DbApp

This project was generated with Angular CLI version 8.2.1.

Further help

To get more help on the Angular CLI use 'ng help' or check out the Angular CLI README.

General Layout

The meat of the project is in the tssgDbApp/src/app folder. The 4 top folders are prefaced with an underscore and are named _guards, _helpers, _models, and _services. There is no significance to the underscore character. It just visually separates these folders from the component folders. I will cover these later.

Next are the component folders. These are the primary Angular structures. The 'normal' default for components is a parent folder with the component name containing four files named:

.component.css
.component.html
.component.spec.ts
.component.ts

Each component is 'usually' designed for a specific 'CRUD' function so you will have name-add (create), name-edit (update) and name-get (read/list and delete). As far as angular is concerned, names have no implicit meaning. Meeting-add could also be xyzq-who-cares. However, we attempt to use meaningful names as to what a component is to accomplish. Therefore, looking at the code you will find components for Meeting (add, edit, get, get-schedule), team (add, edit, get) user (create, edit, get-all, get-curr, home, login, logout) and venue (add, edit, get). Meeting, team, and venue are standard. User is standard with additional functions. Get-all will list all users. Get-curr will list the current logged on user. Home is the home landing page. Login provides the logon form and logoff performs a user logoff.
The navbar component creates the navbar used by all the other components. It gets injected into app.component.html. At the bottom of the app folder is the app component. App.component is the root page. All other components are rendered through this page. Also, the app-routing.module.ts file is here. This is where the Angular routing is set up. It looks complicated but once understood it enables easy page to page routing. Finally, we have the app.modules.ts file. An Angular module class describes how the application parts fit together. Every application has at least one Angular module, the root module that you bootstrap to launch the application. You can call it anything you want. The conventional name is AppModule. (https://v2.angular.io/docs/ts/latest/guide/appmodule.html)

Miscellaneous Files and Folders

_guards folder contains the auth.guard.ts file. This implements a JwtHelperService that verifies the logon state during transaction processing. The JWT token is verified and route restrictions based on the logged-on users' role is enforced.

_helpers folder contains error.interceptor.ts, JWT.interceptor.ts and tssg.ErrorHandler.ts. These files provide various middleware error handlers. The JWT.inteceptor inserts the JWT token into transaction headers.

_models folder contains the model for each of the four major components. These map to the schemas declared in the backend code (DbApi/models/*).

_services folder contains services that that connect DbApp to DbApi. Each service makes AJAX calls to the like named backend service. There is a service for each collection (users, teams, meetings, and venues) in addition to a specific service for authentication.

Back out one level to the tssgDbApp/src folder: In the assets folder is the tssg_logo.png file used in the navbar. In the environments folder there are two important files. Environment.prod.ts and Environment.ts. During production builds, angular will replace the environment.ts file with the environment.prod.ts file. If you look in the angular.json file under configurations/production/fileReplacements you will find the declaration for this. This is how we get system environment variables to the angular code. Remember, the frontend code runs on the client browser and does not have access to server environment variables. In the non-production file (environment.ts) we have constants set up for the variables. During development, you must change this file if it is necessary. The typings.d.ts file sets up an interface for these variables that is used by the environment.prod.ts file. Index.html is the root page. Links and scripts are declared here, and the angular rendered pages will be stuffed into this page. At the very top level is the 'dist/' (distribution) folder. It is immediately under the SCHEDULEDB folder. When you do a build for production, the minimized code bundle will be placed in this folder in a sub-folder named DbApp. You can point to this folder or move it to any location you may wish. Then configure your web server (apache, nginx, . . .) to point to that folder. You can also use the npm server if you desire.

Using the database application (DbApp)

These instructions assume that you are working with a newly installed system and that the three main containers (frontend, backend, and mongo-database) are up and running. From your browser, access the frontend via https://:4200. The following logon page should display:

A simple test to see if the backend code is up and running is:

> Enter through browser:
>
> http://(ip or localhost):7010/venues/test - This should return:
>
> globalRoot: D:\TSSG\tssgDbApi - folders: D:,TSSG,tssgDbApi - packageName: tssgDbApi - __dirname: D:\TSSG\tssgDbApi\controllers
>
> Values will be different based on systems and settings. Additionally, any of the four collection names (users, teams, meetings, or venues) may be used in place of the venues collection used in the above example. Each collection has a 'test' command that has unrestricted access and can be used to prove backend code is up and running. It does not connect to the database and therefore does not prove the database path is functional.

