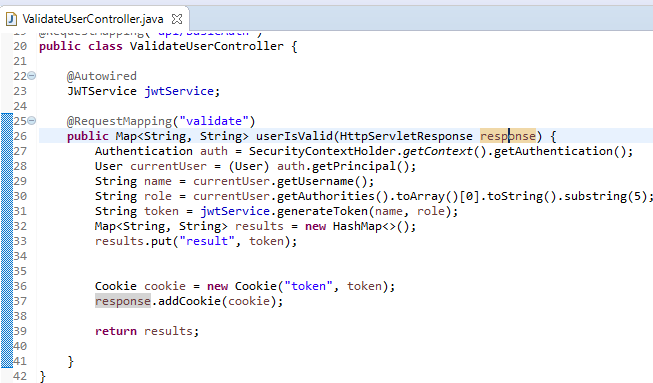
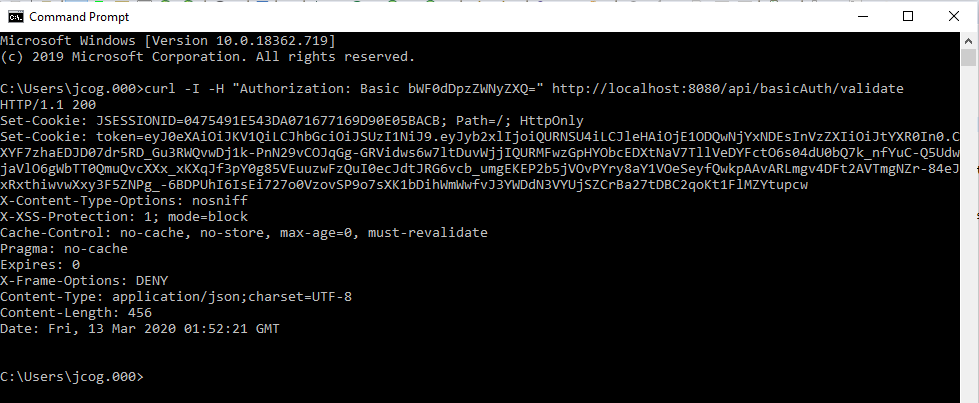
Setting up a JWT cookie:

Create a cookie on the server:



We can test this with curl:



The Cookie section:

C:\Users\jcog.000>curl -I -H "Authorization: Basic bWF0dDpzZWNyZXQ=" http://localhost:8080/api/basicAuth/validate

HTTP/1.1 200

Set-Cookie: JSESSIONID=0475491E543DA071677169D90E05BACB; Path=/; HttpOnly

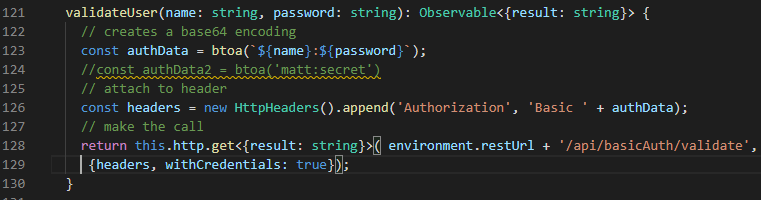
Set-Cookie: token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJyb2xlIjoiQURNSU4iLCJleHAiOjE1ODQwNjYxNDEsInVzZXIiOiJtYXR0In0.CXYF7zhaEDJD07dr5RD\_Gu3RWQvwDj1k-PnN29vCOJqGg-GRVidws6w7ltDuvWjjIQURMFwzGpHYObcEDXtNaV7TllVeDYFctO6s04dU0bQ7k\_nfYuC-Q5UdwjaVlO6gWbTT0QmuQvcXXx\_xKXqJf3pY0g85VEuuzwFzQuI0ecJdtJRG6vcb\_umgEKEP2b5jVOvPYry8aY1VOeSeyfQwkpAAvARLmgv4DFt2AVTmgNZr-84eJxRxthiwvwXxy3F5ZNPg\_-6BDPUhI6IsEi727o0VzovSP9o7sXK1bDihWmWwfvJ3YWDdN3VYUjSZCrBa27tDBC2qoKt1FlMZYtupcw

X-Content-Type-Options: nosniff

Sending the cookie from Angular:

Add withCredentials: true to the headers option in both the getRooms() and the validateUser()

