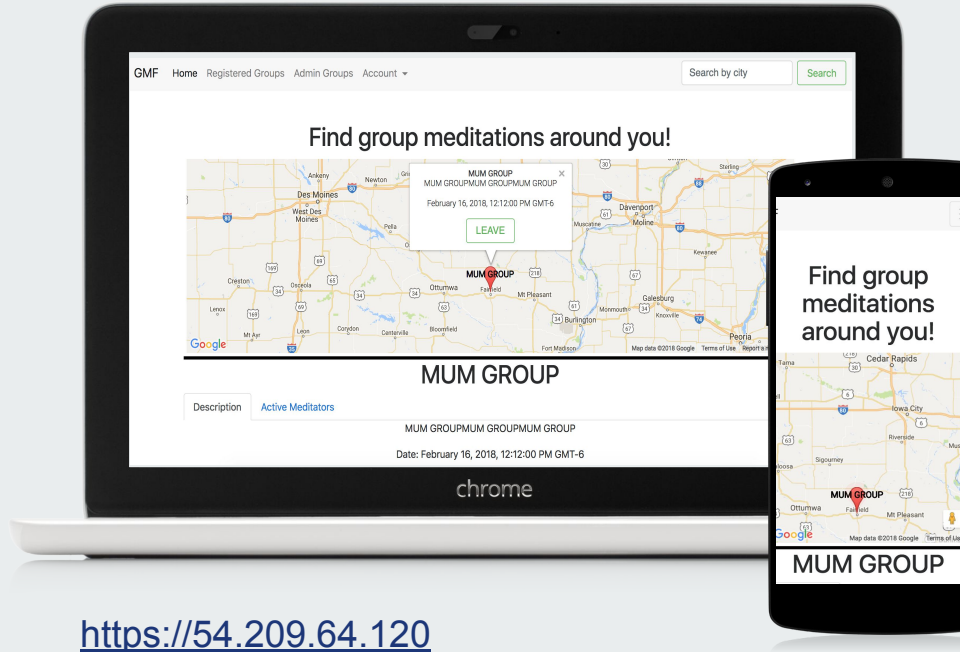




Group Meditation Finder



<https://54.209.64.120>

Outline

Github links

Technologies

User and authentication management

Google maps integration

Group CRUD & Component

GitHub

Backend Repository

<https://github.com/hunterTR/GroupMeditationFinder-Backend>

Frontend Repository

<https://github.com/hunterTR/GroupMeditationFinder-Frontend>

Technologies

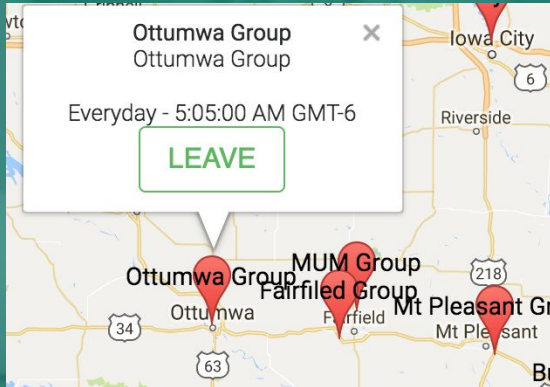


We made them better with;

Mongoose, Express

ng-bootstrap

Google Maps



We are listing meditation groups on the map.

Group CRUD - Model

```
var groupSchema = new mongoose.Schema({
  name: String,
  description: String,
  datetime: Date,
  address: {
    street: String,
    city: String,
    state: String,
  },
  geolocation: [Number],
  users: [{
    name: String,
    joinDate: Date
  }],
  isEveryday: Boolean,
  createDate: Date,
  updateDate: Date,
}, {
  versionKey: false
});
groupSchema.index({geolocation: '2d'});

module.exports = mongoose.model('Group', groupSchema, 'groups')
```

```
> {
  _id: ObjectId("5a7a265bd13e2b2828e98677"),
  address: Object
    street: "1000 N 4th Street"
    city: "Ottumwa "
    state: "IA"
  users: Array
    > 0: Object
      name: "Ottumwa Group"
      description: "Ottumwa Group"
      datetime: 2018-02-16 05:05:00.000
  geolocation: Array
    0: -92.41327288560569
    1: 41.02138394542758
  isEveryday: true
  createDate: 2018-02-06 16:04:11.836
  updateDate: 2018-02-06 16:04:11.836
}
```

Group CRUD - REST API

[routes\groups.js](#)



Path	HTTP VERB	Description
/api/groups/?page=p	GET	Get a list a group data list with page number.
/api/groups//:id([0-9a-f]{24})	GET	Get a single group data
/api/groups/	POST	Create a new group
/api/groups//:id([0-9a-f]{24})	PUT	Update a group
/api/groups//:id([0-9a-f]{24})	DELETE	Delete a group

Group CRUD - ng Component

[group/group.component.ts](#)

[group/group.component.html](#)

