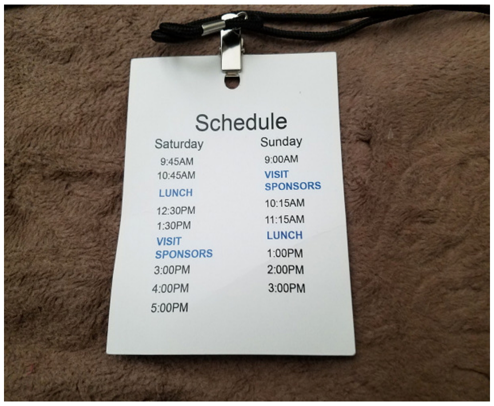
Silicon Valley Codecamp 2017

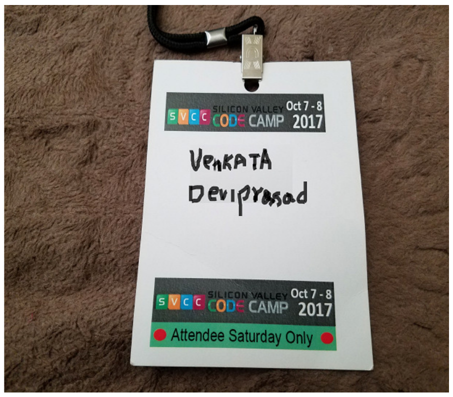
October 7 & 8

Venkata Deviprasad Sura,011409294,venkatadeviprasad.sura@sjsu.edu



Schedule

Saturday	Sunday
9:45AM	9:00AM
10:45AM	VISIT SPONSORS
LUNCH	10:15AM
12:30PM	11:15AM
1:30PM	LUNCH
VISIT SPONSORS	1:00PM
3:00PM	2:00PM
4:00PM	3:00PM
5:00PM	

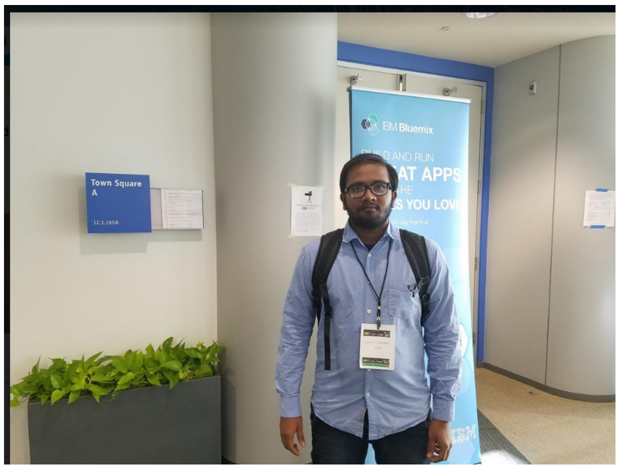


SVCC SILICON VALLEY CODE CAMP Oct 7 - 8 2017

Venkata
Deviprasad

SVCC SILICON VALLEY CODE CAMP Oct 7 - 8 2017

Attendee Saturday Only



Attended sessions:

- Cracking the Coding Interview
- Full Stack Development with JavaScript and NoSQL
- Recursion and DP for coding interview

Session 1:
<u>Title: Cracking the coding Interview:</u>

<u>Url : https://www.siliconvalley-codecamp.com/Session/2017/cracking-the-coding-interview</u>

Programming interviews are a different breed from other interviews and, as such, require specialized skills and techniques. This talk will teach you how to prepare for coding interviews, what top companies like Google, Amazon, Facebook, and Microsoft really look for, and how to tackle the toughest programming and algorithm problems. This is not a fluffy be-your-best talk; it is deeply technical and will discuss specific algorithm and data structure topics.

Comments: The questions and their type in interviews were discussed, Way to approach aproblem and difference between conventional and real way of solving a problem were discussed, type of problems and their frequency in various interviews were discussed

SPEAKERS:Gayle Laakmann McDowell

Session 2:
<u>Title:Full Stack Development with JavaScript and NoSQL:</u>

<u>Url:https://www.siliconvalley-codecamp.com/Session/2017/full-stack-development-with-javascript-and-nosql</u>

Being a backend developer, frontend developer, or system administrator often isn't enough in the modern technology space. Now it is often the expectation that the developer is responsible for all parts of the stack, making them what can be referred to as a Full Stack Developer. We're going to see how to leverage Node.js towards building a very modular web application that can be expanded beyond the web to things like mobile, desktop, and the Internet of Things (IoT). In particular, we're going to be looking at NoSQL as our data layer, Node.js as our logic layer, and Angular as our client layer when building a web application. A breakdown of the session can be seen below: * Building a RESTful API with Node.js that has several CRUD endpoints * Connecting the Node.js application to a NoSQL database for flexible data storage in JSON format and complex querying * Creating and consuming API data from an Angular with TypeScript frontend application As an attendee, you'll walk away with knowledge that can be applied towards creating your own very powerful, yet lightweight, full stack web applications.

