

[Products](#)[Pricing](#)[Documentation](#)[Sign Up](#)[Sign In](#)[Search](#)

## connect-mongo TS

4.6.0 • Public • Published 9 months ago

[Readme](#)[Explore](#) BETA[2 Dependencies](#)[669 Dependents](#)[65 Versions](#)

## connect-mongo

MongoDB session store for **Connect** and **Express** written in Typescript.

[npm](#) [v4.6.0](#)[downloads](#) [438k/month](#)[Sanity check](#) [passing](#)[coverage](#) [82%](#)

Breaking change in V4 and rewritten the whole project using Typescript. Please checkout the [migration guide](#) and [changelog](#) for details.

- [Install](#)
- [Compatibility](#)
- [Usage](#)
  - [Express or Connect integration](#)
  - [Connection to MongoDB](#)
- [Known issues](#)
  - [Native autoRemove causing error on close](#)
  - [MongoError exports circular dependency](#)

- Existing encrypted v3.2.0 sessions are not decrypted correctly by v4
- Events
- Session expiration
- Remove expired sessions
  - Set MongoDB to clean expired sessions (default mode)
  - Set the compatibility mode
  - Disable expired sessions cleaning
- Lazy session update
- Transparent encryption/decryption of session data
- Options
  - Connection-related options (required)
  - More options
  - Crypto-related options
- Development
  - Example application
  - Release
- License

## Install

---

```
npm install connect-mongo
```

```
yarn add connect-mongo
```

- You may also need to run `install mongodb` if you do not have it installed already because `mongodb` is not a `peerDependencies` instead.
- If you are upgrading from v3.x to v4, please checkout the [migration guide](#) for details.
- If you are upgrading v4.x to latest version, you may check the [example](#) and [options](#) for details.

## Compatibility

---

- Support Express up to `5.0`
- Support [native MongoDB driver](#) `>= 4.0`

- Support Node.js 12, 14 and 16
- Support **MongoDB** 3.2+

For extended compatibility, see previous versions **v3.x**. But please note that we are not maintaining v3.x anymore.

## Usage

---

### Express or Connect integration

Express 4.x , 5.0 and Connect 3.x :

```
const session = require('express-session');
const MongoStore = require('connect-mongo');
```

```
app.use(session({
  secret: 'foo',
  store: MongoStore.create(options)
}));
```

```
import session from 'express-session'
import MongoStore from 'connect-mongo'
```

```
app.use(session({
  secret: 'foo',
  store: MongoStore.create(options)
}));
```

### Connection to MongoDB

In many circumstances, `connect-mongo` will not be the only part of your application which need a connection to a MongoDB database. It could be interesting to re-use an existing connection.

Alternatively, you can configure `connect-mongo` to establish a new connection.

### Create a new connection from a MongoDB connection string

---

**MongoDB connection strings** are **the best way** to configure a new connection. For advanced usage, **more options** can be configured with `mongoOptions` property.

```
// Basic usage
app.use(session({
  store: MongoStore.create({ mongoUrl: 'mongodb://localhost/test-app'
}));

// Advanced usage
app.use(session({
  store: MongoStore.create({
    mongoUrl: 'mongodb://user12345:foobar@localhost/test-app?authSour
    mongoOptions: advancedOptions // See below for details
  })
}));
```

### Re-use an existing native MongoDB driver client promise

---

In this case, you just have to give your `MongoClient` instance to `connect-mongo`.

```
/*
** There are many ways to create MongoClient.
** You should refer to the driver documentation.
*/

// Database name present in the connection string will be used
app.use(session({
  store: MongoStore.create({ clientPromise })
}));

// Explicitly specifying database name
app.use(session({
```

```
store: MongoStore.create({
  clientPromise,
  dbName: 'test-app'
})
}));
```

## Known issues

---

**Known issues** in GitHub Issues page.

### Native autoRemove causing error on close

- Calling `close()` immediately after creating the session store may cause error when the async index creation is in process when `autoRemove: 'native'`. You may want to manually manage the autoRemove index. [#413](#)

