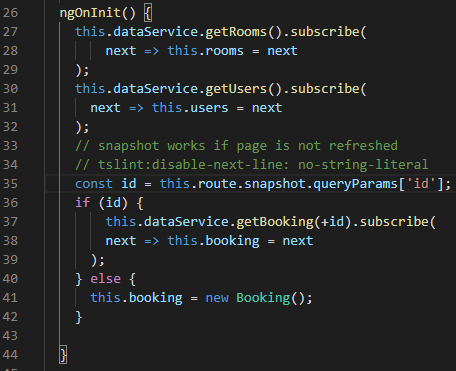
Prefetching Data:

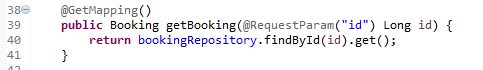
Implementing Editing Bookings:

Add and Edit Functionality

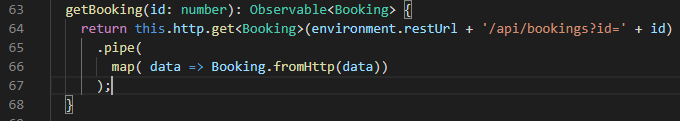
In the edit-booking.component.ts class



In the Spring API RestBookingsController.java

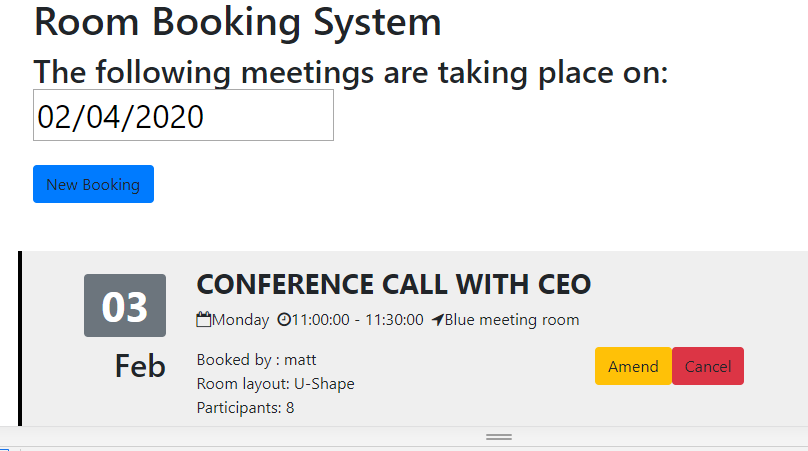


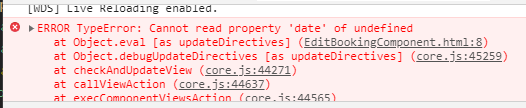
Now, go to the data.service.ts class



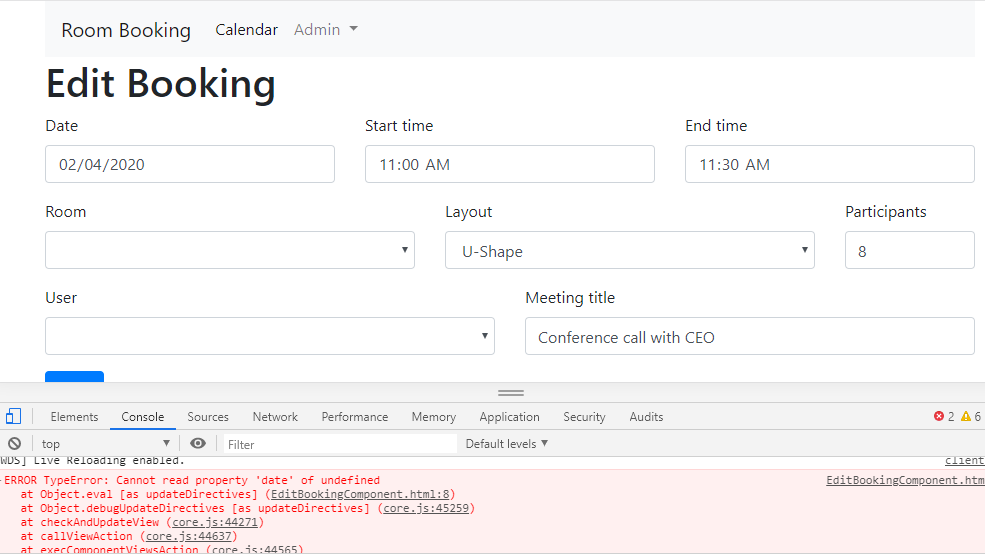
Now, let us look at the UI for Editing a Booking

