Building a Model

Mark Scott
linkedin.com/in/tripletdad99
@tripletdad99

pluralsight
hardcore dev and IT training

Introduction

Build Model…
From Schema

Relationship
Model ←→ Document

Construct,
Save Documents

Our First Model

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

var customerSchema = new Schema({
        name:       String,
        address:    String,
        city:       String,
        state:      String,
        country:    String,
        zipCode:    Number,
        createdOn:  Date,
        isActive:   Boolean
});

// Build a model from the customer schema…
var Customer = mongoose.model('Customer', customerSchema);

// Add a custom property to the schema…
customerSchema.add({ discountCode: String });

var DiscountedCust = mongoose.model('DiscountCust', customerSchema);
```

```javascript
// standup.server.model.js file

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

// Stand-up Meeting Notes Schema Definition...
var standupSchema = new Schema({
    memberName: String,
    project: String,
    workYesterday: String,
    workToday: String,
    impediment: String,
    createdOn: { type: Date, default: Date.now }
});

// Expose (export) the model now...
module.exports = mongoose.model( 'Standup', standupSchema );
```

Documents

Mongoose Document   MongoDB Document

```
var personSchema = new Schema({
    firstName: String,
    lastName: String
});

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

var bob = new Person({
    firstName: 'Bob',
    lastName: 'Doe'
});
```

Sub-Documents

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

// Pre-define sub-documents...
var subCategory = new Schema({
        name: String,
        description: String,
        isActive: Boolean });

var subAnswers = new Schema({
        answerText: String,
        isCorrect: Boolean,
        displayOrder: Number });

var subQuestions = new Schema({
        type: { type: String },
        text: String,
        answers: [ subAnswers ] });

// Define main document schema...
var quizSchema = new Schema({
    name: String,
    description: String,
    categories: [ subCategory ],
    questions: [ subQuestions ]
});
```

name
description
isActive

categories [...]

questions [...]
  type
  text

  answers [...]

    answerText
    isCorrect
    displayOrder

```javascript
// build Quiz model from schema
var Quiz = mongoose.model('Quiz',
quizSchema);

// document instance of model
var quiz1 = new Quiz({
    name: 'Favorite Things Quiz',
    description: 'Demo Quiz',
    categories: [{
        name: 'Favorites',
        description: 'Favorite
things quiz category',
        isActive: true }]

// abbreviated...
});

// save document...
quiz1.save(callback);
```

name
description
isActive

categories [...]
  name
  description
  isActive

questions [...]
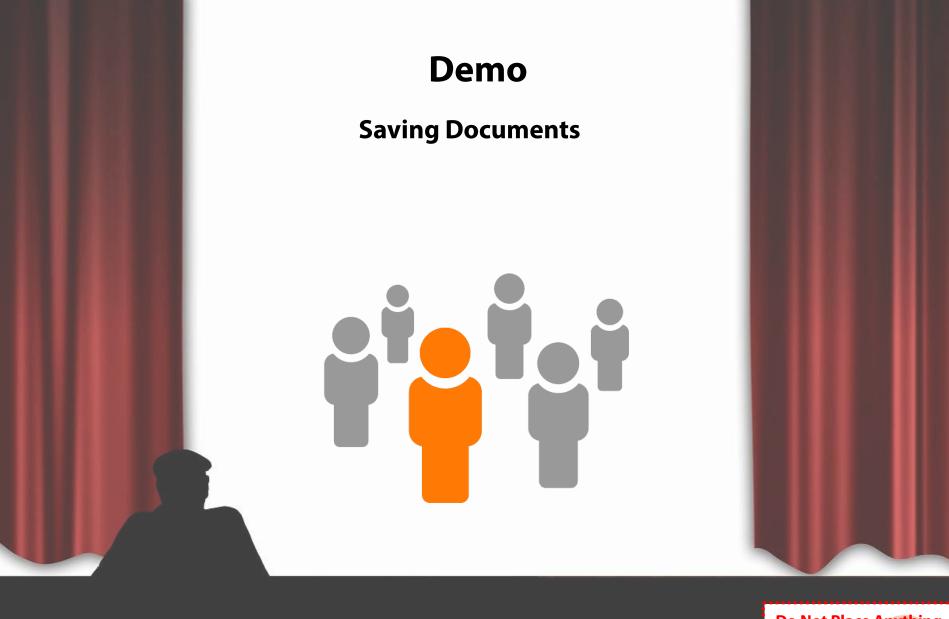
answers [...]

```javascript
var categories = [];
// Fill in the categories array
var cat1 = { name: 'Test1', description: 'Test 1 category', isActive: true };
var cat2 = { name: 'Test2', description: 'Test 2 category', isActive: true };

categories.push(cat1, cat2);

var questions = [];
// Add in some questions and answers...
var q1 = { type: 'Multiple-Choice', text: 'What is your favorite color?',
      answers: [
            { answerText: 'Red', isCorrect: false, displayOrder: 1 },
            { answerText: 'White', isCorrect: false, displayOrder: 2 },
            { answerText: 'Blue', isCorrect: true, displayOrder: 3 }
      ] };

var q2 = { type: 'Multiple-Choice', text: 'What is your favorite animal?',
      answers: [
            { answerText: 'Dog', isCorrect: true, displayOrder: 1 },
            { answerText: 'Cat', isCorrect: false, displayOrder: 2 },
            { answerText: 'Squirrel', isCorrect: false, displayOrder: 3 }
      ] };

questions.push(q1, q2);

// Create the parent quiz document - supply the categories and questions now...
var quiz2 = new Quiz({
    name: 'Example Quiz',
    description: 'Example Quiz long description...',
    categories: categories,
    questions: questions });
```

Demo

Saving Documents

Module Summary

**Build Model…
From Schema**

Relationship
Model ←→ Document

**Construct,
Save Documents**