Comments: Full stack development using angular, no sql and node js were discussed, a sample application was performed , letting us understand how to use nosql database and using it to create asimple web application.

Speaker : Nic Raboy

Session 3:
Title: Recursion and DP for coding interviews:

URL: https://www.siliconvalley-codecamp.com/Session/2017/recursion-and-dp-for-coding-interviews

It is no secret that the bar to crack technical interviews is only getting higher with each passing year.

Recursion is pervasive in many such interviews. Whether it is dealing with basic algorithms of Sorting, whether it's simple data-structures like Linked Lists, Arrays & Strings, or whether it's about traversing complex data-structures like Trees and Graphs, you're bound to come across Recursion-based solutions.

Recursion can be a difficult concept to grasp for many of us, but it doesn't have to be. It's one of those things that can be reigned with practice, and once reigned, it gives dividends all through your practice.

Once you're comfortable with recursion, then Dynamic Programming also becomes easier, because Recursion is the first logical step to solving many DP problems. In this session, we'll patiently walk through some Recursion and DP problems, until time permits.

Comments: Discussions on
What is DP?
how does it start? what is the main start of DP?
A simple example of a binary tree and dp in finding its height and depth were discussedd.

(Extended Code/Project Assignment or Volunteer):

I chose to try out the demo from the talk myself, I chose full stack development over the three topics, i tried using Nodejs, mongoDB , angular and express to create a simple web application myself:

Summary of my work:

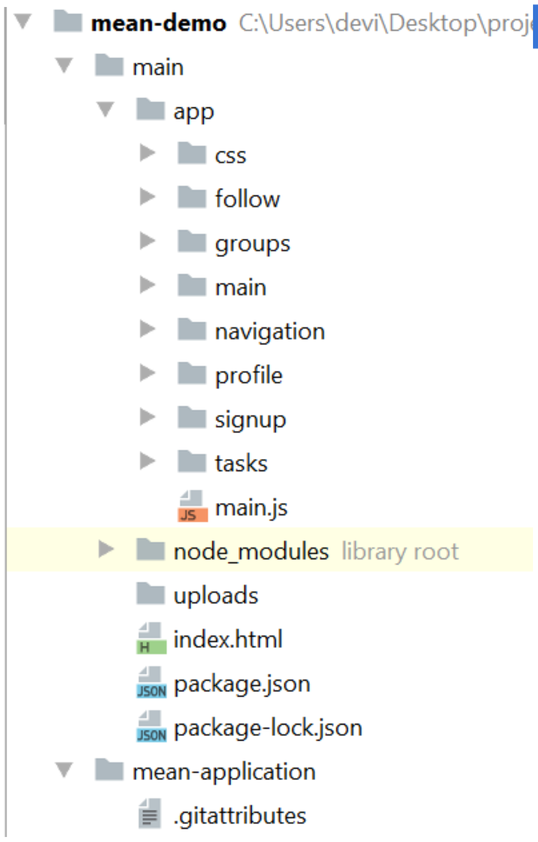
I created a social network roomie application, where all users can post information, form groups , see other post and notifications.

Using node js to cretae server and by running node main, server is ready,
As a part of database, i used MongoDb, installed using nom install and connection to the mongoDb is done by using mongoose.connect, the routing between different js routes are done by expressjs, all are mentioned in main server,js