You need to select the Amend button to edit:



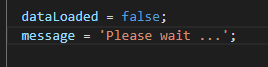


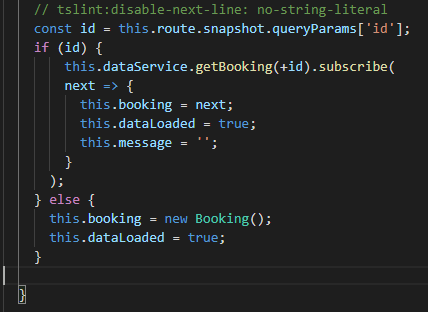
The Room and User are not preloading:



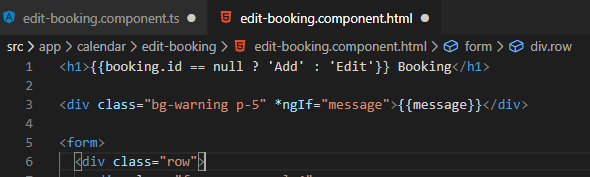
Let us not display the date until it is loaded:

Go into the edit-bookings-component.ts class





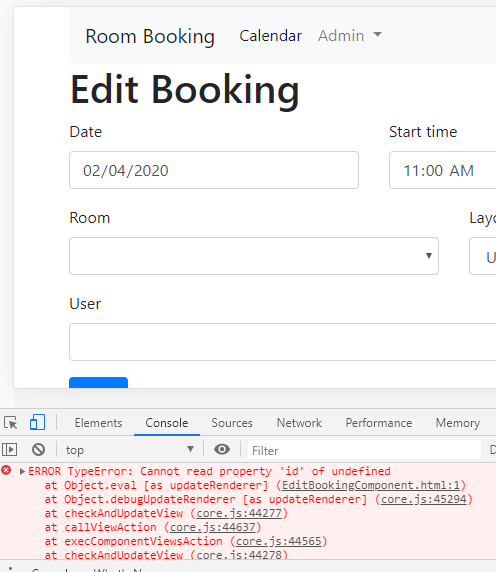
Display the message:



Do not show the form if the data is not loaded



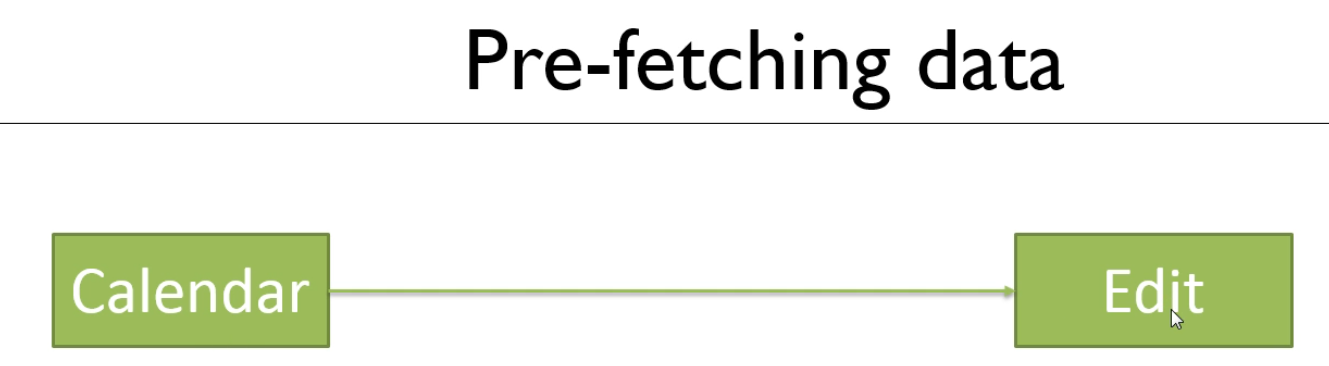
Now let us see if this helps the loading