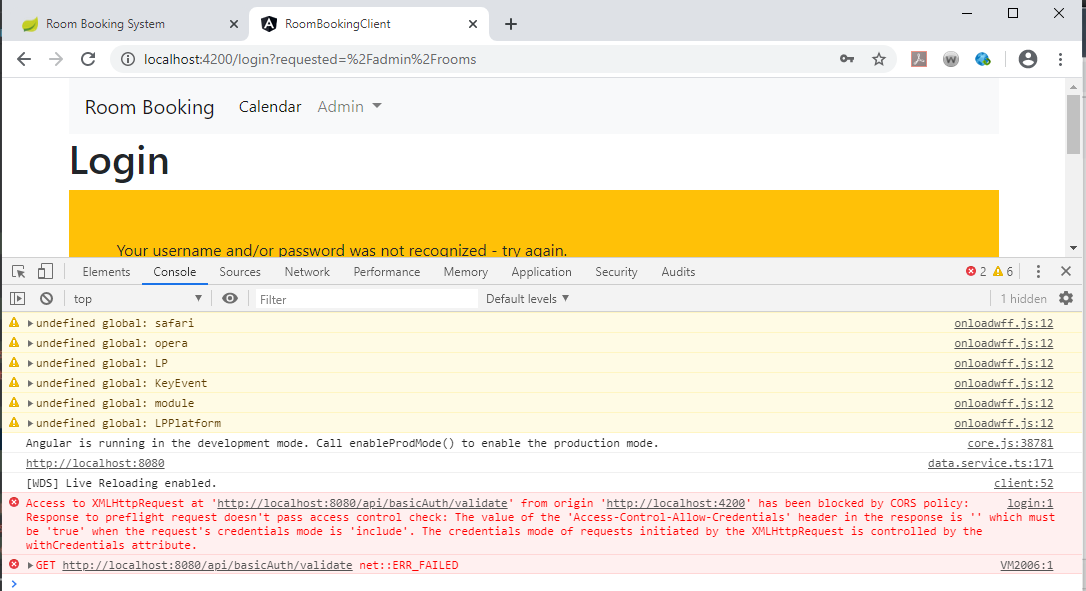
in the data.service.ts file.



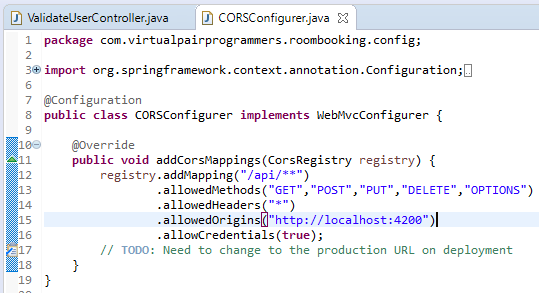


Cookies will now get sent to the server when the REST calls are made.

Test the cookie sending capability:

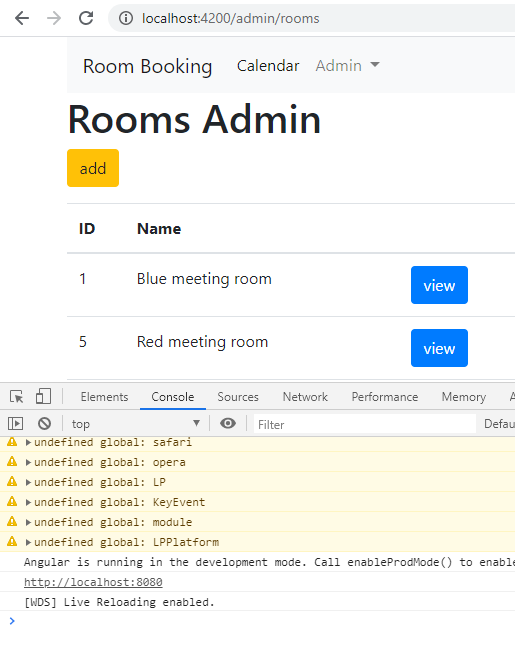


Notice that the withCredentials is blocked by CORS



Adding the allowCredentials(true) to the addCorsMappings will remove the block.

