Persistence

with MongoDB

MongoDB

- An example of a NoSQL database
- Is schema-less
  - Do not define tables and columns in advance
  - Store new data however is needed
- Stored as BSON
    - Similar to JSON, but with a few more data types
      - JSON is essentially serialized JavaScript objects
        - » I.e. what JavaScript objects would look like if represented as an object literal
    - Therefore MongoDB essentially stores JavaScript objects
      - Easy to save a JavaScript object
      - Easy to restore as a JavaScript object

SQL vs NoSQL

- Browse comparison on:
  - http://www.mongodb.org/display/DOCS/SQL+to+Mongo+Mapping+Chart

Vs. RDBMS

- It is unclear whether it is beneficial or not to be thinking in terms of RDBMS and mapping it to Mongo.
- My intuition:
  - Forget about RDBMS in this case
  - Just see Mongo as a simple way to store, query, and retrieve JavaScript objects

Database structure

- Database
  - A database is a set of collections
- Collections
  - A collection is a set of documents
- Documents
  - A document is (essentially) a JSON string

Alternative mental model

- Save and find JSON documents
  - Each JSON document is not restricted to have the same structure, but they mostly do
  - Each document has (globally) unique _id
- A *Collection* is a set of JSON documents
- A *Database* is a set of JSON collections

Example document

- What is passed to mongodb:

  { "name" : "apple",

  "price" : 1.99}

- Mongodb adds in an _id:

  { "name" : "apple",

  "price" : 1.99,

  "_id" : ObjectId("35414c4ebb264d7000000000") }

Mongod, mongo, mongodb drivers

- Mongod – the MongoDB database server
  - Listens by default on port 27017
  - Requests / responses via a MongoDB protocol
- Mongo – a MongoDB shell application
  - A JavaScript shell to interact with MongoDB
  - Can do all database operations
- MongoDB drivers
  - Exist for many languages
  - Provides a language-specific API for interacting with MongoDB

Who uses MongoDB?

- Scan:
  - https://www.mongodb.com/who-uses-mongodb

Our use…

- SQL and NoSQL DBMSs each have their strengths
- Our purpose in 67-328:
  - Exposure: have basic knowledge of it
  - An easy way to store and retrieve data in a form very close to JavaScript objects (BSON)

Installing MongoDB

- Browse to:
  https://docs.mongodb.com/manual/administration/install-community/
- Follow the download instructions:
  - MacOS: I found installing HomeBrew and then MongoDB to be easy.
- Experiment with mongo shell:
  - Start:
    https://docs.mongodb.org/getting-started/shell/import-data/
  - Follow their example through *Remove Data*

Studio 3T

- Studio 3T has a free version to view/edit mongoDB databases

Node.js MongoDB Driver

- In the mongo shell, you can directly interact with the mongod in a REPL.
- To interact with mongod from within a node program, use the npm module mongodb:
  - npm install mongodb -- save

Review mongo-fruit.js

- Run mongodb
- Run mongo
- Run mongodb example

- You can find the API for the demonstrated collection methods at:
  - http://mongodb.github.io/node-mongodb-native/2.2/api/Collection.html

Lab / Homework

- Due by Wednesday (or first 10 min in class):
  - mLab Introduction
- Due November 20
  - Implement simple CRUD operations for what you are persisting in your final project.
  - Have at least a single web page from which you can get, post, (put,) and delete documents to a MongoDB database.
  - You don't have to do all your final project collections, but you need to do one collection of documents, and at least 3 attributes per document.