Still got some errors

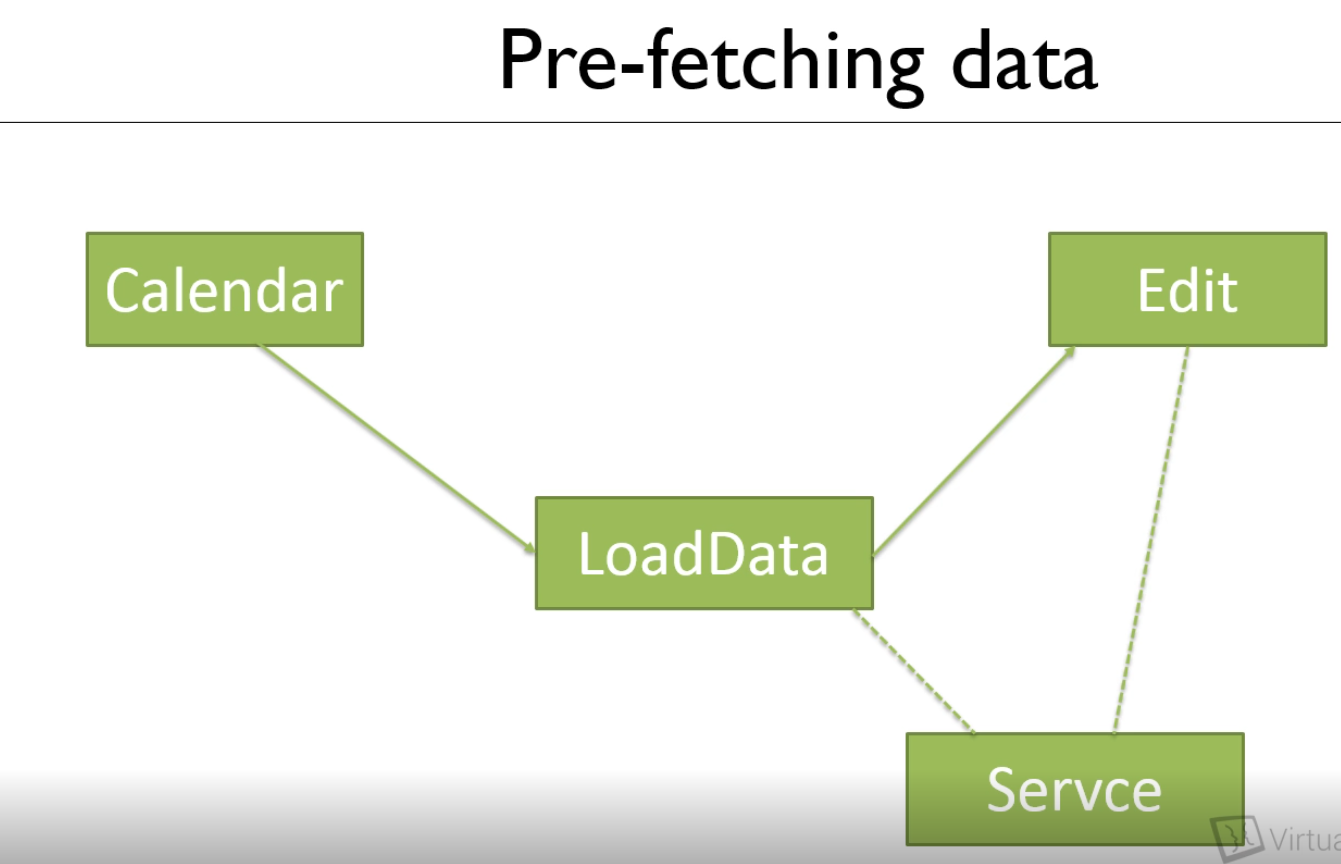
The Booking data is available before the User and Room

