

Node.js - Mongoose



Mongoose

Defining your schema

```
var mongoose = require('mongoose');
var Schema = mongoose.Schema;

var blogSchema = new Schema({
  title: String,
  author: String,
  body: String,
  comments: [{ body: String, date: Date }],
  date: { type: Date, default: Date.now },
  hidden: Boolean,
  meta: {
    votes: Number,
    favs: Number
  }
});
```

```
var Blog = mongoose.model('Blog', blogSchema);
```

Mongoose

Mongoose provides a straight-forward, schema-based solution to model your application data

- Features
 - Schemas to model your entities
 - Built-in type casting
 - Predefined validation along with custom validations
 - Virtual attributes
 - Query building
 - Business logic hooks

```
// npm install mongoose --save
const mongoose = require('mongoose');
var Cat = mongoose.model('Cat', { name: String });
var kitty = new Cat({ name: 'Zildjian' });
mongoose.connect(url, function (err) {
  if (err) throw err;
  console.log('Successfully connected');
  kitty.save(function (err) {
    if (err) { console.log(err); } else {
      console.log('meow');
      mongoose.connection.close(); } }); });
```

Mongoose

Instance methods

```
// define a schema
var animalSchema = new Schema({ name: String, type: String });

// assign a function to the "methods" object of our animalSchema
animalSchema.methods.findSimilarTypes = function (cb) {
  return this.model('Animal').find({ type: this.type }, cb);
}
```

Now all of our animal instances have a findSimilarTypes method available to it.

```
var Animal = mongoose.model('Animal', animalSchema);
var dog = new Animal({ type: 'dog' });

dog.findSimilarTypes(function (err, dogs) {
  console.log(dogs); // woof
});
```

Mongoose

Statics

```
// assign a function to the "statics" object of our animalSchema
animalSchema.statics.findByName = function (name, cb) {
  return this.find({ name: new RegExp(name, 'i') }, cb);
}
```

```
var Animal = mongoose.model('Animal', animalSchema);
Animal.findByName('fido', function (err, animals) {
  console.log(animals);
});
```

Mongoose

Indexes

```
var animalSchema = new Schema({
  name: String,
  type: String,
  tags: { type: [String], index: true } // field level
});

animalSchema.index({ name: 1, type: -1 }); // schema level
```

Mongoose

Virtuals

```
var personSchema = new Schema({ // define a schema
  name: {
    first: String,
    last: String
  } });

var Person = mongoose.model('Person', personSchema);

// create a document
var per = new Person({
  name: { first: 'Walter', last: 'White' } });
```

```
console.log(per.name.first + ' ' + per.name.last); // Walter White
```

```
personSchema.virtual('name.full').get(function () {
  return this.name.first + ' ' + this.name.last; });
console.log('%s is..', per.name.full); // Walter White is..
```

Mongoose

Queries

```
var Person = mongoose.model('Person', yourSchema);
// find each person with a last name matching 'Ghost', selecting the
`name` and `occupation` fields
Person.findOne({ 'name.last': 'Ghost' }, 'name occupation', function
(err, person) {
  if (err) return handleError(err);
  console.log('%s %s is a %s.', person.name.first, person.name.last,
person.occupation) // Space Ghost is a talk show host.
})
```

Mongoose

- <http://mongoosejs.com/docs/guide.html>
- Tarea: Modelar con Mongoose el mismo ejemplo planteado para NodeJS