To prove the backend/database path is operating, enter:

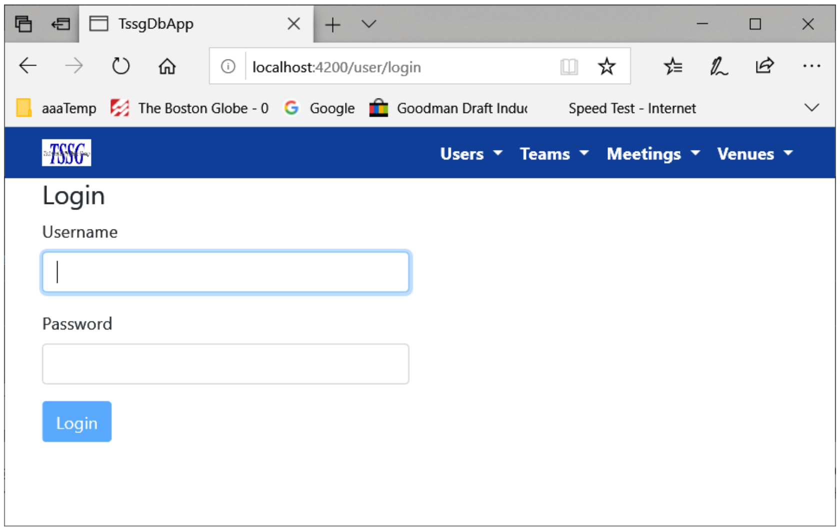
http://(ip or localhost):7010/meetings/webSchedule/WedGenMtg - This should return 0 – 3 meetings for the WedGenMtg team, depending on the current date and time and the meetings existing in the current database. Also, the database must contain a team with the name of WedGenMtg. If you have different 'teams' setup, use one of existing team names. You may also use the literal 'default' in place of the team name. Case is significant.

If no meetings qualify, you will receive an empty array: []

If one, two or three meetings exist, the array will contain the meeting data: […]

Proceed with logging on by connecting to https://localhost | IP :4200

The following logon page should display:

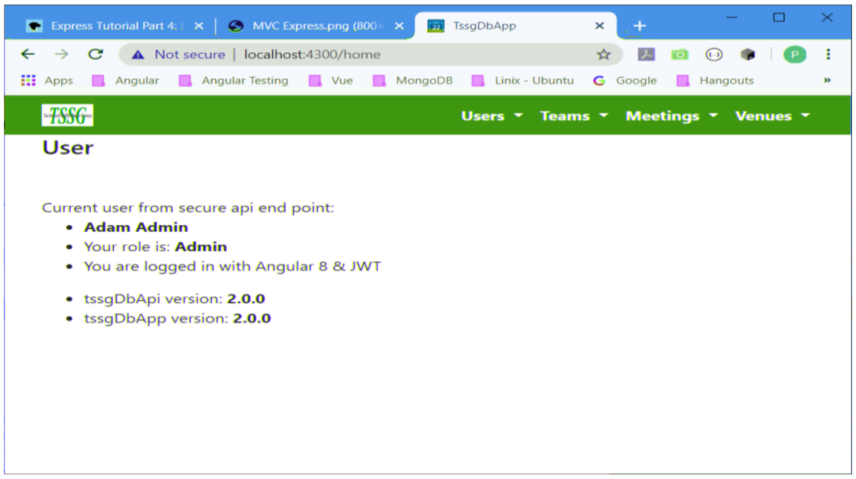


If you are using the raw database supplied with this document, login with the admin account. If other accounts have been added,
login with an admin account. In the startup database supplied with this document, there are two accounts that you can logon with.
Their username and passwords are:

admin/adminpw
user/userpw

The admin account has role admin and user account has role user.