Let us Pre-fetch the data – non production method first



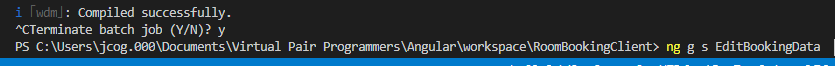
Instead of doing the above we are going to use an intermediate component:

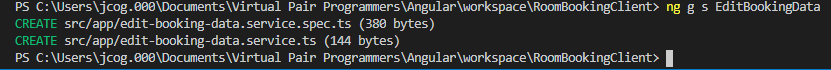
LoadData component



Create a new component and a service

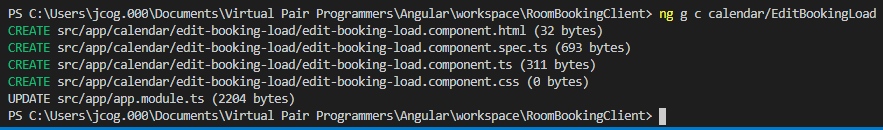
Angular ng, Generate g, Service s EditBookingData <Service> is added automatically



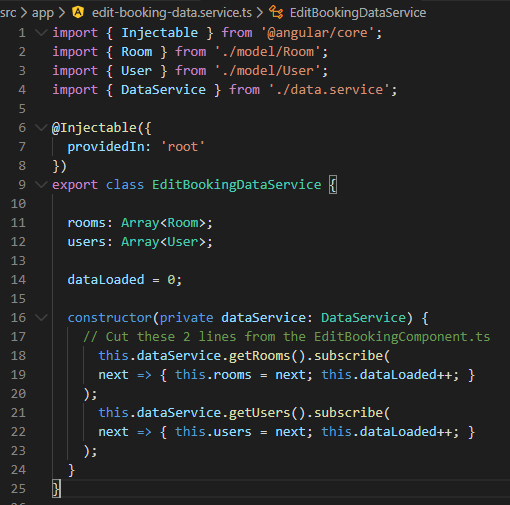


Angular ng, Generate g, Component c under Calendar calendar/EditBookingLoad

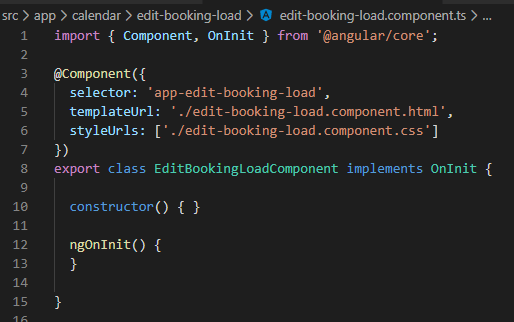




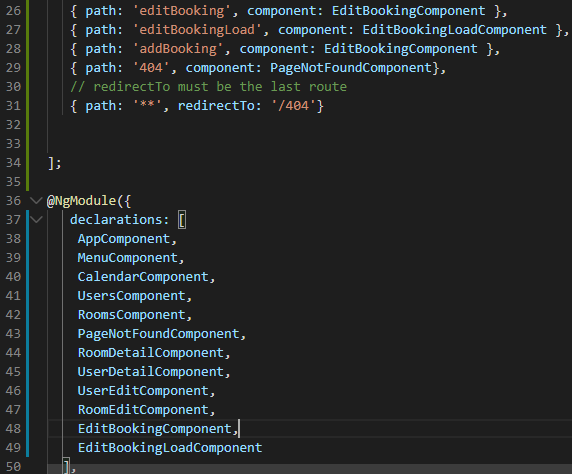
Code the Service first



