Donor’s Application

Crossover Assessment by Mateusz Grzesiukiewicz

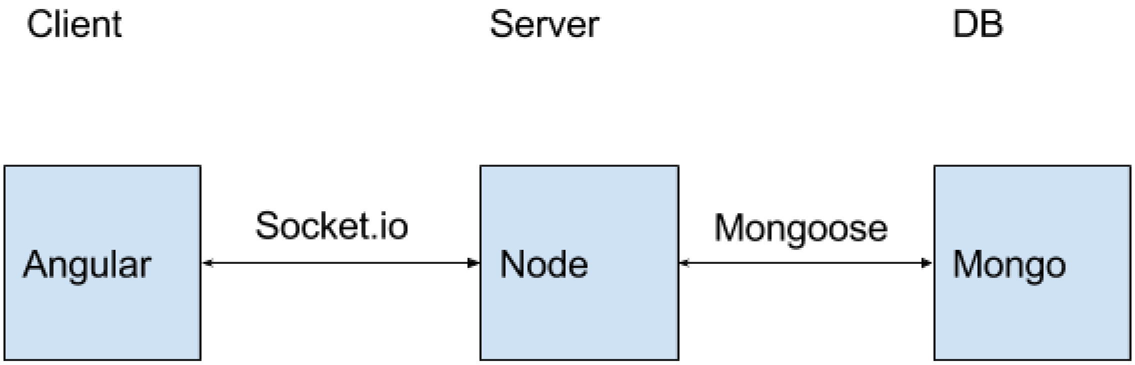
Hi Crossover team, I’m delighted that I can show off with this assignment. I promise you will be happy with the work done. However, some of the bits remain unfinished.

Technology stack: MEAN

- Angular + Esri (Arcgis) Angular plugin
- Mongo + Mongoose
- Express
- Node + Npm

Architecture

High level overview



Frontend - Angular

Modules:

- Core
- Donorsmaps
- Donorpins
- User (Not used)

1. Highly configurable: app/config
2. Separation of concerns in each module:
   Controllers, Routing, Routing policies, Components, Views, Css, Images, Models
3. Especially component “Donor pin” along with controller to handle creating/updating/deleting.
4. Prod / Dev / Local environments
5. Dockerizing
6. Package managers: bower + npm

Web sockets

Server

Maintains array of connected clients. For each client it stores its id and range that map displays. The server is responsible for deciding whether message should be sent to client or not (outside range). By message we mean either of those:

- Remove donor pin
- Update donor pin
- Create donor pin

Server additionally listens on individual clients:

- Range update message: then broadcasts to this client updated pins per new range.

Based on the calculation server:

- Sends update/create/delete message to selected clients.

Client

Maintains array of donor pins displayed on the page. Each array element consists of:

- reference to the donor pin model object,
- reference to layer on the map.

When donor pin should be added/deleted/updated the array is being used as pointer to the objects and if necessary addition of the new ones / removal of existing ones.

Backend Server API

Serves as connector to the Mongo store using Mongoose. The API is exposing the following endpoint:
/api/donorpins/:donorpinId

Additionally as per range message coming through web sockets it performs range lookup of pins:

```
Donorpin.find({ x: { '$gte': currentConnections[client.id].range.xmin, '$lte':
currentConnections[client.id].range.xmax }, y: { '$gte':
currentConnections[client.id].range.ymin, '$lte': currentConnections[client.id].range.ymax } })
```

It is also in charge of maintaining accessability (which is Donor's application is free for all - if you have link).

Database - Mongo

Convinient storage when working with JSONs. It is robust and supports indexing. Mongo server 3.x (Wired Tiger) is lately 3 to 7 times faster.
Donor pin model is pretty straightforward:
First name, Last Name, Contact number, Email address, Blood group, Latitude, Longitude, X, Y,

Plus additionally:

- discreteIdentifier used for generating link: not selected by default
- User information
- Creation date

Mongoose schema is also validated on the backend side. Obviously.

Screenshots - Modeling

As the assignment is mostly technical I decided no to draw views before creating. The initial though was having a map on the front page that is clickable.

- Click opens popup form to create new donor pin
- Clicking on the donorpin shows information on the right hand side in special box/popup. (This unfortunatelly I failed to acomplish: some trouble with binding click event and popup form provided by ArcGis)