Test Admin > rooms > login



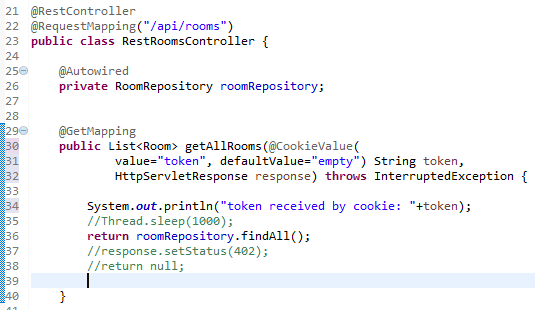
No errors this time

Set the Browser for site settings and look for local cookies:

Setttings >> Search for ‘site settings’ >> expand the arrow >> cookies and site data >> secondary search ‘localhost’ >> expand the arrow >> token

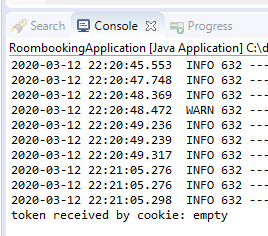


Let us print out the returning cookie on the server:



Restart the server:

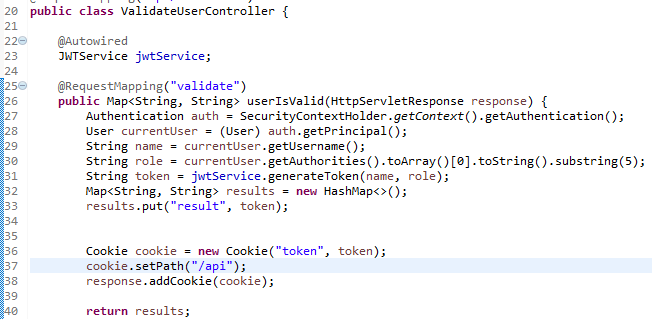
Then test again, then check the server console for the result:



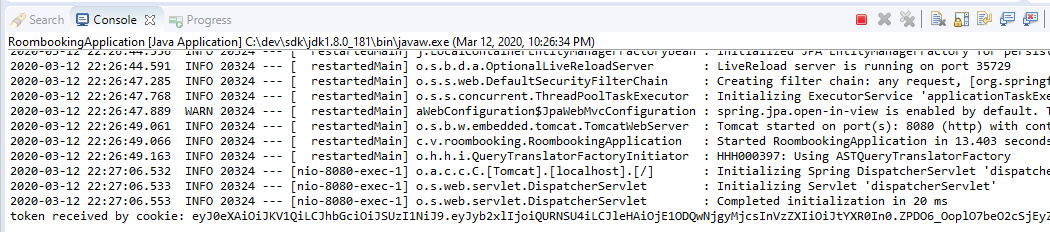
Well there is a problem:

The Path for sending back was /api/basicAuth and we are in the wrong place to receive the cookie.

Now, we should change the path that is used for sending the cookie back.

  
Notice the path is set to /api

Test again:

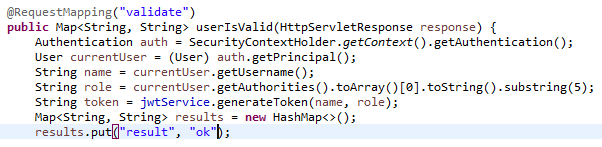


Now that better:

**Using the cookie for authentication:**

Now the cookie is going back and forth successfully.

Let us replace the token with “ok” in ValidateUserController.java class.



Refactor the JWTAuthenticationFilter to use a cookie:

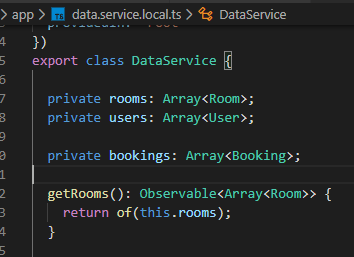


Now, change the angular code to use cookie and not a Bearer:

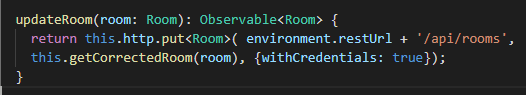
Removed headers and Bearer:



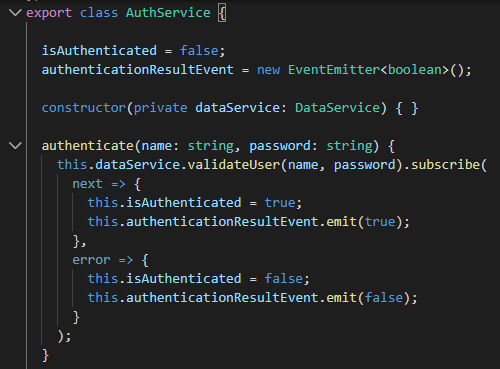
Go to the other method and the data.local.service.ts (no changes here )



The other method in the data.service.ts file is updateRoom()

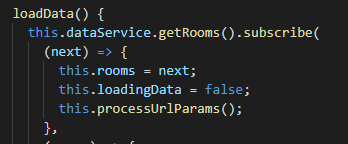


Now, the auth.service.ts file. Remove jwtToken lines.



