

Stack Overflow is a community of 4.7 million programmers, just like you, helping each other.

Join them; it only takes a minute:

Sign up

Join the Stack Overflow community to:

Ask programming questions

Answer and help your peers

Get recognized for your expertise

Mongoose find()

I just started playing with mongoose and mongo. I have the following code:

```
var ninjaSchema = mongoose.Schema({
  name: String,
  skill: Number
});

var Ninja = mongoose.model('Ninja',ninjaSchema);

module.exports = {
  init : function(){
    console.log('Connecting to database');
    mongoose.connect('mongodb://localhost/mydb');
    var db = mongoose.connection;
    db.on('error', console.error.bind(console, 'connection error:'));
    db.once('open', function callback () {
      console.log('Successfully connected!');
    });
  },
  createNinja : function(name,skill){
    var n = new Ninja({name:name,skill:skill});
    n.save(function(err,n){
      if (err)
        console.log('saving failed');
      console.log('saved '+ n.name);
    });
  },
  getNinjas : function(){
    var res = null;
    res = Ninja.findOne({}, 'name skill',function(err,docs){
      if (err)
        console.log('error occured in the query');
      return 'ninja name: '+docs.name+' ninja skill: '+docs.skill;
    });
    return res;
  }
}
```

There is no problem in adding entries to the database but I have problems retrieving them. I am a bit confused about how the whole thing works. my understanding is the following: there are the schemas, which are like classes in oop, so just a blueprint for a record in the database. The model is a record, OK, maybe a bit more, since I saw that you can add a method to the model. Well... I don't really understand how to use them. Can you give me a clue what really are they? Back to the subject: when issuing the find command, it calls the anonymous function and docs should be the result right? Now how do I access them? Since now if I log the res I get the following:

```
{ options: {},
  safe: undefined,
  _conditions: {},
  _updateArg: {},
  _fields: { name: 1, skill: 1 },
  _geoComparison: undefined,
  op: 'findOne',
  model:
    { [Function: model]
      base:
        { connections: [Object],
          plugins: [],
          models: [Object],
          modelSchemas: [Object],
          options: {} },
        modelName: 'Ninja',
        model: [Function: model],
        db:
          { base: [Object],
            collections: [Object],
            models: {},
            replica: false,
            hosts: null,
            host: 'localhost',
            port: 27017,
            user: undefined,
            pass: undefined,
            name: 'mydb',
            options: [Object],
            _readyState: 1,
            _closeCalled: false,
            _hasOpened: true,
            _listening: true,
            _events: [Object],
            db: [Object] },
          schema:
            { paths: [Object],
              subpaths: {},
              virtuals: [Object],
              nested: {},
              inherits: {},
              callQueue: [],
              _indexes: [],
              methods: {},
              statics: {},
              tree: [Object],
              _requiredpaths: [],
              options: [Object],
              _events: {} },
            options: undefined,
            collection:
              { collection: [Object],
                opts: [Object],
                name: 'ninjas',
                conn: [Object],
                queue: [],
                buffer: false } } } }
```

Also if I use `Ninja.find(...,function(err,docs){ ... })` how do I go trough the docs? Or how do I retrieve my records?

node.js mongoose express

asked Apr 6 '13 at 14:21

Pio

1,706 ● 4 ● 23 ● 48

1 Answer

I've found the fault. It was more of a conceptual one: I am dealing with asynchronous calls and am trying to return the result from another function and don't know when it will execute. So what happens is I make the request that the db query be executed and return the result, which turns out to be null. This code:

```
getNinjas : function(){
  var res = null;
  Ninja.find({}, 'name skill',function(err,docs){
    if (err)
      console.log('error occured in the database');
    console.log(docs);
  });
  return res;
}
```

returns null, but! the console.log(docs) prints to the console all the values from the database, what I was trying to do. Now I need to make changes, most likely pass a callback which will be executed upon the receiving of the results.

With the changes the code looks like this:

```
getNinjas : function(res){
  var twisted = function(res){
    return function(err, data){
      if (err){
        console.log('error occured');
        return;
      }
      res.send('My ninjas are:\n');
      console.log(data);
    }
  }

  Ninja.find({}, 'name skill',twisted(res));
}
```

So like this I am able to pass the response object so I can send the name of my ninjas :)

edited Feb 28 '14 at 15:13

Arcane Engineer

2,936 ● 3 ● 30 ● 62

answered Apr 6 '13 at 15:27

Pio

1,706 ● 4 ● 23 ● 48

1

Ah you beat me to it. I was working up a nice answer for you, but you got it. You'll want to pass a callback to which you send the results. - [numbers1311407](#) Apr 6 '13 at 15:29