If successful, the user home page will display account information.



User

Current user from secure api end point:

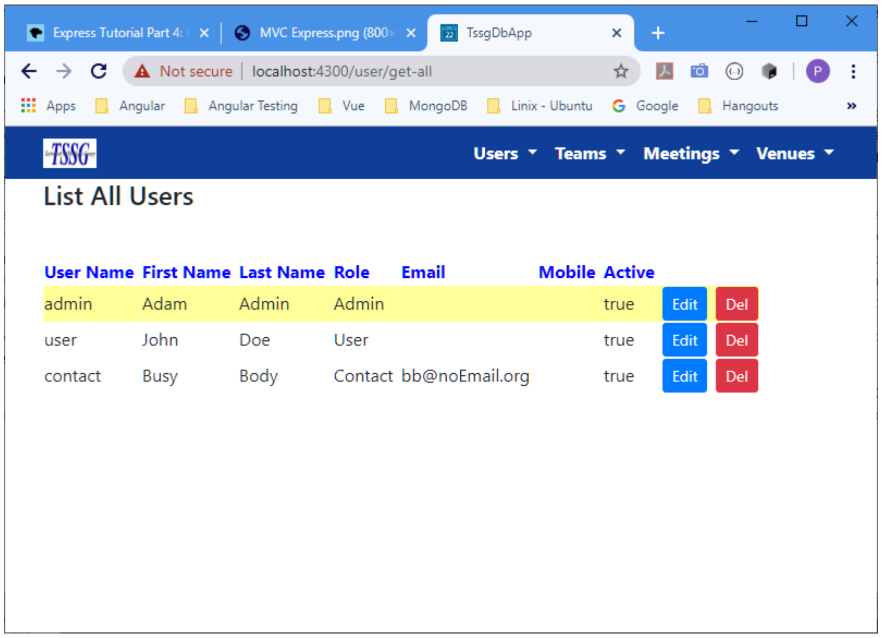
- **Adam Admin**
- Your role is: **Admin**
- You are logged in with Angular 8 & JWT

- tssgDbApi version: **2.0.0**
- tssgDbApp version: **2.0.0**

When a logon is successful, a Java Web Token (JWT) is created and stored in web page headers. The JWT encodes (not encrypts) the username and role. This information is used to authenticate and authorize requests. Because the JWT is not encrypted it is not secure. Using an encryption protocol (TLS/https) provides the necessary security. Currently, the JWT is valid for 2 hours. If you are logged on greater than 2 hours, you will be notified that your JWT has expired when you attempt a request. When this occurs, you are automatically logged off. Simply logon and a new JWT will be created and be valid for 2 hours.

We can have a conversation about the JWT validity length of 2 hours. It's just a value I chose that seemed to be reasonable. There are also methods to consider as to how to change this value as it currently requires a coding change. It wouldn't take much to convert it to a system variable. Something to consider about the JWT validity length is that the browser maintains the JWT for the remainder of the two-hour period until you log off or close the browser. This is browser dependent. If you are logged on and kill the browser, Chrome and Firefox will maintain the JWT and Edge will clear it. This means that if you are logged on with Chrome or Firefox and kill the browser, anyone using your machine can restart the browser and be logged on to tssg-tech. Convenient for developers, bad for security.

From the user's dropdown you can add a user, list all users, list current user, login and logout. From the list all users, you can edit accounts or delete them. List current user lists the user that is currently logged on. This user can edit their account but not change their role. Also, they cannot delete their account. If a logged-on user does not have the role admin, the only options available are logout or list current. Again, these are choices I made, and we can certainly make changes.



