mongoose.connect('mongodb://user:pass@localhost:port/database');

// replica sets

**var** uri = 'mongodb://user:pass@localhost:port,anotherhost:port,yetanother:port/mydatabase';

mongoose.connect(uri);

// with options

mongoose.connect(uri, options);

// connecting to multiple mongos

**var** uri = 'mongodb://hostA:27501,hostB:27501';

**var** opts = { mongos: true };

mongoose.connect(uri, opts);

// optional callback that gets fired when initial connection completed

**var** uri = 'mongodb://nonexistent.domain:27000';

mongoose.connect(uri, **function**(error) {

// if error is truthy, the initial connection failed.

})

// grab the user model

var User = require('./app/models/user');

// create a new user

var newUser = User({

name: 'Peter Quill',

username: 'starlord55',

password: 'password',

admin: true

});

// save the user

newUser.save(function(err) {

if (err) throw err;

console.log('User created!');

});

**FIND ALL**

// get all the users

User.find({}, function(err, users) {

if (err) throw err;

// object of all the users

console.log(users);

});

**FIND ONE**

// get the user starlord55

User.find({ username: 'starlord55' }, function(err, user) {

if (err) throw err;

// object of the user

console.log(user);

});

**FIND BY ID**

// get a user with ID of 1

User.findById(1, function(err, user) {

if (err) throw err;

// show the one user

console.log(user);

});

**QUERYING**

You can also use MongoDB [query syntax](http://docs.mongodb.org/manual/reference/operator/query/).

// get any admin that was created in the past month

// get the date 1 month ago

var monthAgo = new Date();

monthAgo.setMonth(monthAgo.getMonth() - 1);

User.find({ admin: true }).where('created\_at').gt(monthAgo).exec(function(err, users) {

if (err) throw err;

// show the admins in the past month

console.log(users);

});

[**#Update**](https://scotch.io/tutorials/using-mongoosejs-in-node-js-and-mongodb-applications#update)

Here we will find a specific user, change some attributes, and then re-save them.

**GET A USER, THEN UPDATE**

// get a user with ID of 1

User.findById(1, function(err, user) {

if (err) throw err;

// change the users location

user.location = 'uk';

// save the user

user.save(function(err) {

if (err) throw err;

console.log('User successfully updated!');

});

});

Remember that since we created the function to change the updated\_atdate, this will also happen on save.

**FIND AND UPDATE**

An even easier method to use since we dont have to grab the user, modify, and then save. We are just issuing a mongodb findAndModify command.

// find the user starlord55

// update him to starlord 88

User.findOneAndUpdate({ username: 'starlord55' }, { username: 'starlord88' }, function(err, user) {

if (err) throw err;

// we have the updated user returned to us

console.log(user);

});

**FIND BY ID AND UPDATE**

// find the user with id 4

// update username to starlord 88

User.findByIdAndUpdate(4, { username: 'starlord88' }, function(err, user) {

if (err) throw err;

// we have the updated user returned to us

console.log(user);

});

[**#Delete**](https://scotch.io/tutorials/using-mongoosejs-in-node-js-and-mongodb-applications#delete)

**GET A USER, THEN REMOVE**

// get the user starlord55

User.find({ username: 'starlord55' }, function(err, user) {

if (err) throw err;

// delete him

user.remove(function(err) {

if (err) throw err;

console.log('User successfully deleted!');

});

});

**FIND AND REMOVE**

// find the user with id 4

User.findOneAndRemove({ username: 'starlord55' }, function(err) {

if (err) throw err;

// we have deleted the user

console.log('User deleted!');

});

**FIND BY ID AND REMOVE**

// find the user with id 4

User.findByIdAndRemove(4, function(err) {

if (err) throw err;

// we have deleted the user

console.log('User deleted!');

});

Room.find({}).sort('-date').exec(function(err, docs) { ... });

Room.find({}).sort({date: -1}).exec(function(err, docs) { ... });

Room.find({}).sort({date: 'desc'}).exec(function(err, docs) { ... });

Room.find({}).sort({date: 'descending'}).exec(function(err, docs) { ... });

Room.find({}).sort([['date', -1]]).exec(function(err, docs) { ... });

Room.find({}, null, {sort: '-date'}, function(err, docs) { ... });

Room.find({}, null, {sort: {date: -1}}, function(err, docs) { ... });