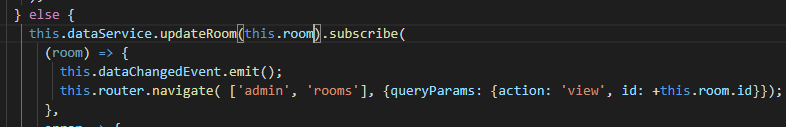
Now, the rooms.component.ts file

Remove jwtToken



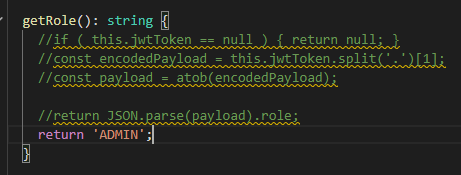
Now, the rooms-edit.component.ts file

Remove jwtToken



Now, back in the auth.service.ts file

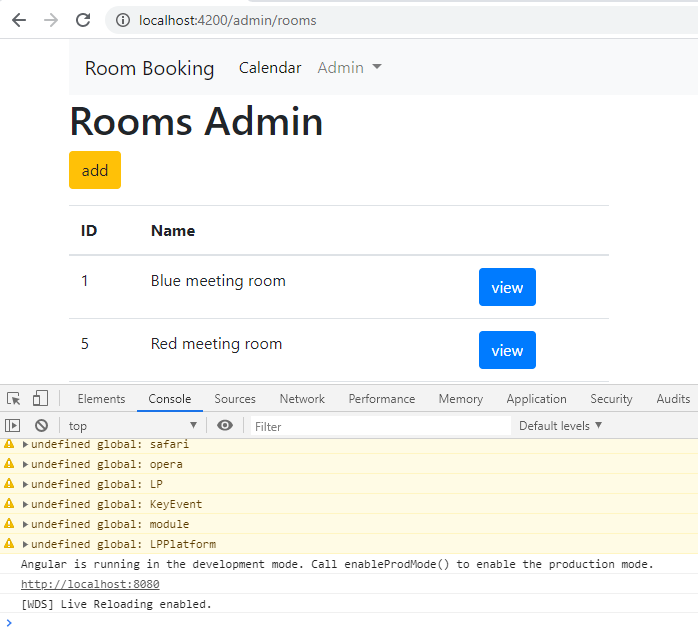
Comment out the body and add a return ‘ADMIN’;



Test this:

Login > Admin > rooms

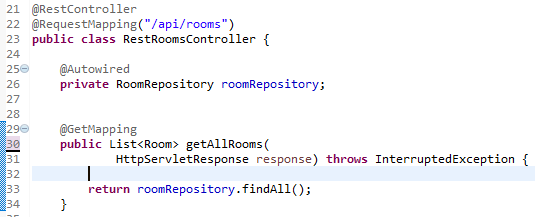
See if you get access!!



It works

**Securing the cookie and getting the users role:**

In the RestRoomsController we can remove the println statement:

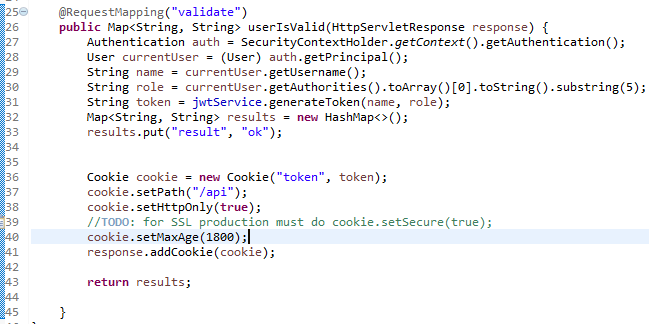


Now, go to the ValidateUserController:

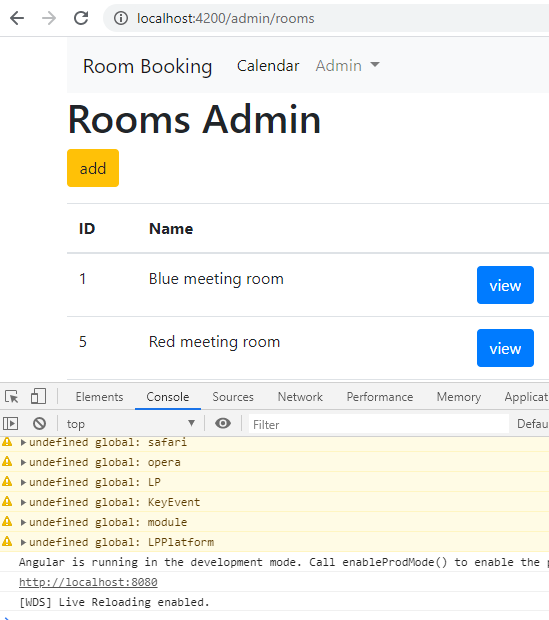
Add the HttpOnly to remove javascript access

Add a maxAge for the cookie expiring the same as the token

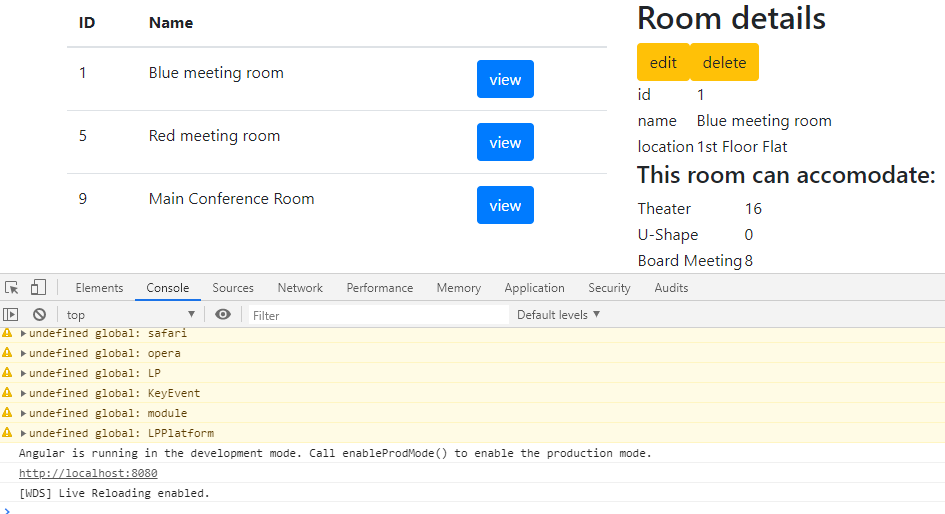
Add but comment out for testing the secure(true) so, the cookie can be used with SSL



Test the login and the Edit capability:



**Login works, let us check the Edit – First Floor Flat**

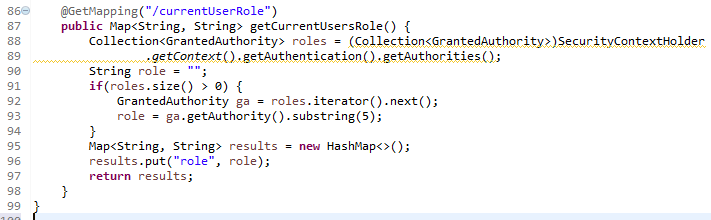


**It works**

Return to the auth.service.ts file:

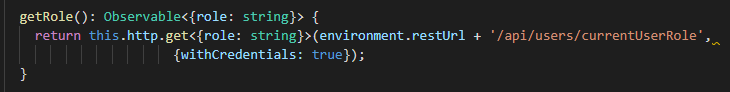
We cannot ask the token for the role, so we have to get from a server call.

Go to RestUsersController and add a method for roles:

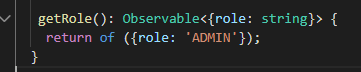


Now, call the method from angular

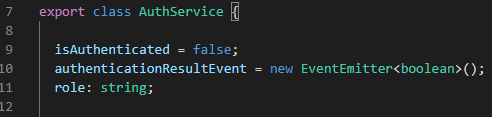
Add to the data.service.ts file the method getRole()

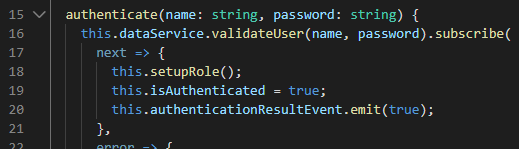


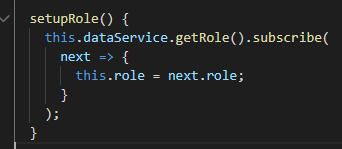
Update the data.local.service.ts file with getRole()