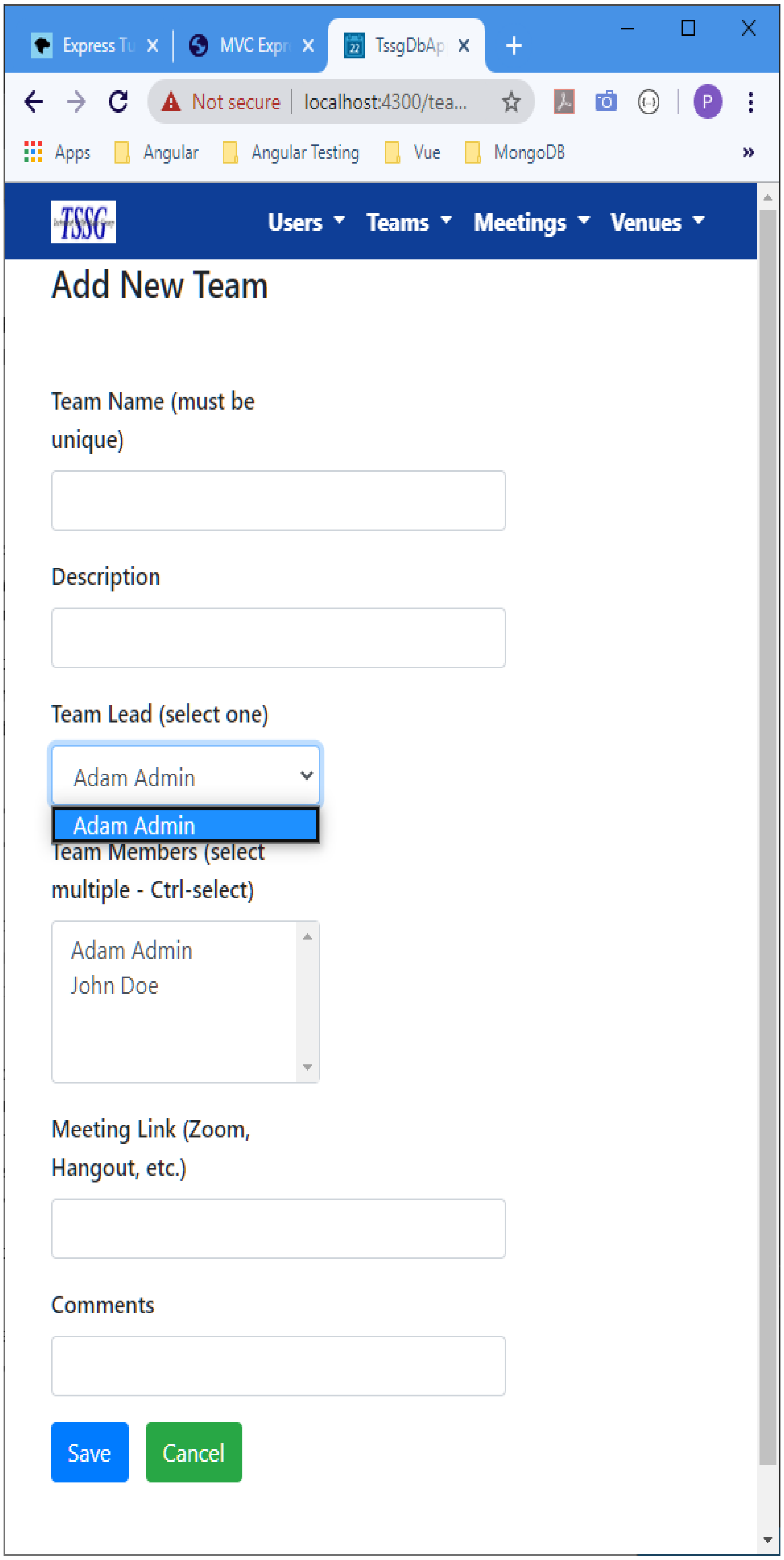
List All Users

User Name	First Name	Last Name	Role	Email	Mobile	Active		
admin	Adam	Admin	Admin			true	Edit	Del
user	John	Doe	User			true	Edit	Del
contact	Busy	Body	Contact	bb@noEmail.org		true	Edit	Del

Because this is an initialized system, spend some time familiarizing yourself with the navbar.

Add some users. They can be test or real. It would help to have at least two each with role admin, user, and contact. It is important to add users first as teams and venues require user entries. To add, edit or delete users, you must be logged on with an admin account.

