Deep Dive: MEANJS 3

MEAN STACK

- MongoDB
  - https://docs.mongodb.com/
- Express
  - http://expressjs.com/en/guide/routing.html
- AngularJS
  - https://docs.angularjs.org/tutorial
  - **NOTE: Mean uses angular 1.x Angular 2.x+ is developed very differently.**
- Node
  - https://nodejs.org/en/about/
  - https://developers.google.com/v8/

Other Tools

- Mongoose
  - http://mongoosejs.com/docs/index.html
  - ODM - Object Document Mapper
- SocketIO
  - https://socket.io/get-started/chat/
- Bootstrap
  - https://getbootstrap.com/docs/3.3/

Review CRUD

- CREATE (POST)
  - /articles/
- READ (GET)
  - /articles/ - gets a list of items.
  - /articles/:id - gets a specific id.
- UPDATE (PUT)
  - /articles/:id
- DELETE (DELETE)
  - /articles/:id

NOTE: Other urls can be used, not just these!

Common HTTP Status codes

- https://en.wikipedia.org/wiki/List_of_HTTP_status_codes
- 200 OK
- 201 Created
- 301 Moved Permanently - Browser usually caches this.
- 302 Found - Regular redirection without caching
- 400 Bad Request - Client error. Like bad form data.
- 401 Unauthorized - Authentication is required.
- 403 Forbidden - Permission.
- 404 Not Found - Resource or URI not found.
- 405 Method Not Allowed - Request method is not supported.
- 500 Internal Server Error - Something went wrong.
- 502 Bad Gateway - Has to do with proxy.

Clarification on Document based storage (MongoDB)

Document refers to a JSON like structure.

```
[{
  "_id": "sldkfjsdf",
  "title": "This is a document.",
  "content": "This is the content of a document.",
  "views": 5
},
{
  "_id": "sldkfjsdg",
  "title": "This is a document 2.",
  "content": "This is the content of a document 2.",
  "views": 1,
  "likes": 0,
  "dislikes": 0,
  "tags": [
    "document", "content"
  ]
}]
```

Relational Database

_id	title	content	views
sldkfjsdf	This is a document.	This is the content of a document.	5
sldkfjsdg	This is document 2.	null	0

MongoDB Document Features (compared to Relational Database).

- Document can take any structure.
    - Document 1 and 2 can exist in the same collection.
    - No limit in number of 'columns' or sub-documents.
    - In a relational database, this cannot happen. Document 1 has to have the same fields as document 2.
- _id is unique across the database.
    - MongoDB is more of a key/value type of Document database.
    - In a relational database, multiple records in different tables can have _id 1.
- Collections are used instead of tables.
    - While document 1 and 2 are different, they share a lot in common. You group these in collections.
    - Types are also not enforced (at the database level). Use software (like Mongoose to do this).
    - For relational tables, the number of columns and types are enforced.
    - Indexes can be applied to these documents.

Let's look at code!