### MongoError exports circular dependency

The following error can be safely ignore from [official reply](#).

```
(node:16580) Warning: Accessing non-existent property 'MongoError' of module exports inside ci
(Use `node --trace-warnings ...` to show where the warning was created)
```

### Existing encrypted v3.2.0 sessions are not decrypted correctly by v4

v4 cannot decrypt the session encrypted from v3.2 due to a bug. Please take a look on this issue for possible workaround. [#420](#)

## Events

---

A `MongoStore` instance will emit the following events:

Event name	Description	Payload
<code>create</code>	A session has been created	<code>sessionId</code>
<code>touch</code>	A session has been touched (but not modified)	<code>sessionId</code>

Event name	Description	Payload
update	A session has been updated	sessionId
set	A session has been created OR updated ( <i>for compatibility purpose</i> )	sessionId
destroy	A session has been destroyed manually	sessionId

## Session expiration

---

When the session cookie has an expiration date, `connect-mongo` will use it.

Otherwise, it will create a new one, using `ttl` option.

```
app.use(session({
  store: MongoStore.create({
    mongoUrl: 'mongodb://localhost/test-app',
    ttl: 14 * 24 * 60 * 60 // = 14 days. Default
  })
}));
```

**Note:** Each time an user interacts with the server, its session expiration date is refreshed.

## Remove expired sessions

---

By default, `connect-mongo` uses MongoDB's TTL collection feature (2.2+) to have `mongod` automatically remove expired sessions. But you can change this behavior.

**Set MongoDB to clean expired sessions (default mode)**

`connect-mongo` will create a TTL index for you at startup. You **MUST** have MongoDB 2.2+ and administration permissions.

```
app.use(session({
```

```
store: MongoStore.create({
  mongoUrl: 'mongodb://localhost/test-app',
  autoRemove: 'native' // Default
}))
}));
```

**Note:** If you use `connect-mongo` in a very concurrent environment, you should avoid this mode and prefer setting the index yourself, once!

#### Set the compatibility mode

In some cases you can't or don't want to create a TTL index, e.g. Azure Cosmos DB.

`connect-mongo` will take care of removing expired sessions, using defined interval.

```
app.use(session({
  store: MongoStore.create({
    mongoUrl: 'mongodb://localhost/test-app',
    autoRemove: 'interval',
    autoRemoveInterval: 10 // In minutes. Default
  })
}));
```

#### Disable expired sessions cleaning

You are in production environnement and/or you manage the TTL index elsewhere.

```
app.use(session({
  store: MongoStore.create({
    mongoUrl: 'mongodb://localhost/test-app',
    autoRemove: 'disabled'
  })
}));
```

## Lazy session update

---

If you are using **express-session**  $\geq 1.10.0$  and don't want to resave all the session on database every single time that the user refresh the page, you can lazy update the session, by limiting a period of time.

```
app.use(express.session({
  secret: 'keyboard cat',
  saveUninitialized: false, // don't create session until something s
  resave: false, //don't save session if unmodified
  store: MongoStore.create({
    mongoUrl: 'mongodb://localhost/test-app',
    touchAfter: 24 * 3600 // time period in seconds
  })
}));
```

by doing this, setting `touchAfter: 24 * 3600` you are saying to the session be updated only one time in a period of 24 hours, does not matter how many request's are made (with the exception of those that change something on the session data)

## Transparent encryption/decryption of session data

---

When working with sensitive session data it is **recommended** to use encryption

```
const store = MongoStore.create({
  mongoUrl: 'mongodb://localhost/test-app',
  crypto: {
    secret: 'squirrel'
  }
});
```

## Options

---

### Connection-related options (required)

One of the following options should be provided. If more than one option are provided,



each option will take precedence over others according to priority.

