

Build your own REST
APIs with



Disclaimer

- I am no expert in this, just toying around...
- Can't be unseen

What is node.js?

- Javascript framework to build network applications
- Event-driven, non-blocking I/O
- Built on Chrome's Javascript runtime (V8)

Why do I care?

- Simple enough to build your own REST API for your apps
- Can interact with databases, especially with NoSQL databases like MongoDB which is based on JSON.

Yes, this MongoDB...

```
} catch (Exception e) {  
    if (!((_ok) ? true : (Math.random() > 0.1))) {  
        return res;  
    }  
}
```

Express.js

- Popular module that makes node.js even easier to use

Client side bonus

- Client frameworks written in Javascript like Backbone.js or Batman.js (made here at Shopify) let you build clients and servers using the same language

Code walkthrough

Config

```
1  var application_root = __dirname,
2      express = require("express"),
3      path = require("path"),
4      mongoose = require("mongoose");
5
6  var app = express();
7
8  // Config
9  app.configure(function() {
10     app.set('port', process.env.PORT || 3000);
11     app.set('view engine', 'jade');
12     app.use(express.logger());
13     app.use(express.bodyParser());
14     app.use(express.methodOverride());
15     app.use(app.router);
16     app.use(express.static(path.join(application_root, "public")));
17     app.use(express.errorHandler({ dumpExceptions: true, showStack: true }));
18 });
```

Database+Schema+Validation+Model

```
20 // Database
21 mongoose.connect('mongodb://localhost/cocoaheads_database');
22
23 // Schema
24 var Schema = mongoose.Schema;
25
26 var MeetingSchema = new Schema({
27   date: { type: Date, required: true },
28   location: { type: String, required: true },
29   agenda: { type: String },
30   modified: { type: Date, default: Date.now }
31 });
32
33 // Validations
34 MeetingSchema.path('date').validate(function (v) {
35   now = new Date();
36   return v > now;
37 }, 'Meeting date should be in the future');
38
39 // Model
40 var MeetingModel = mongoose.model('Meeting', MeetingSchema);
41
```

REST APIs

```
42 // URLs
43 app.get('/', function(req, res) {
44     res.render('index.jade', {layout: false});
45 })
46
47 app.get('/api', function(req, res) {
48     res.send('Cocoaheads Meetings API is running');
49 });
50
51 app.get('/api/meetings', function(req, res) {
52     return MeetingModel.find(function(err, meetings) {
53         if (!err) {
54             return res.send(meetings);
55         } else {
56             return console.log(err);
57         }
58     });
59 });
60
```

REST APIs (cont'd)

```
61 app.post('/api/meetings', function(req, res) {
62     var meeting;
63
64     console.log("POST: ");
65     console.log(req.body);
66
67     meeting = new MeetingModel({
68         date: req.body.date,
69         location: req.body.location,
70         agenda: req.body.agenda
71     });
72
73     meeting.save(function(err) {
74         if (!err) {
75             return console.log("created");
76         } else {
77             return console.log(err);
78         }
79     });
80
81     return res.send(meeting);
82 });
```

REST APIs (cont'd)

```
84 app.get('/api/meetings/:id', function(req, res) {
85     return MeetingModel.findById(req.params.id, function(err, meeting) {
86         if (!err) {
87             return res.send(meeting);
88         } else {
89             return console.log(err);
90         }
91     });
92 });
93
94 app.put('/api/meetings/:id', function(req, res) {
95     return MeetingModel.findById(req.params.id, function(err, meeting) {
96         meeting.date = req.body.date;
97         meeting.location = req.body.location;
98         meeting.agenda = req.body.agenda;
99         meeting.modified = new Date();
100
101         return meeting.save(function(err) {
102             if (!err) {
103                 console.log("updated");
104             } else {
105                 console.log(err);
106             }
107
108             return res.send(meeting);
109         });
110     });
111 });
112
```

REST APIs (cont'd)

```
112
113 app.delete('/api/meetings/:id', function(req, res) {
114     return MeetingModel.findById(req.params.id, function(err, meeting) {
115         return meeting.remove(function(err) {
116             if (!err) {
117                 console.log("removed");
118                 return res.send('');
119             } else {
120                 console.log(err);
121             }
122         });
123     });
124 });
125
```

Server launch

```
126 // Launch Server
127 app.listen(app.get('port'));
128
129 console.log('Server listening on port ' + app.get('port'));
```

Demo

Thank you!

@philippegitard