Now the Component

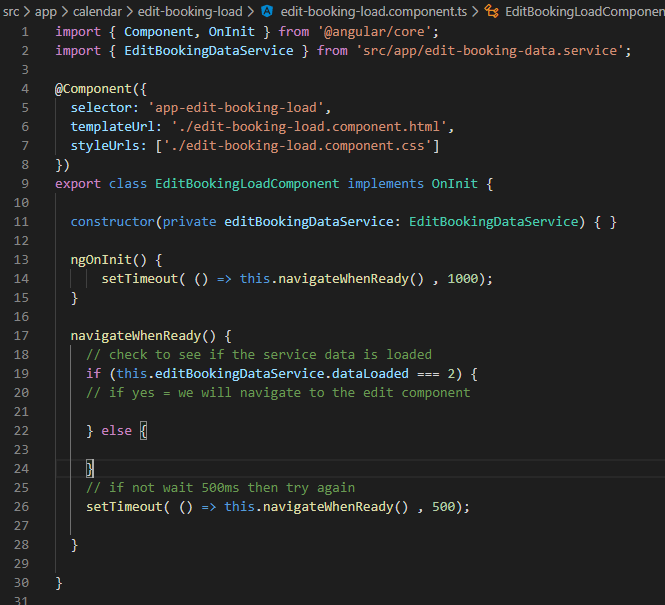


Set up the URL in app-module.ts Add the editBookingLoad url and @NgModule

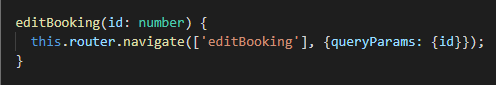


Let us put some html for the Please Wait …

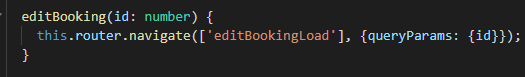




Now we need to make sure we navigate to the editBookingLoad in the calendar.component.ts



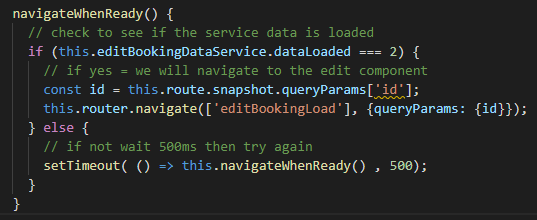
Change to



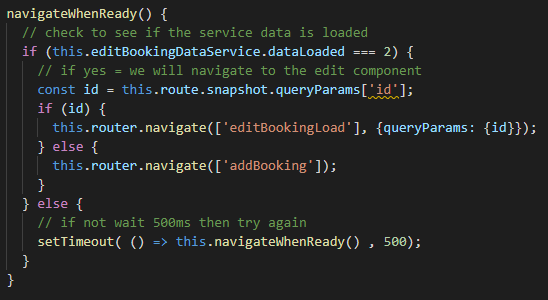
Now copy the route to the load component and add the id and route



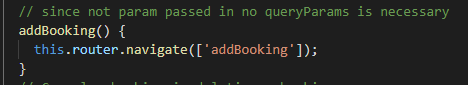
Now fix the setTimeout to the else block