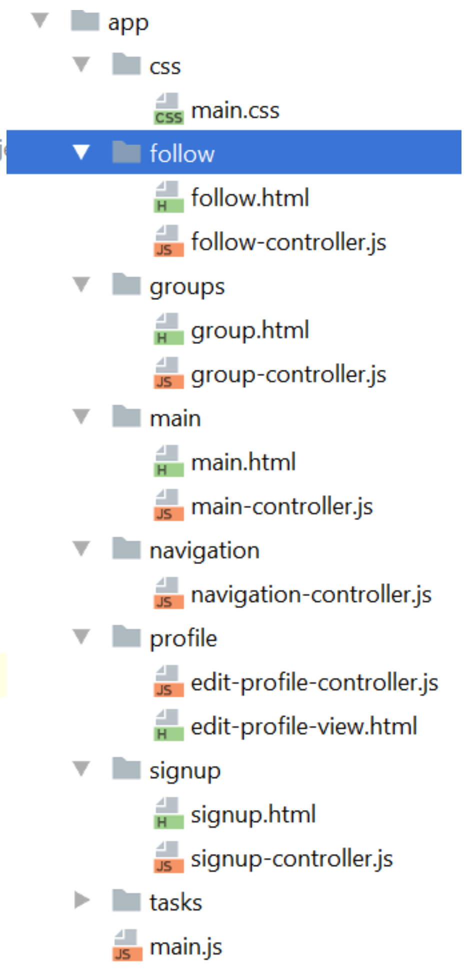
I created datasets to make the user insert or retrieve data using datasets, using angular in to display the get api data and post the post api data.

My directory structure for all items excluding server components as below,

First structure is just an outline and second one show all frontend controller components, and css

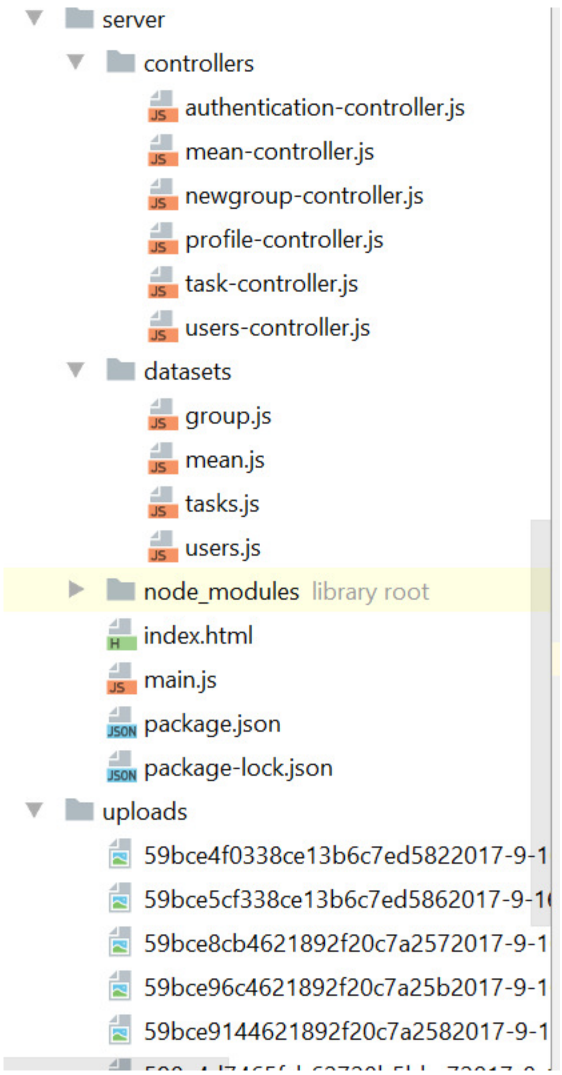


```
▼ mean-demo  C:\Users\devi\Desktop\proj
    ▼ main
        ▼ app
            ▶ css
            ▶ follow
            ▶ groups
            ▶ main
            ▶ navigation
            ▶ profile
            ▶ signup
            ▶ tasks
              main.js
        ▶ node_modules  library root
          uploads
          index.html
          package.json
          package-lock.json
    ▼ mean-application
          .gitattributes
```



```
▼ app
    ▼ css
          main.css
    ▼ follow
          follow.html
          follow-controller.js
    ▼ groups
          group.html
          group-controller.js
    ▼ main
          main.html
          main-controller.js
    ▼ navigation
          navigation-controller.js
    ▼ profile
          edit-profile-controller.js
          edit-profile-view.html
    ▼ signup
          signup.html
          signup-controller.js
    ▶ tasks
      main.js
```

My server component directory structure is listed as below:

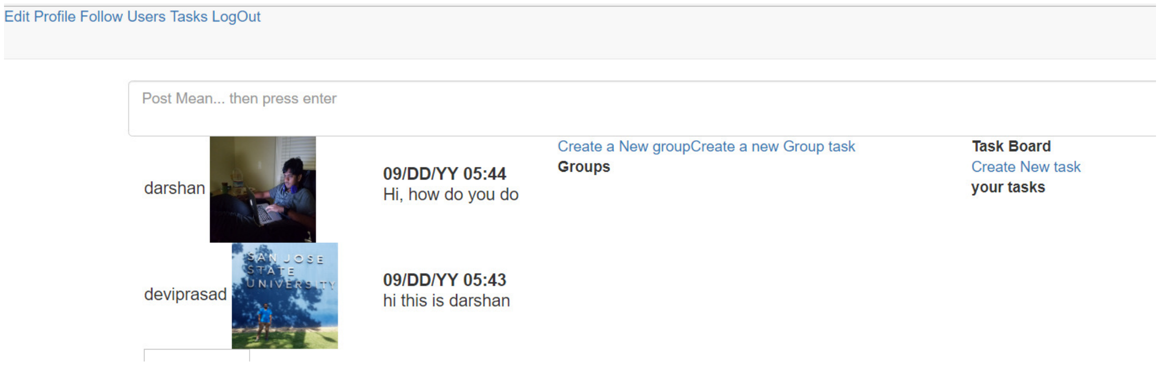


```
▼ server
  ▼ controllers
      authentication-controller.js
      mean-controller.js
      newgroup-controller.js
      profile-controller.js
      task-controller.js
      users-controller.js
  ▼ datasets
      group.js
      mean.js
      tasks.js
      users.js
  ▶ node_modules  library root
    index.html
    main.js
    package.json
    package-lock.json
▼ uploads
    59bce4f0338ce13b6c7ed5822017-9-1
    59bce5cf338ce13b6c7ed5862017-9-1
    59bce8cb4621892f20c7a2572017-9-1
    59bce96c4621892f20c7a25b2017-9-1
    59bce9144621892f20c7a2582017-9-1
```

The server componenets have datasets and controllers backend and you can also find the upload section where we upload the users photo

Upon running the server node main, and mongod we could see our page as below

Routing from front end:
I have used ui-sref state references newly introduced in angular js



Edit Profile Follow Users Tasks LogOut

Post Mean... then press enter

darshan 09/DD/YY 05:44 Hi, how do you do

deviprasad 09/DD/YY 05:43 hi this is darshan

Create a New groupCreate a new Group task

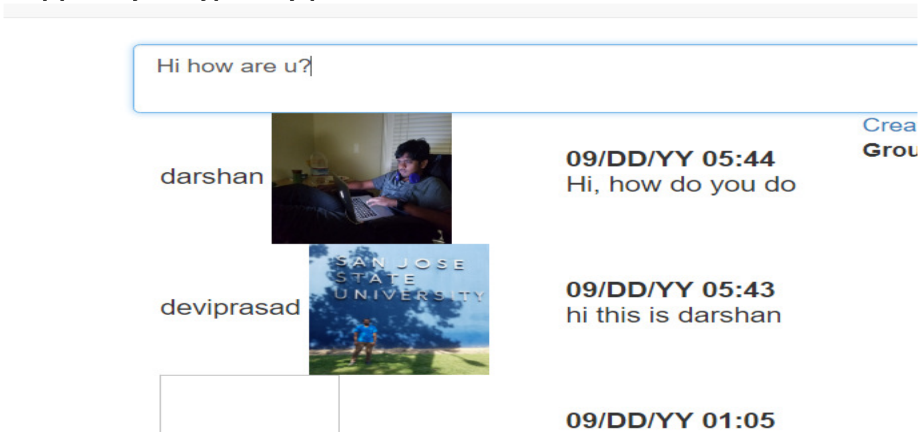
Groups

Task Board

Create New task

your tasks

We v=can post any info if you want , see other peoples info suppose you type any post info as below:



Hi how are u?

darshan 09/DD/YY 05:44 Hi, how do you do

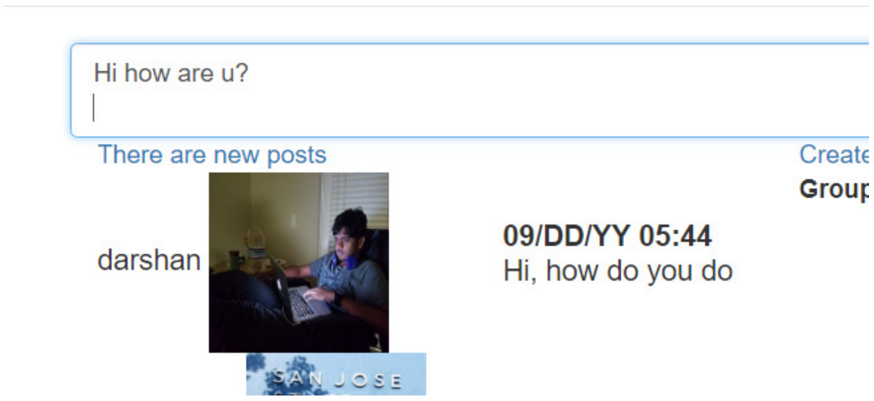
deviprasad 09/DD/YY 05:43 hi this is darshan

09/DD/YY 01:05

Crea

Grou

After pressing enter in text box:
polling is done and you will find something of this sort(There are new Posts)



Hi how are u?