Currently, the main purpose of this application is to schedule meetings for the Wednesday TSSG meeting. Meetings are scheduled for a team at a venue for a date and time. Therefore, to create meetings you must have teams and venues. The initialized database contains a few venues which are sufficient for this exercise. The initialized database also has a few teams, so no action is needed there.
If one is creating a team, the team lead selection is limited to users with role admin. The team members selection will display all users with role admin or user but not contact. Choices are limited to what is in the database which is why it helps to add a few users for each role.
Create one or more teams.



Express Tu × MVC Expr × TssgDbAp × +

Not secure | localhost:4300/tea...

Apps Angular Angular Testing Vue MongoDB »

TSSG Users ▾ Teams ▾ Meetings ▾ Venues ▾

Add New Team

Team Name (must be unique)

Description

Team Lead (select one)

Adam Admin

Adam Admin

Team Members (select multiple - Ctrl-select)

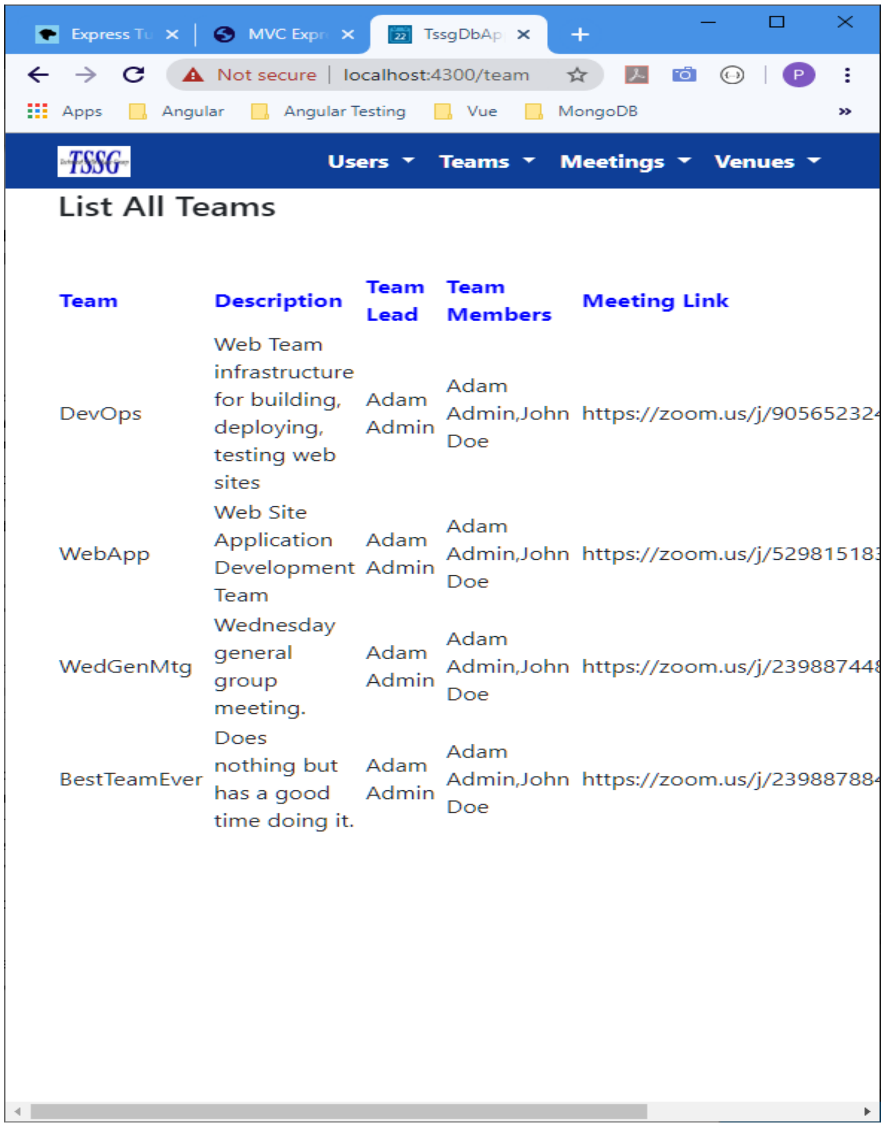
Adam Admin
John Doe

Meeting Link (Zoom, Hangout, etc.)

Comments

Save Cancel

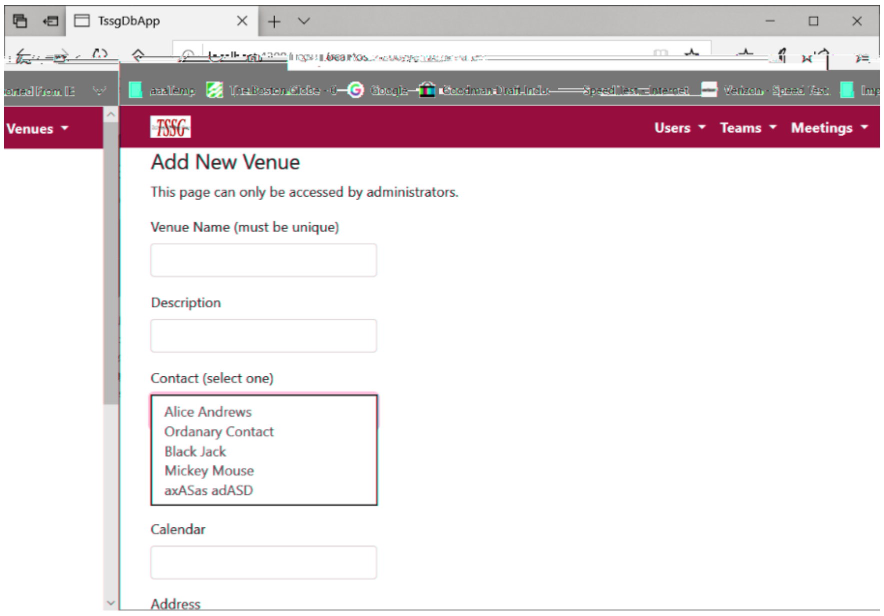
Select 'Save' and a list of teams containing the new team will display.



List All Teams

Team	Description	Team Lead	Team Members	Meeting Link
DevOps	Web Team infrastructure for building, deploying, testing web sites	Adam Admin	Adam Admin,John Doe	https://zoom.us/j/90565232
WebApp	Web Site Application Development Team	Adam Admin	Adam Admin,John Doe	https://zoom.us/j/52981518
WedGenMtg	Wednesday general group meeting.	Adam Admin	Adam Admin,John Doe	https://zoom.us/j/23988744
BestTeamEver	Does nothing but has a good time doing it.	Adam Admin	Adam Admin,John Doe	https://zoom.us/j/23988788

If necessary, add one or more venues.



Add New Venue

This page can only be accessed by administrators.

Venue Name (must be unique)

Description

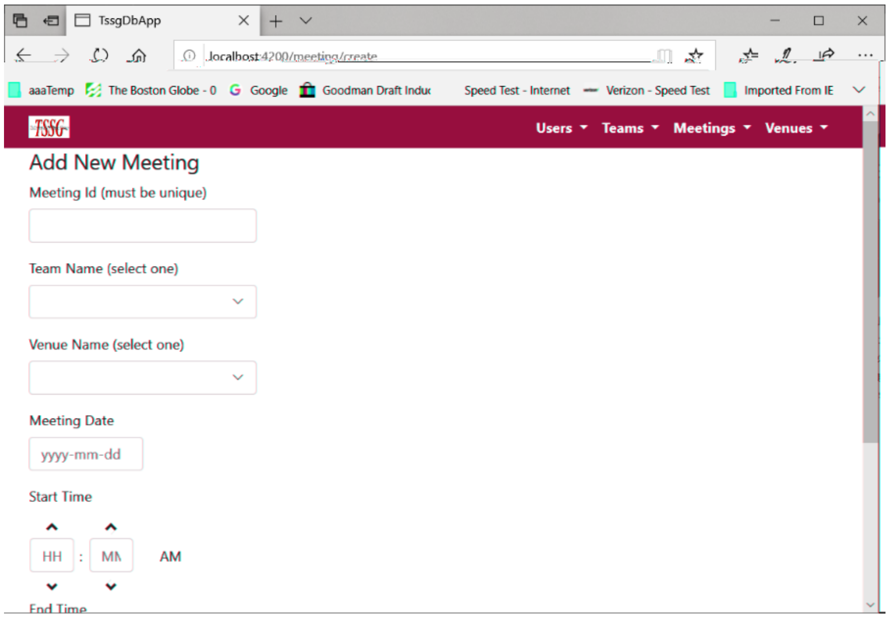
Contact (select one)