Create					
Name	Address	Geolocation	Datetime	User Count	Operations
mum group 1	1000 N 4th Street Fairfield IA	-91.925602 41.0697175	02/15/2018 09:30	0	<button>Update</button> <button>Delete</button>
mum group 1	1000 N 4th Street Fairfield IA	-91.925602 41.0697175	02/15/2018 09:30	0	<button>Update</button> <button>Delete</button>
mum group 1	1000 N 4th Street Fairfield IA	-91.925602 41.0697175	02/15/2018 09:30	0	<button>Update</button> <button>Delete</button>
mum group 1	1000 N 4th Street Fairfield IA	-91.925602 41.0697175	02/15/2018 09:30	0	<button>Update</button> <button>Delete</button>
mum group 1	1000 N 4th Street Fairfield IA	-91.925602 41.0697175	02/15/2018 09:30	0	<button>Update</button> <button>Delete</button>
mum group 1	1000 N 4th Street Fairfield IA	-91.925602 41.0697175	02/15/2018 09:30	0	<button>Update</button> <button>Delete</button>
wwwww	teststt test IA	-90.745918 41.0203226	02/10/2018 06:06	1	<button>Update</button> <button>Delete</button>
wwwww	teststt test IA	-90.745918 41.0203226	02/10/2018 06:06	1	<button>Update</button> <button>Delete</button>
Ottumwa Group	1000 N 4th Street Ottumwa IA	-92.413272 41.0213839	02/16/2018 05:05	1	<button>Update</button> <button>Delete</button>
Ottumwa Group	1000 N 4th Street Ottumwa IA	-92.413272 41.0213839	02/16/2018 05:05	1	<button>Update</button> <button>Delete</button>

< 1 2 >

Group CRUD - ng Component

[group/group.component.ts](#)

[group/group.component.html](#)

Add/Edit Group

×

Name

mum group 1

Description

mum group 1

Datetime

02/15/2018 09:30

Street

1000 N 4th Street

City

Fairfield

State

Iowa

Geolocation

-91.9365883525461,40.999276863674744



isEveryday



Submit

Close

User (Model/Services/Controller)

```
var location = new mongoose.Schema({
  type: String,
  coordinates: [Number]
});

var openID = new mongoose.Schema({
  provider: String,
  token: String,
  id: String
});

var userSchema = new mongoose.Schema({
  name: String,
  password: String,
  email: String,
  role: String,
  openID: openID,
  location: location
});

userSchema.index({email: 1});
```

```
service.create = create; // create new user
service.authenticate = authenticate; // authenticate the user
service.activate = activate; // activate the user account
service.getByJWT = getByJWT; // read the user data from JWT
service.update = update; // update the user data
service.addKey = addKey; // add admin's key
```

```
router.post('/auth', authChecker, authenticate);
router.post('/register', registerChecker, register);
router.post('/addkey', addKey);
router.get('/user', getUser); //this read the user from his token
router.put('/:_id', update);
```



Security Handling

- Adding Form validations for the user while signing-up.
- Adding Checkers for the req params on each endpoint (using express-validator).
- Hashing the password with salt.
- Generate JWT with a secret key.
- Using guards for the routes (authGuard/adminGuard).

Referenced from:

<http://jasonwatmore.com/post/2017/02/22/mean-with-angular-2-user-registration-and-login-example-tutorial>

Admins


The idea of the admins account management is allowing the root admin to add new admin keys, that has to be used by any new admin to register.

```
var adminKeySchema = new mongoose.Schema({  
  key: String  
});  
  
adminKeySchema.index({key: 1});
```


Register New

UserAdmin


Name

 Admin


E-Mail Address

 you@example.com


Password

 Password

Confirm Password

 Repeat Password

Admin Key

 Admin's Secret Key

Register

Already Member?

Validation samples

Register New

UserAdmin

Name

23

✗ Name must be at least 4 characters long

E-Mail Address

tess

✗ Email is invalid

Password

••

✗ Password must be at least 6 characters long

Confirm Password

•••••

✗ Password not match!

Register

Already Member?

Register New

UserAdmin

selem

mselem@mum.edu

Wrong!
Email "mselem@mum.edu" is already in our system!

Deployment

- Database, Backend, and Frontend deployed on one t2.micro EC-2 instance.
- Serving the dist output of ng-cli using Apache and proxy the API to the running node server on port 3000, and Rewriting the routes for the SPA through this .htaccess (No needs to handle it in EXPRESS / API)

```
RewriteEngine On
RewriteBase /
RewriteRule ^index\.html$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.html [L]
```

- Adding self-certified SSL to allow the browser reading the location from the user.

Scalability [TOO much meditators]



- The Front-End eligible to be running and distributed over CDN.
- Sharding the DB with Geo-Sharding and Distributed Spatial Indexing (R*-tree)
- Divide Backed-End API into microservices (behind load-balancers)to be like.
 - Services for the public search query (searching for groups).
 - Services for the logged in users (users operations).

More time, More features..



- Allow the Users to save his preferences about the meditations time slots, and we could send him notifications once new group created near to him within this time slots.
- Make the users to donate for the admin who creating the GM, to help the users organize more and more GM's.
- Receive notification if a group has been canceled or rescheduled.
- Allow the meditators to leave feedbacks for the admins for their GM organization,.....

Questions??

