
Mongoose

The MongoDB ODM

Overview

- [What is Mongoose?](#)
 - [Defining a model in Mongoose](#)
 - [Custom methods](#)
 - [Creating documents](#)
 - [Reading documents](#)
 - [Updating documents](#)
 - [Deleting documents](#)
 - [Using pre](#)
 - [Peer challenge](#)
-

What is Mongoose?

Survey says...

- An object modeling package for Node
 - Does Object Document Mapping (ODM)
 - Maps a Mongo document to an object in your code
 - Simplifies making MongoDB queries in Node
-

Defining a model

Mongoose models

- Constructors that we define
 - Represent documents in our database
 - Defined by a Mongoose Schema object
-

—

Example

Talent model (1 of 4)

```
// require mongoose
var mongoose = require('mongoose');
// get the mongoose Schema object
var Schema = mongoose.Schema;
```

```
// create our schema
var talentSchema = new Schema({
  talent_id: Number,
  name: {type: String, required: true},
  address: String,
  skills: [String],
  wage_req: [{
    min: Number,
    max: Number,
    wage_type: {type: String, enum: ['salary',
    'hourly']}
  }],
});
```

Talent model (3 of 4)

```
    history: [{  
        org: String,  
        start: Date,  
        end: Date  
    }],  
    current: String  
});
```

Talent model (4 of 4)

```
// create the model
var Talent = mongoose.model('Talent',
talentSchema);
// export it to make it available
module.exports = Talent;
```

Custom methods

We can...

Create custom methods for our schemas

—

Example

Check wage min/max values

```
talentSchema.methods.normalizeWageRange =  
function() {  
    this.wage_req.forEach(function(el, ind, arr) {  
        if(el.min > el.max) {  
            var temp = el.min;  
            el.min = el.max;  
            el.max = temp;  
        }  
    }  
}
```

Creating documents

Create a talent document

```
var Talent = require('./models/talent');

var talent = new Talent({
  talent_id: 9,
  name: 'Lisa Leslie',
  address: 'Gardena, CA',
  skills: ['basketball'],
  ...});
talent.save(function(err){
  if (err) throw err;
});
```

Reading documents

Read all talent documents

```
var Talent = require('./models/talent');

Talent.find({}, function(err, talents) {
  if(err) throw err;

  // array with objects representing talents
  console.log(talents);
});
```

Read matching documents

```
var Talent = require('./models/talent');

var criteria = {name: 'Lisa Leslie'};

Talent.find(criteria, function(err, talents) {
  if(err) throw err;

  // array with found talents
  console.log(talents);
});
```

Read 1 matching documents

```
var Talent = require('./models/talent');

var criteria = {name: 'Lisa Leslie'};

Talent.findOne(criteria, function(err, talent) {
  if(err) throw err;

  // talent object
  console.log(talent);
});
```

Read by ID

```
...  
var ObjectId = mongoose.Types.ObjectId;  
var id = ObjectId('566e11de839c3c945065f730');  
  
Talent.findById(id, function(err, talent) {  
    if(err) throw err;  
  
    // talent object  
    console.log(talent);  
});
```

Advanced querying

```
...
Talent.find({})
    .where('talent_id')
    .gt(4)
    .exec(function(err, talents) {
        if(err) throw err;

        // all talents with talent_ids > 4
        console.log(talents);
    });
```

[More query operators](#)

Updating documents

Read and update

```
var Talent = require('./models/talent');

var criteria = {name: 'Lisa Leslie'};
var update = {current: 'Los Angeles Sparks'};

Talent.findOneAndUpdate(criteria, update,
  function(err, talent) {
    if(err) throw err;

    // updated talent object
    console.log(talent);
  });
```

Read by ID and update

```
...  
var ObjectId = mongoose.Types.ObjectId;  
var id = ObjectId('566e11de839c3c945065f730');  
var update = {current: 'Los Angeles Sparks'};  
  
Talent.findByIdAndUpdate(id, update,  
    function(err, talent) {  
        if(err) throw err;  
  
        // updated talent object  
        console.log(talent);  
    });
```

Deleting documents

Read and delete

```
var Talent = require('./models/talent');

var criteria = {name: 'Lisa Leslie'};

Talent.findOneAndRemove(criteria, function(err) {
  if(err) throw err;

  console.log('Talent deleted');
});
```

Read by ID and delete

```
...  
var ObjectId = mongoose.Types.ObjectId;  
var id = ObjectId('566e11de839c3c945065f730');  
  
Talent.findByIdAndRemove(id, function(err) {  
    if(err) throw err;  
  
    console.log('Deleted talent');  
});
```

Using pre

What is pre?

A mongoose method ("hook") that allows us to perform a function prior to executing a call to the database.

[More on mongoose hooks](#)

—

Example

Run our method before save

```
// on every save, perform this logic
talentSchema.pre('save', function(next) {
  // check the wage range validity
  this.normalizeWageRange();

  next();
});
```

Resources

Installing mongoose Node pkg

1. Open a terminal
 2. Go to your project directory
 3. Type `npm install mongoose --save`
 4. Press enter key
-

Mongoose schema types

- String
- Number
- Date
- Buffer
- Boolean
- Mixed
- ObjectId
- Array

<http://mongoosejs.com/docs/schematypes.html>

Advanced query operators

<https://docs.mongodb.org/manual/reference/operator/query/>

Pre and post hooks

<http://mongoosejs.com/docs/middleware.html>

Peer Challenge

Instructions

1. [Fork this repo](#)
 2. Go to `routes/talent.js`
 3. Update each of the routes with the correct logic
 4. Use your REST client of choice to test that it works
-