**Documentation on how to run the server**

**Author**: Kyle Ahn

!! Updated on Feb 11, 2018 !!

Currently, the master branch contains a working MVP that is ready for the first snapshot week. “Dist” file that is necessary to run the server on EC2 is already up.

**Running the project on a local machine**

1. A working MVP is in the master branch. You could easily clone it to your local machine.
2. After cloning the master branch, you would need to have node\_modules.
   1. First, install Node.js from <https://nodejs.org/en/>.
   2. After downloading and installing Node.js, you could confirm that Node.js and npm are installed by running,

$ node –v

$ npm –v

* 1. These commands should give you v8.9.4 for node and v5.6.0 ( for the current development environment ).

1. Once you have your npm and Node.js set up on your local machine, please go to ‘umbuy\project\umbuy\web>’ directory where you could see ‘package.json’.
2. Run ‘$ npm install’ to install the necessary files such as node\_modules which contain ‘express’, ‘mysql’, ‘aws-sdk’ and etc.
3. If you have successfully installed everything, you should be able to run the server on localhost:4200 by now.
   1. You could run the following command to easily run the project on localhost. Please make sure you run this command on ‘web’ directory.

$ ng serve

* 1. You can run Karma testing by running,

$ ng test

* 1. You can build your own dist file by running,

$ ng build –prod

1. Once you are able to run ‘ng serve’ successfully, you can now get MySQL database ready.

**Set-up the Database locally**

Requirements:

Microsoft .NET Framework 4.5

Microsoft Visual C++ 2015 Redistributable Package

1. Download msi installer from http://dev.mysql.com/downloads/windows/installer/
2. Install the msi installer and follow the steps
3. Choose the Standalone MySQL Server and select port 3306
4. If MySQL Workbench did not get installed, you can download it at: http://dev.mysql.com/downloads/workbench/ and install it.
5. MySQL Workbench can be installed using the Windows MSI Installer package. The MSI package bears the name
6. mysql-workbench-community-version-winarch.msi, where version indicates the MySQL Workbench version number,
7. and arch the build architecture (winx64).
8. For us, the MySQL Workbench was located at: C:\Program Files\MySQL\MySQL Workbench 6.3 CE. By default, it should install everything for MySQL under
9. C:\Program Files\MySQL. Go to the directory where MySQL Workbench is located.
10. Open MySQLWorkbench.exe
11. Click File tab -> Open Model
12. Select the database.mwb from our project at umbuy\project\umbuy\web\src\app\persistence
13. That should import our databases into your local MySQL Workbench for you to see.
14. We need to synchronize our database to localhost to be able to connect to it through express. Click Database tab -> Synchronize model
15. The only thing you need to change is the user name and password to match the user name and password you entered when you installed MySQL Server in step 2-3
16. Continue to push next until the dialog closes. Now your database should be running locally, and you should be able to connect locally.
17. You need to go to localhost.js in umbuy\project\umbuy\web\src\app\persistence and update:

var connection = mysql.createConnection({

host: ‘127.0.0.1’,

user: 'yourLocalMySqlUsername',

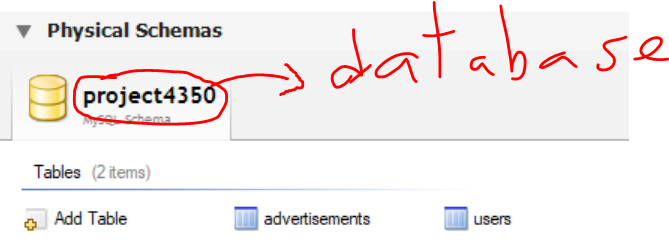
password: " yourLocalMySqlPassword",

database: 'project4350',

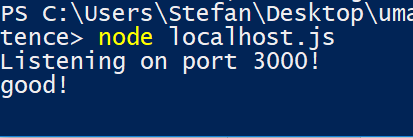
port: '3306'

);

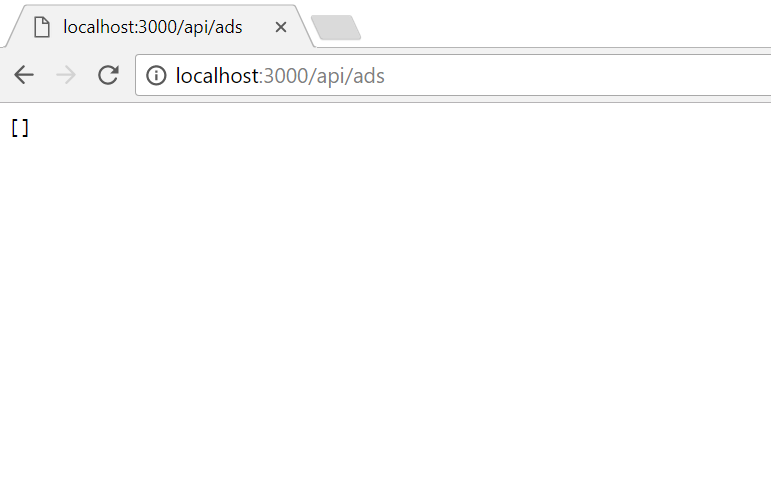
1. 127.0.0.1 means localhost, change the user and password fields to match the username and password you chose for MySQL, the database name is shown in the screenshot below and port is 3306 which is the default. If you change the port while setting up MySQL, make sure to change port to that port number.



1. Now from the directory where localhost.js is located (umbuy\project\umbuy\web\src\app\persistence), in command line type: node localhost.js. You should get a message like below:



1. Now, you can test it out and refer to umbuy\project\umbuy\web\documentation\http\_api\_documentation.docx
2. **Important Note: You need to run node localhost.js and ng serve at the same time ( therefore 2 different command prompts or powershells).**
3. **CHECK: To make sure you set the database up correctly, do node localhost.js and then once you see the message as above, open your browser and type localhost:3000/api/ads and you should get [] since there is no data locally at first. If you have data from the previous snapshot, then it’s fine if you get data back as long as there is no error in the command line or browser. It should look like below if there is no data:**



**Running the project on AWS EC2.**

1. First, you need to complete the steps for “**Running the project on a local machine**” because you need “dist” directory to run the project on the server.
2. You could Ubuntu to connect to the server.
   1. Have your public key ready.
   2. Open your terminal and go to the directory where you have saved your public key.
   3. Run ‘chmod 400 KyleKeyValid.pem’ to have your public key recognised.
   4. Connect to the instance using its public DNS
      1. ssh -i "KyleKeyValid.pem" [ubuntu@ec2-18-217-86-148.us-east-2.compute.amazonaws.com](mailto:ubuntu@ec2-18-217-86-148.us-east-2.compute.amazonaws.com)
   5. login id is ‘ubuntu’
3. After connecting to the server successfully, go to ‘/home/ubuntu/deployment’. If you run ‘ls’, then you should be able to see ‘server.js’ file. In the same directory, please transfer your previously created ‘dist’ folder.
4. Run ‘node server.js’ to start the service. You could access the website at ec2-18-217-86-148.us-east-2.compute.amazonaws.com:9000.
   1. You should see ‘CONNECTED’ message in the console.

**Accessing MySQL on the server.**

1. If you have successfully connected to the server and are able to run ‘node server.js’, then you are able to connect to MySQL on the server as well.
2. Please run ‘mysql –u kyle –p’ to log into MySQL. The password is ‘team6best’.
3. Please use ‘sampledb’ as the database by running ‘USE sampledb;’.
4. You could see the tables, ‘advertisements’ and ‘users’, by running ‘SHOW TABLES;’ and you could see the definition of these tables by running ‘DESCRIBE <table\_name>;’.