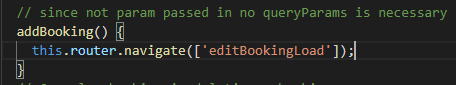
Add the addBooking



In the Calendar change the addBooking

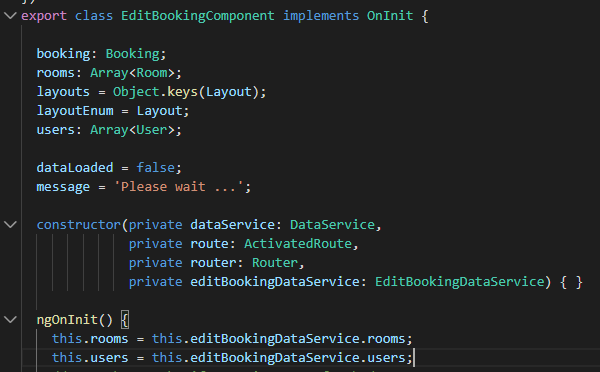


To

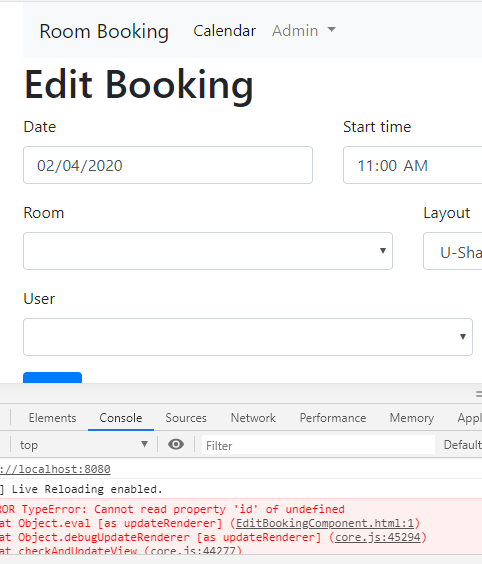


Calendar >> LoadData >> Service

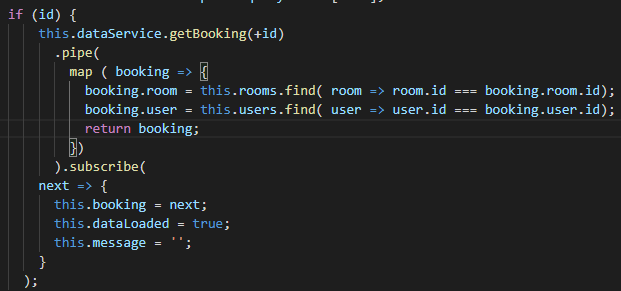
Now finish the last part in edit-booking-component.ts



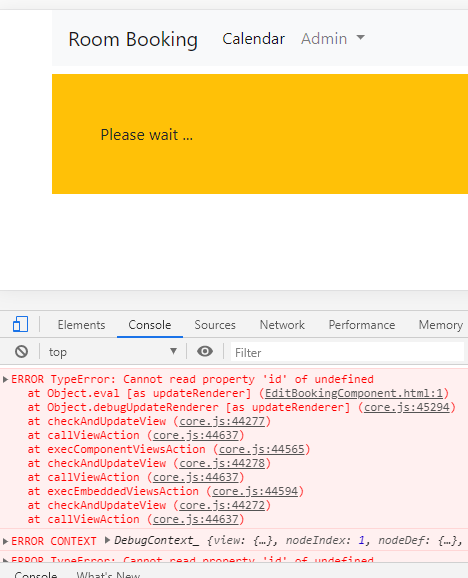
Still have an error



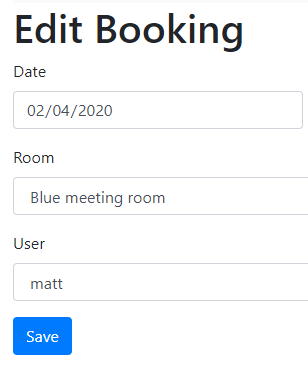
Which room and which user should be assigned as the default:



Test this

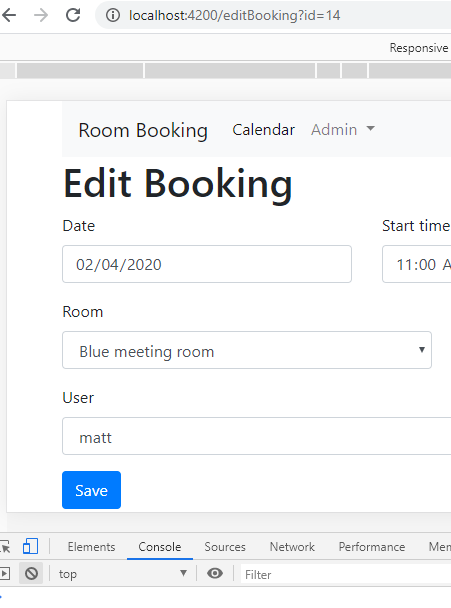


Let us start from the beginning to see if from Amend the room and user prepopulate despite the error