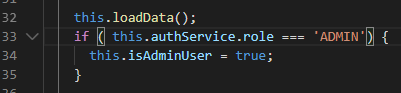
Now, the call in the auth.service.ts file



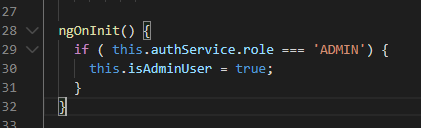




Now, replace getRole() with the property role in the rooms.component.ts file

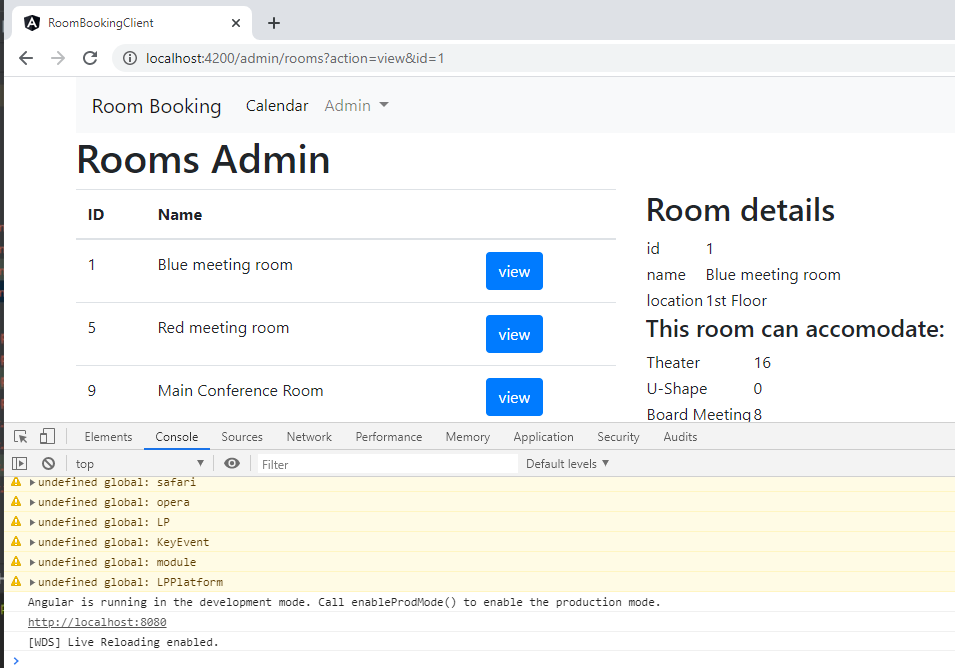