Priority	Option	Description
1	<code>mongoUrl</code>	A <b>connection string</b> for creating a new MongoClient connection. If database name is not present in the connection string, database name should be provided using <code>dbName</code> option.
2	<code>clientPromise</code>	A Promise that is resolved with MongoClient connection. If the connection was established without database name being present in the connection string, database name should be provided using <code>dbName</code> option.
3	<code>client</code>	An existing MongoClient connection. If the connection was established without database name being present in the connection string, database name should be provided using <code>dbName</code> option.

#### More options

Option	Default	Description
<code>mongoOptions</code>	<pre>{   useUnifiedTopology:     true }</pre>	Options object for <b>MongoClient</b> method. Can be used with <code>mongoUrl</code> option.
<code>dbName</code>		A name of database for storing sessions. It is used with <code>mongoUrl</code> or <code>clientPromise</code> . Takes precedence over others.

Option	Default	Descri
		database name p connection string
collectionName	'sessions'	A name of collect storing sessions.
ttl	1209600	The maximum lif (seconds) of the s will be used to se session.coo if it is not yet set. days.
autoRemove	'native'	Behavior for rem sessions. Possibl 'native', 'i and 'disable
autoRemoveInterval	10	Interval (in minu autoRemove (c interval.
touchAfter	0	Interval (in secur session updates.
stringify	true	If true , connect serialize sessions JSON.string setting them, and them with JSON when getting the useful if you are u MongoDB doesn'

Option	Default	Descri
<code>serialize</code>		Custom hook for sessions to Mong helpful if you need the session before
<code>unserialize</code>		Custom hook for sessions from Mc can be used in sc you need to support types of serializa objects and JSON need to modify tl before using it in
<code>writeOperationOptions</code>		Options object to MongoDB write c that supports it ( <code>remove</code> ). Use the write concern exception: If <code>auto</code> set to <code>'interval</code> concern from the <code>writeOperat</code> object will get ov
<code>transformId</code>		Transform origin <code>sessionId</code> in want to use as st
<code>crypto</code>		Crypto related op below.

#### Crypto-related options

Option	Default	Description
secret	false	Enables transparent crypto in accordance with <a href="#">OWASP session management recommendations</a> .
algorithm	'aes-256-gcm'	Allows for changes to the default symmetric encryption cipher. See <a href="#">crypto.getCiphers()</a> for supported algorithms.
hashing	'sha512'	May be used to change the default hashing algorithm. See <a href="#">crypto.getHashes()</a> for supported hashing algorithms.
encodeas	'hex'	Specify to change the session data cipher text encoding.
key_size	32	When using varying algorithms the key size may be used. Default value 32 is based on the AES blocksize.
iv_size	16	This can be used to adjust the default <b>IV</b> size if a different algorithm requires a different size.
at_size	16	When using newer AES modes such as the default GCM or CCM an authentication tag size can be defined.

## Development

```
yarn install
docker-compose up -d
# Run these 2 lines in 2 shell
yarn watch:build
```

```
yarn watch:test
```

## Example application

```
yarn link
cd example
yarn link "connect-mongo"
yarn install
yarn start
```

## Release

Since I cannot access the setting page. I can only do it manually.

### Install

1. Bump version, update CHANGELOG.md and README. Commit and push.  

```
> npm i connect-mongo
```
2. Run `yarn build && yarn test && npm publish`
3. `git tag vX.Y.Z && git push --tags`

### Repository

 [github.com/jdesboeufs/connect-mongo](https://github.com/jdesboeufs/connect-mongo)

## License

### Homepage

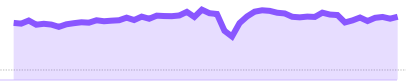
 [github.com/jdesboeufs/connect-mongo#readme](https://github.com/jdesboeufs/connect-mongo#readme)

The MIT License

### Weekly Downloads

## Keywords

105,451



Version	License
connect mongo mongodb session express 4.6.0	MIT

Unpacked Size	Total Files
109 kB	14

Issues	Pull Requests
11	1

Last publish  
9 months 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](#)