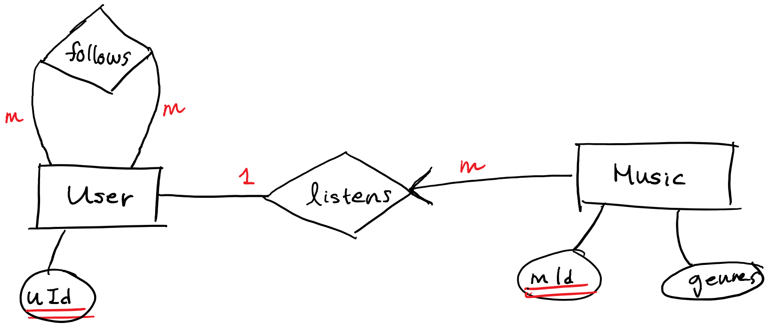
Node.js + Express + MongoDB / Moongoose + Chrome Advanced REST client. + Mocha

Chrome Advanced REST client: Manual HTTP Testing

Mocha: Testing / Script Runner.

ER Model



- Note : system will keep track of every song that a user has listened to rather than enforcing a limit.

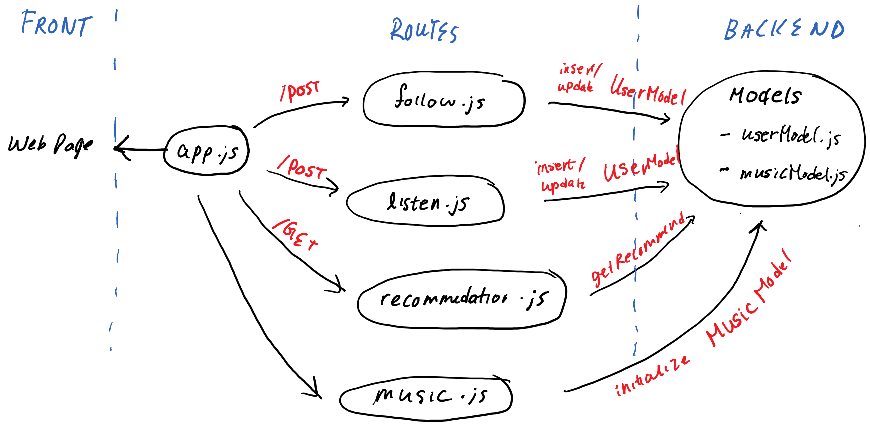
(2) server loads music.json upon startup

(3) scripts.md loads

- follow.json
- listen.json
- displays recommendations of user A

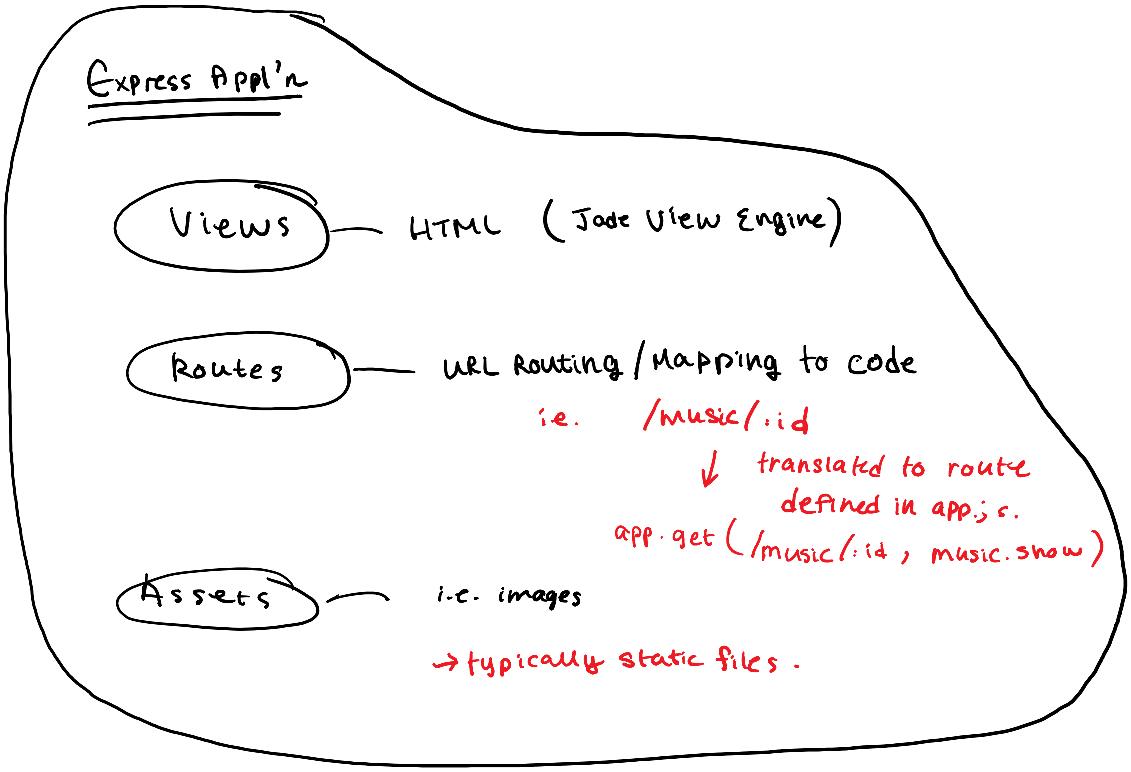
(1) Runner to initiate

High level Components



Due Diligence

Thursday, February 11, 2016 3:48 AM



Express Appl'n

Views — HTML (Jade View Engine)

Routes — URL Routing / Mapping to code

i.e. /music/:id

↓ translated to route defined in app.js.

app.get (/music/:id , music.show)

Assets — i.e. images

→ typically static files.

Required Routes

(1) POST /follow

(2) POST /listen

(3) GET /recommendations.

Mongoose

- Object modelling Tool used w/ MongoDB (JS)
- validators, defaults, getter, setter, indexes etc.
- allows schema def'ns in MongoDB (otherwise schemaless)

Mocha

- Promotes behavior driven development
- Node JS / Expect JS.
- Offers :
  - Reporting
  - Asynchronous support
  - Rich Configurability

Mocha Hooks

→ "logic executed when associated event occurs"

↳ before suite / before each / after / after Each () } support for async.

↳ clean-up test / setup ⇒ dB data