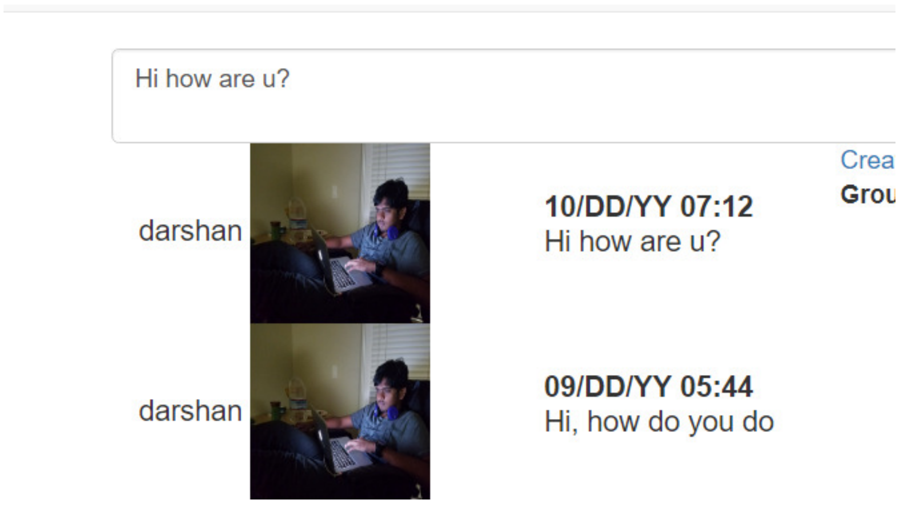
There are new posts

darshan 09/DD/YY 05:44 Hi, how do you do

Create

Group

Upon clciking the link or just refreshing page, you will find the new post

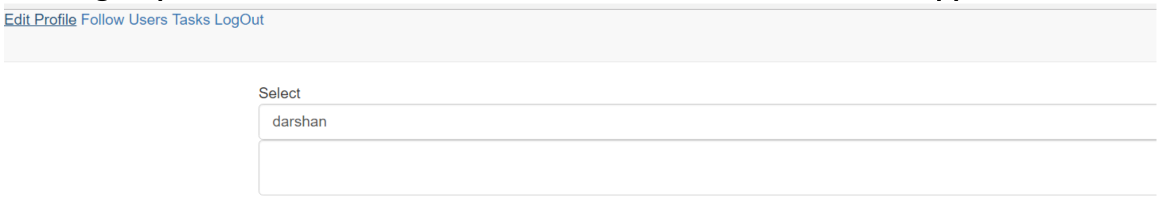

Hi how are u?

Crea
Grou

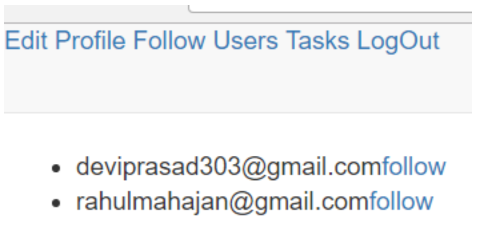
darshan 10/DD/YY 07:12
Hi how are u?

darshan 09/DD/YY 05:44
Hi, how do you do

User can edit his profile, change username, profile ;
create groups, follow other users, here are some screen shots of the application:


Edit Profile Follow Users Tasks LogOut

Select
darshan


Edit Profile Follow Users Tasks LogOut

- deviprasad303@gmail.comfollow
- rahulmahajan@gmail.comfollow

Development: the groups section is added in the ui part, idea is to make user able create their own groups;
The task section is also added, user should be able to addd tasks and get notified individually or in a group,