Replace getRole() with the property role in the room-details.component.ts file

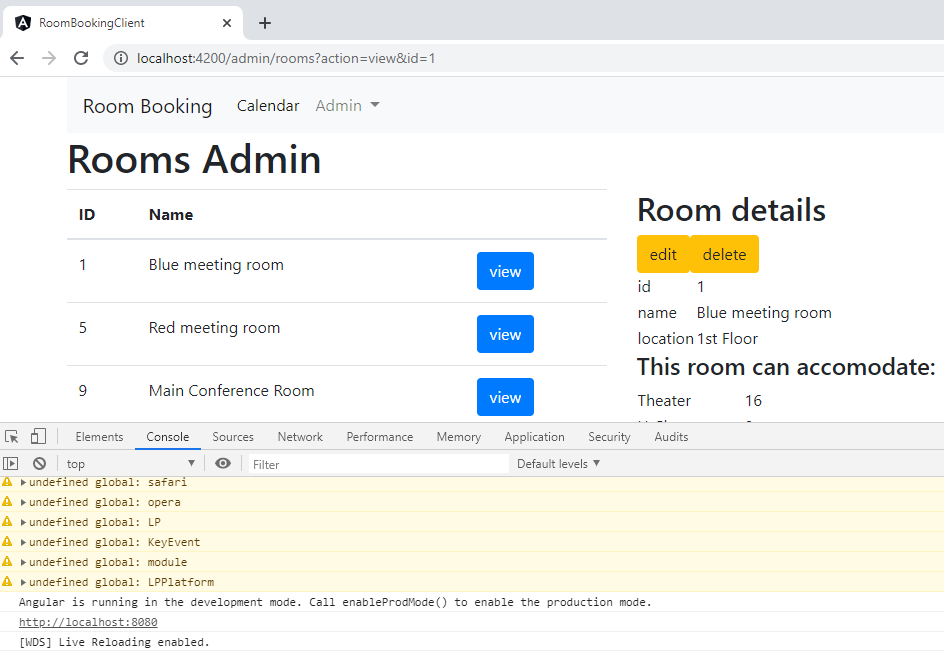


Now, test with user jane and then matt

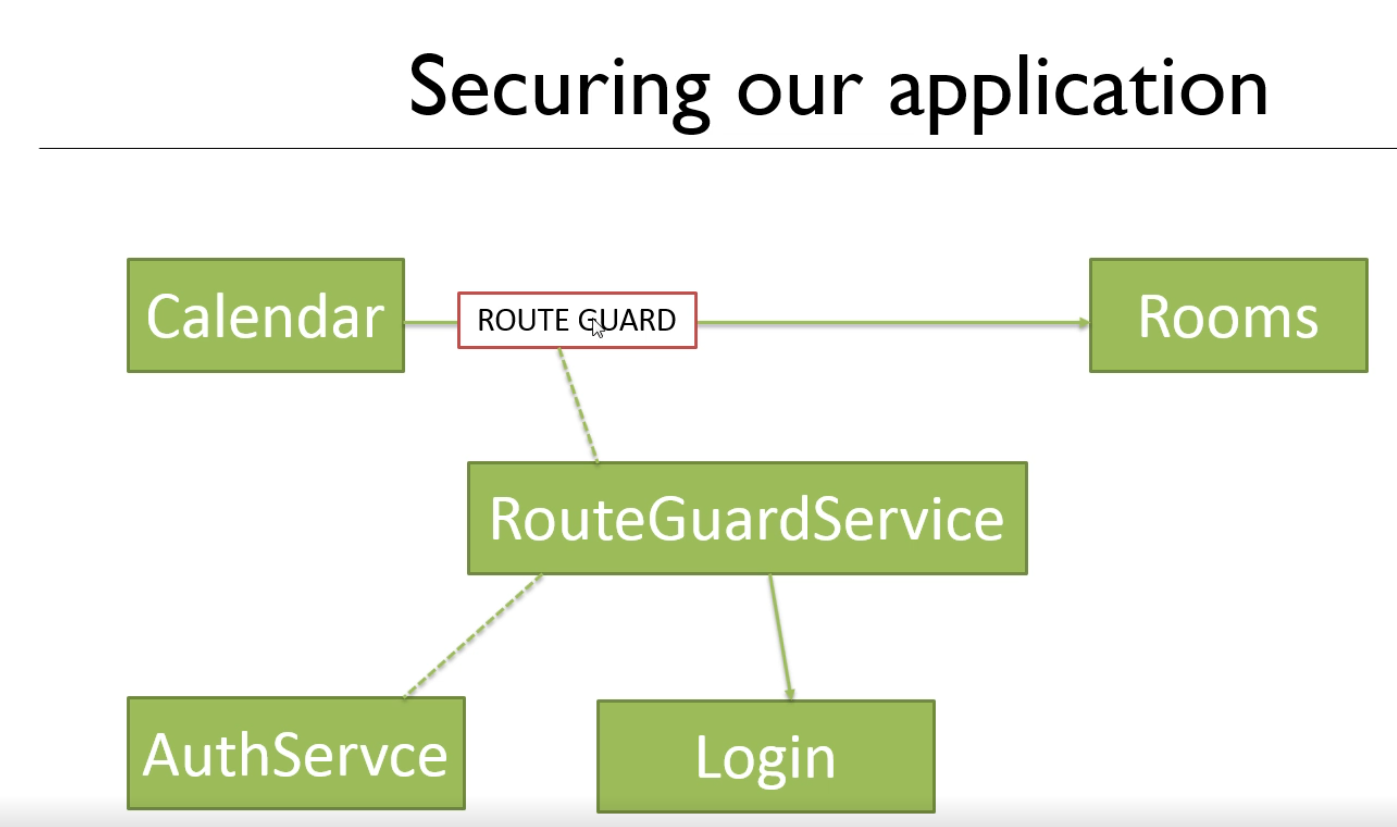
Login as jane:



Login as matt



**Surviving the REFRESH:**



Check if the user already authenticated: