

# Debugging Client – Server interactions

Aditya Emani, Graduate Student, Texas A&M University, College Station, TX

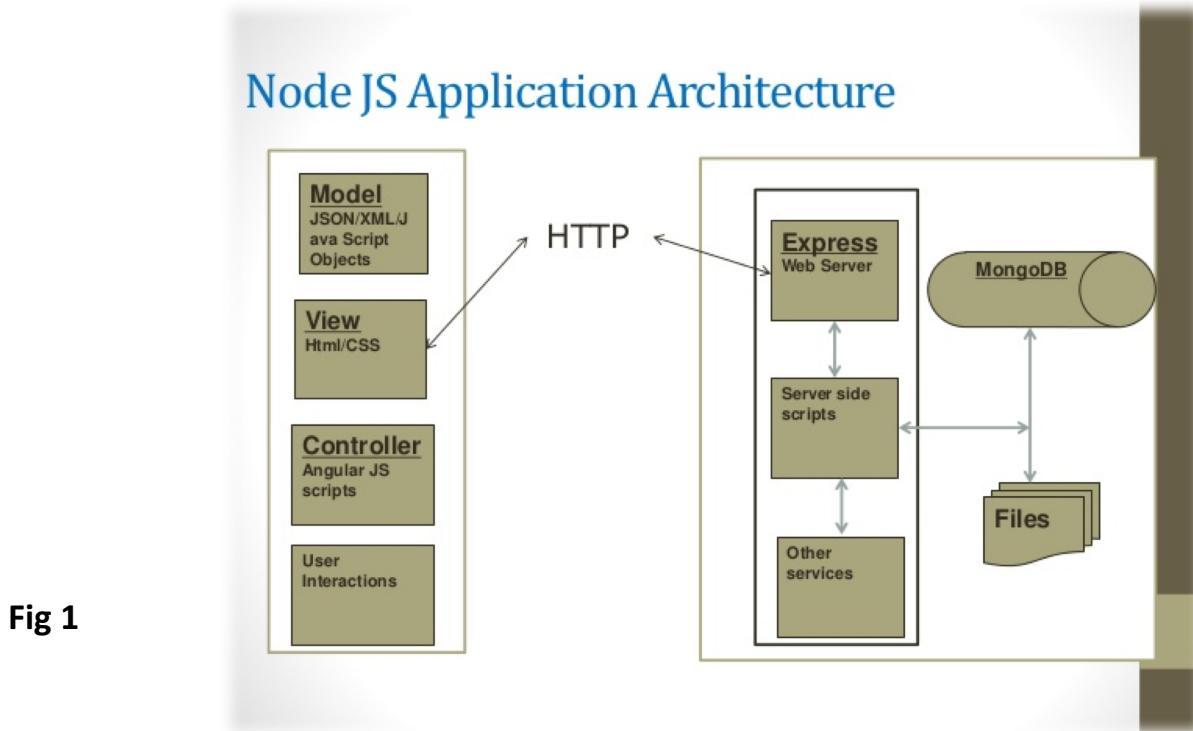
Email: [prasaadem@tamu.edu](mailto:prasaadem@tamu.edu)

April 05, 2017

**Abstract:** Understanding client-server interactions always plays a crucial role in designing an efficient and reliable interface to enable data transmission over internet. Overlook in designing the server can grant users special privileges resulting in huge loss. I had worked on this problem in the past two months and designed a server along with a few API requests to help me understand more. This article discusses challenges I faced in designing server and summarizes bugs in 4 directions: Server, Network, Client, Database.

## Architecture:

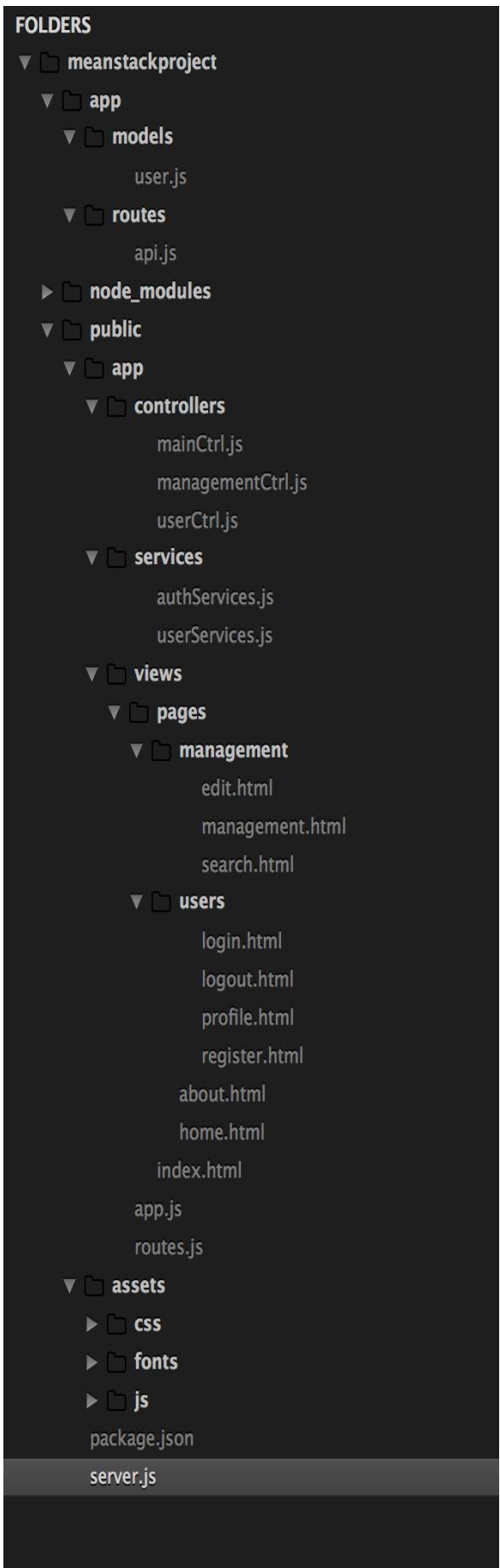
The project architecture is as shown in Figure 1.



**Client** application showed in fig 1 is designed using basic JavaScript, html and Angular.JS  
**Server** is designed using Express.js and Node.js  
**MongoDB** is used as a database for this project.

## Project Tree Structure:

- User.js is the user modal for the database. User credentials will be stored in this modal.
- Api.js file consists of all the server API
- Controllers implement sessions and user interactions with the views.
- Services implement authentication and user access with routes
- Management views contain all the admin access to the pages
- User Views contain all the public access to the pages
- Routes.js implements all the routes in the application
- Server.js runs the server.
- Package.js has all the packages required for the application.



## **Issues encountered**

**Case 1:** Front-end should restrict the access of management page from the public usage in routes file.

**Failure:** Anyone without a login credential can access this page by typing the url in the browser.

**Case 2:** Any tokens created for the user on the client side should be destroyed when the user logout.

**Failure:** If the token is not deleted, users home page will be displayed when he opens the login page.

**Case 3:** Validation of the token must be done at every login.

**Failure:** A user is logged in to the application which creates a token in his browser. Now the admin deletes this user from the database. The user can still log in to the application using his token.

**Case 4:** All the databases used for the application must be in sync all the time.

**Failure:** Consider having two databases for the application, the administrator modifies one of the moderator to user. This reflects in database 1. But the user now accesses the application and he is accessing database 2. He will be logged in as moderator.

**Case 5:** Server should restrict the access of management page from user portal in routes file.

**Failure:** User can view the management page url in his home page.

**Case 6:** Server should restrict user from the access of management information from management portal in management controller.

**Failure:** User can view the management information in management page.

**Case 7:** A moderator should be restricted from editing the user and admin in the server.

**Failure:** Moderator can edit himself to be an admin.

**Case 8:** Special limitations should be made for poor network.

**Failure:** User can view the management page url in his home page for few seconds before the page loads properly.

**Case 9:** Server should restrict the access of management page from user portal in routes file.

**Failure:** A logged in user can view the management page by typing the url in the browser.

**Case 10:** While creating users, server should check if the username is already taken.

**Failure:** Failure to meet this condition will result in user logging in as a different user.

## **Results:**

All the above mentioned issues are fixed and the final project is stored in a github repository.  
For further information, <https://github.com/prasaadem/meanstackproject.git>

## **References:**

<https://nodejs.org/en/>

<https://www.tutorialspoint.com/nodejs/>

<https://github.com/expressjs/body-parser>