Main js application code:

```
var express=require('express');
var mongoose=require('mongoose');
var body_parser=require('body-parser');
var path=require('path');
//var multer=require('multer');
var multipart=require('connect-multiparty');
var multipartMiddleware=multipart();

var app=express();
var authenticationController=require('../server/controllers/authentication-
controller');
var profileController=require('../server/controllers/profile-controller');
var meanController=require('../server/controllers/mean-controller');
var usersController=require('../server/controllers/users-controller');
var tasksController=require('../server/controllers/task-controller');
var groupController=require('./controllers/newgroup-controller');
mongoose.connect('mongodb://localhost:27017/mean-demo', {useMongoClient: true});
app.use(multipartMiddleware);
app.use(body_parser.json());
//app.use(multipartMiddleware);
app.use('/app',express.static(path.join(__dirname,'../main', "/app")));
app.use('/node_modules',express.static(path.join(__dirname ,'../main',"/node_mod
ules")));
app.use('/uploads',express.static(path.join(__dirname,'../../mean-
demo','/uploads')));
app.get('/api/mean/get',meanController.getMean);

app.get('/api/user/get',usersController.getUsers);
app.get('/api/group/GetAllNames',usersController.getUserNames);
app.get('/api/group/GetAllTasks',usersController.GetAllTasks);
//app.get('/api/group/GetAllTasksbyId',usersController.getgroups);
//app.get('/api/group/GetAllGroupsbyid',tasksController.getGroups);
app.get('/',function(req,res){
res.sendFile(path.join(__dirname, '../main', 'index.html'));});


app.post('/api/user/login',authenticationController.login);
app.post('/api/user/signup',authenticationController.signup);
app.post('/api/profile/editPhoto',multipartMiddleware,profileController.updatePh
oto);
app.post('/api/profile/updateUsername',profileController.updateUsername);
app.post('/api/profile/updateBio',profileController.updateBio);
app.post('/api/mean/post',meanController.postMean);
app.post('/api/users/follow',usersController.followUser);
app.post('/api/users/unfollow',usersController.unfollowUser);
app.post('/api/user/removetask',tasksController.removetask);
//app.post('/api/user/task/display_comments',tasksController.displaycomments);
app.post('/api/group/members',groupController.addtogroup);
app.post('/api/task/add',tasksController.addtask);

app.get('/api/user/login',function(req,res){
    res.send('hi hello')});
app.get('/api/profile/editPhoto',function(req,res){
    res.send('hi hello')});


app.listen('3000',function(){
    console.log("listening for localhost 3000");
});
```

Controller code for page after login:

```
/**
 * Created by devi on 8/4/2017.
 */

(function(){
    angular.module('mean-demo')
        .controller('MainController',['$scope','$http','$interval',
        function($scope,$http,$interval){
            $scope.users = [];
            $scope.mygroups = [];
            if(localStorage['User-Data']!== undefined){
                $scope.user=JSON.parse(localStorage['User-Data']);
                console.log($scope.user);
            }

            $http.get('api/group/GetAllNames').
            then(function (response) {
                $scope.users = response.data;});
            $http.get('api/group/GetAllTasks').
            then(function (response) {
                $scope.tasks = response.data;
                console.log($scope.tasks);
            });

            $http.get('api/group/GetAllGroups').
            then(function (response) {
                $scope.mygroups = response.data;});

            $scope.SendMean=function(event){
                if(event.which===13){
                    console.log($scope.user.image);
                    var request={
                        user:$scope.user.username||$scope.user.email,
                        userId:$scope.user._id,
                        userImage:$scope.user.image,
                        content:$scope.newMean
                    }

$http.post('api/mean/post',request).success(function(response){
                        console.log(response)
                        $scope.means=response;
                    }).error(function(error){
                        console.error(error);
                    })
                }
            };

            function getMean(initial) {
                $http.get('api/mean/get').success(function (response) {

if(initial){
    $scope.means=response;
}


else{
    if(response.length>$scope.means.length) {
        $scope.incomingmeans = response;
        console.log('it is working');
    }
}
                    }
                )
            };
            $interval(function(){
                //getUser(true);
                getMean(false);
                if($scope.incomingmeans){
                $scope.difference=
$scope.incomingmeans.length-$scope.means.length;}
                console.log("this is working");
            },5000);
                $scope.setNewMean=function(){
                    $scope.means=angular.copy($scope.incomingmeans);
                    $scope.incomingmeans=undefined;
                }
            $scope.createNewGroup=function () {
                  //---$scope.
            }
            getMean(true);
        }]);
}());
```

I have added the code to github, the github url link is
https://github.com/deviprasad303/svcc-demo

I have not included the node modules in the git hub code.