MEAN Stack

27 Jun 2013 - Kirk Morales

About Me

- Background in web analytics consulting, analysis, and technical implementation
- Web application developer & product manager
- Currently working at AMEX, but shifting to freelance
- Competitive Olympic Weightlifter

about.me/kirk.morales
@kirkmorales
kirk.morales@gmail.com

MmEAN Stack

MongoDB
NoSQL BSON document-based database.

MongooseJS
MongoDB object modeling for NodeJS

ExpressJS
Web Application Framework for NodeJS.

AngularJS
JavaScript MVC framework.

NodeJS
Platform build on V8 runtime engine for building fast, scalable, event-driven, non-blocking network applications.

MongoDB (mongodb.org)

- Non-relational store of BSON objects - superset of JSON, which includes binary data.
- Documents can be queried on any object property.
- Atomic operations with guaranteed unique document IDs.
- Indexes can be made on any combination of object properties for faster querying.
- Data types include: String, Numbers, Binary, Mongo Date, ObjectID (document references), sub-objects, arrays.
- Robust querying - exact matches, regex, >, <, in array, exists, geo-spatial.
- Horizontal (sharding) and vertical (replica sets) scaling.

Mongoose (mongoosejs.com)

- Object modeling for MongoDB
- Document instance methods - extend your Mongo documents with custom functions.
- Common Mongo operation wrappers (find, update, etc)
- Indexes enforcement
- Strict data type enforcement

ExpressJS (expressjs.com)

- Simple web application framework
- Support for robust path routing, sessioning, templates (Jade), custom middleware
- Automatic query string and body parsing
- Auto conversion between objects and JSON

AngularJS (angularjs.org)

- MVC JavaScript Framework
- Client-side routing
- Templates
- Filters
- Event handlers
- Custom services

NodeJS (nodejs.org)

- Event-driven
- Non-blocking
- Uses Google V8 runtime engine
- Lightweight
- Plugins exist to extend to production-ready server:
  - Clustering
  - Daemonizing
  - Logging

Building the App

Create Server

app.js - NodeJS, Mongoose, MongoDB, ExpressJS

- Declare dependencies
- Create new express server
- Set configuration options for dev/production/all
- Define routes
- Connect to MongoDB
- Bind server to port

Define Post Schema

lib/post.js - Mongoose

```
var postSchema = new mongoose.Schema({
  date:     {type: Date, default: Date.now},
  author:   String,
  title:    String,
  content:  String
});

// Create a reverse index on date so we can sort on it quicker
postSchema.index({date: -1});

// Create post object from schema
var Post = mongoose.model('Post', postSchema);
```

Create Route Handlers

lib/post.js - ExpressJS

Defined as exported functions

- getPosts
- getPostContent
- savePost

Create Page Template

views/index.jade - Jade (ExpressJS), AngularJS

- Use Jade syntax to declare HTML5
- Write template variable to page

```
var blogPosts = !{JSON.stringify(blogPosts)};
```

- Markup with AngularJS syntax

Create Angular Controllers

public/js/app.js - AngularJS

- Controller for post list
  - Posts are loaded with the page and brought into scope
  - Post content is loaded via AJAX

- Controller for adding a new post
  - Adds content via AJAX and pushes onto post list

Kirk Morales

about.me/kirk.morales
@kirkmorales
kirk.morales@gmail.com

END