new project:

```
npm init -y
```

Next, install the driver as a dependency.

```
npm install mongodb
```

### Start a MongoDB Server

For complete MongoDB installation instructions, see [the manual](#).

1. Download the right MongoDB version from [MongoDB](#)
2. Create a database directory (in this case under `/data`).
3. Install and start a `mongod` process.

```
mongod --dbpath=/data
```

You should see the **mongod** process start up and print some status information.

### Connect to MongoDB

Create a new **app.js** file and add the following code to try out some basic CRUD operations using the MongoDB driver.

Add code to connect to the server and the database **myProject**:

**NOTE:** Resolving DNS Connection issues

Node.js 18 changed the default DNS resolution ordering from always prioritizing ipv4 to the ordering returned by the DNS provider. In some environments, this can result in `localhost` resolving to an ipv6 address instead of ipv4 and a consequent failure to connect to the server.

This can be resolved by:

- specifying the ip address family using the MongoClient `family` option (`MongoClient(<uri>, { family: 4 } )`)
- launching mongod or mongos with the ipv6 flag enabled ([--ipv6 mongod option documentation](#))
- using a host of `127.0.0.1` in place of localhost
- specifying the DNS resolution ordering with the `--dns-resolution-order` Node.js command line argument (e.g. `node --dns-resolution-order=ipv4first`)

```
const { MongoClient } = require('mongodb');

// or as an es module:
// import { MongoClient } from 'mongodb'

// Connection URL
const url = 'mongodb://localhost:27017';
const client = new MongoClient(url);

// Database Name
const dbName = 'myProject';

async function main() {
  // Use connect method to connect to the server
  await client.connect();
  console.log('Connected successfully to server');
  const db = client.db(dbName);
  const collection = db.collection('documents');

  // the following code examples can be pasted here...

  return 'done.';
}

main()
  .then(console.log)
  .catch(console.error)
  .finally(() => client.close());
```

Run your app from the command line with:

```
node app.js
```

The application should print **Connected successfully to server** to the console.

### Insert a Document

Add to **app.js** the following function which uses the **insertMany** method to add three documents to the **documents** collection.

```
const insertResult = await collection.insertMany([ { a: 1 }, { a: 2 }, { a: 3 } ]);
console.log('Inserted documents =>', insertResult);
```

The **insertMany** command returns an object with information about the insert operations.

### Find All Documents

Add a query that returns all the documents.

```
const findResult = await collection.find({}).toArray();
console.log('Found documents =>', findResult);
```

This query returns all the documents in the **documents** collection. If you add this below the insertMany example you'll see the document's you've inserted.

### Find Documents with a Query Filter

Add a query filter to find only documents which meet the query criteria.

```
const filteredDocs = await collection.find({ a: 3 }).toArray();
console.log('Found documents filtered by { a: 3 } =>', filteredDocs);
```

Only the documents which match 'a' : 3 should be returned.

### Update a document

The following operation updates a document in the **documents** collection.

```
const updateResult = await collection.updateOne({ a: 3 }, { $set: { b: 1 } });
console.log('Updated documents =>', updateResult);
```

The method updates the first document where the field **a** is equal to **3** by adding a new field **b** to the document set to **1**.  
updateResult contains information about whether there was a matching document to update or not.

### Remove a document

Remove the document where the field **a** is equal to **3**.

```
const deleteResult = await collection.deleteMany({ a: 3 });
console.log('Deleted documents =>', deleteResult);
```

### Index a Collection

**Indexes** can improve your application's performance. The following function creates an index on the **a** field in the **documents** collection.

```
const indexName = await collection.createIndex({ a: 1 });
console.log('index name =', indexName);
```

For more detailed information, see the **indexing strategies page**.

## Error Handling

If you need to filter certain errors from our driver we have a helpful tree of errors described in **etc/notes/errors.md**.

It is our recommendation to use instanceof checks on errors and to avoid relying on parsing error.message and error.name strings in your code. We guarantee instanceof checks will pass according to semver guidelines, but errors may be sub-classed or their messages may change at any time, even patch releases, as we see fit to increase the helpfulness of the errors.

Any new errors we add to the driver will directly extend an existing error class and no existing error will be moved to a different parent class outside of a major release. This means instanceof will always be able to accurately capture the errors that our driver throws.

```
const client = new MongoClient(url);
await client.connect();
const collection = client.db().collection('collection');

try {
  await collection.insertOne({ _id: 1 });
  await collection.insertOne({ _id: 1 }); // duplicate key error
} catch (error) {
  if (error instanceof MongoServerError) {
    console.log(`Error worth logging: ${error}`); // special case for some reason
  }
  throw error; // still want to crash
}
```

## Nightly releases

If you need to test with a change from the latest main branch our mongodb npm package has nightly versions released under the nightly tag.

npm install mongodb@nightly

Nightly versions are published regardless of testing outcome. This means there could be sematic breakages or partially implemented features. The nightly build is not suitable for production use.

## Next Steps

- MongoDB Documentation
- MongoDB Node Driver Documentation
- Read about Schemas
- Star us on GitHub

## License

Apache 2.0


© 2012-present MongoDB **Contributors**  
© 2009-2012 Christian Amor Kvalheim

### Keywords

mongodb driver official

## Provenance

Built and signed on

 **GitHub Actions**

[View build summary](#)

Source Commit

[github.com/mongodb/node-mongodb-native@bcb97ff](https://github.com/mongodb/node-mongodb-native@bcb97ff)

Build File

[.github/workflows/release.yml](#)

Public Ledger

[Transparency log entry](#)

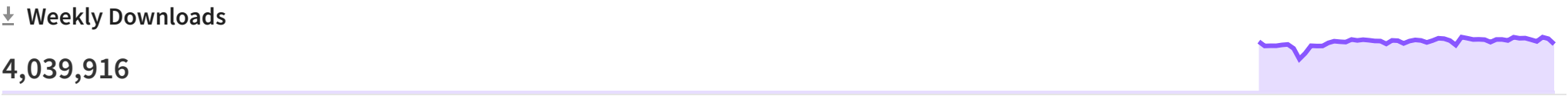
[Share feedback](#)


### Install

```
> npm i mongodb
```

Repository  
 [github.com/mongodb/node-mongodb-native](https://github.com/mongodb/node-mongodb-native)

Homepage  
 [github.com/mongodb/node-mongodb-native](https://github.com/mongodb/node-mongodb-native)



Version	License
6.2.0 	Apache-2.0
Unpacked Size	Total Files
2.84 MB	375

Last publish  
14 days ago

Collaborators



>Try on RunKit

🚩Report malware



Support

- Help
- Advisories
- Status
- Contact npm

Company

- About
- Blog
- Press

Terms & Policies

- Policies
- Terms of Use
- Code of Conduct
- Privacy