

# Nesting Sub Documents





How we can nest sub-documents

# Object format

```
{  
  Name: 'Patrick',  
  Age: 38,  
  Books: [  
    {title: 'Name of the Wind', pages: 400},  
    {title: 'Wise Man's Fear', pages: 500}  
  ]  
}
```



Create file "nesting\_test.js" in the folder "test"

```
const assert = require('assert');
const mongoose = require('mongoose');
const Author = require('../models/authors');

// Describe our tests
describe('Nesting records', function(){

  // Create tests
  it('Creates an author with sub-documents', function(done){

    var pat = new Author({
      name: 'Patrick Rothfuss',
      books: [{title: 'Name of the Wind', pages: 400}]
    });

    pat.save().then(function(){
      Author.findOne({name: 'Patrick Rothfuss'}).then(function(record){
        assert(record.books.length === 1);
        done();
      });
    });
  });
});
```

# Terminal

npm run test

The screenshot shows the Robo 3T 1.4 application window. On the left, a tree view shows the database structure: 'testaroo' > 'Collections (2)' > 'authors'. The main panel displays the query `db.getCollection('authors').find({})` and its results. The results are shown in a table with columns 'Key', 'Value', and 'Type'.

Key	Value	Type
(1) ObjectId("61235ec62b3e1f4960a54f32")	{ 4 fields }	Object
_id	ObjectId("61235ec62b3e1f4960a54f32")	ObjectId
name	Patrick Rothfuss	String
books	[ 1 element ]	Array
_v	0	Int32

```
...
describe('Nesting records', function(){
  beforeEach(function(done){
    // Drop the collection
    mongoose.connection.collections.authors.drop(function(){
      done();
    });
  });

  it('Creates an author with sub-documents', function(done){
    ...
  });

  it('Adds a book to an author', function(done){
    var pat = new Author({
      name: 'Patrick Rothfuss',
      books: [{title: 'Name of the Wind', pages: 400}]
    });
    pat.save().then(function(){
      Author.findOne({name: 'Patrick Rothfuss'}).then(function(record){
        // add a book to the books collection
        record.books.push({title: 'Wise Man's Fear', pages: 500});
        record.save().then(function(){
          Author.findOne({name: 'Patrick Rothfuss'}).then(function(record){
            assert(record.books.length === 2);
            done();
          });
        });
      });
    });
  });
});
```



Insert code

The diagram shows two arrows originating from the text 'Insert code'. One arrow points to the `beforeEach` block, and the other points to the `it('Adds a book to an author')` block.

# Terminal

npm run test

Robo 3T - 1.4

File View Options Window Help

New Connection (6)

- System
- AlienDBex
- config
- django\_mongodb\_tu...
- testaroo
  - Collections (2)
    - authors
    - mariochars
  - Functions
  - Users

db.getCollection('authors').find(...

New Connection localhost:27017 testaroo

```
db.getCollection('authors').find({})
```

authors 0 sec.

Key	Value	Type
(1) ObjectId("6123604bc90ada22a435cf96")	{ 4 fields }	Object
_id	ObjectId("6123604bc90ada22a435cf96")	ObjectId
name	Patrick Rothfuss	String
books	[ 2 elements ]	Array
[0]	{ 3 fields }	Object
_id	ObjectId("6123604bc90ada22a435cf97")	ObjectId
title	Name of the Wind	String
pages	400	Int32
[1]	{ 3 fields }	Object
_id	ObjectId("6123604bc90ada22a435cf9a")	ObjectId
title	Wise Man's Fear	String
pages	500	Int32
_v	1	Int32