Alice Andrews
Ordanary Contact
Black Jack
Mickey Mouse
axASas adASD

Calendar

Address

Now you can create meetings. Meetings require both team and venue entries.



Add New Meeting

Meeting Id (must be unique)

Team Name (select one)

Venue Name (select one)

Meeting Date

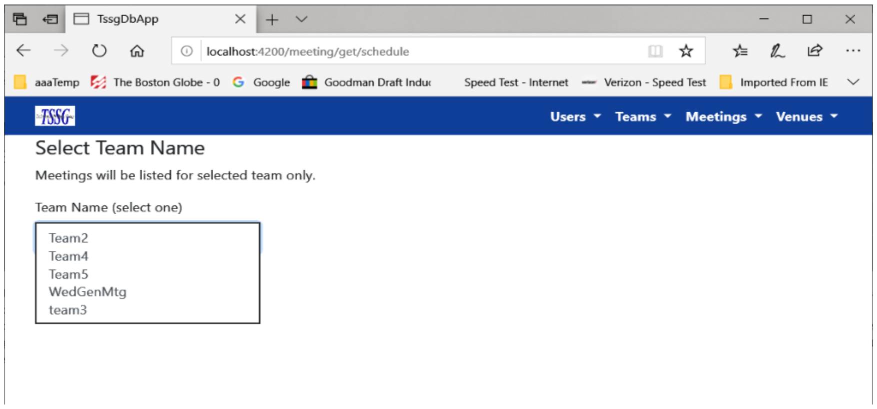
yyyy-mm-dd

Start Time

HH : MM AM

End Time

Select Meetings/List Team Meetings. The following page will display:



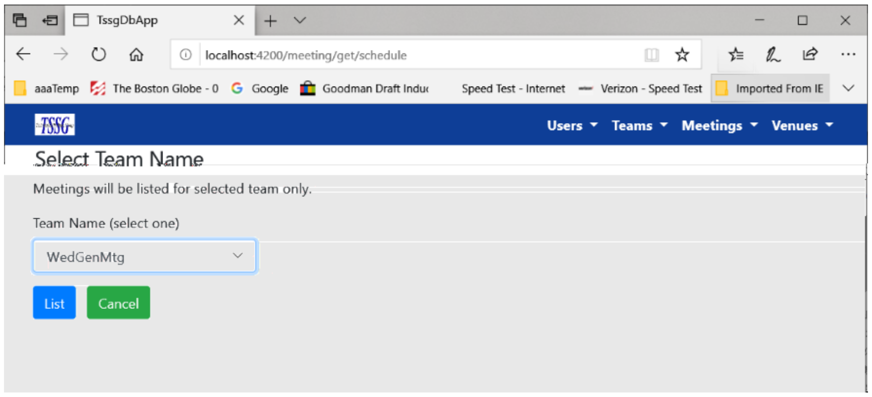
Select Team Name

Meetings will be listed for selected team only.

Team Name (select one)

Team2
Team4
Team5
WedGenMtg
team3

Select a team name



Select Team Name

Meetings will be listed for selected team only.

Team Name (select one)

WedGenMtg

Select List. The scheduled (0, 1, 2, or 3) meetings for that team will display.
To qualify, a meeting date must be >= today's date and for
meetings for today's date the meeting end time must be > current time.

You can also run a test outside of the frontend code. This is described at the beginning of the above 'Using the database application (DbApp)' section.