In the edit-booking-component.html file add a null safety





No errors

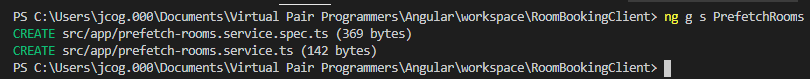
Using a Resolver:

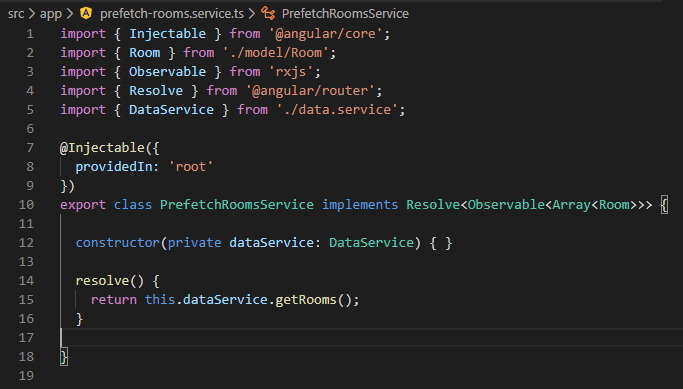
Production Pre-fetch solution

A resolver resolves an Observer

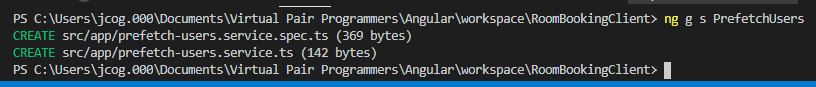
Create 2 services to replace the solution that we have just created

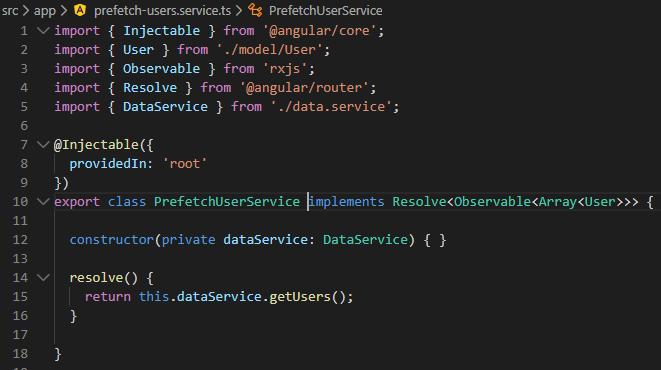
>ng g s PrefetchRooms



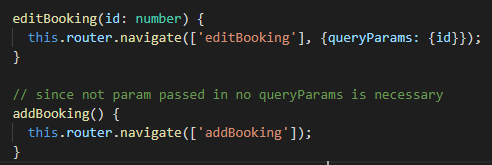


>ng g s PrefetchUsers

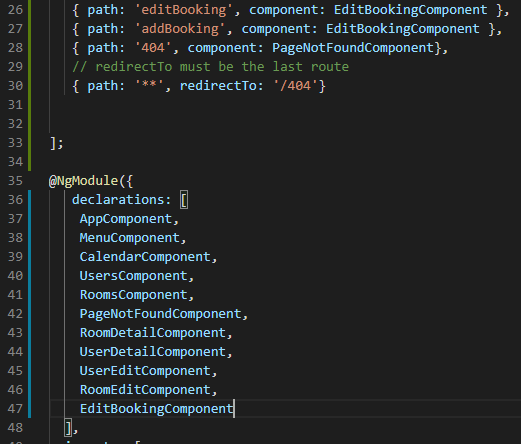




Go to the Calendar-component.ts and replace the editBookingLoad object:

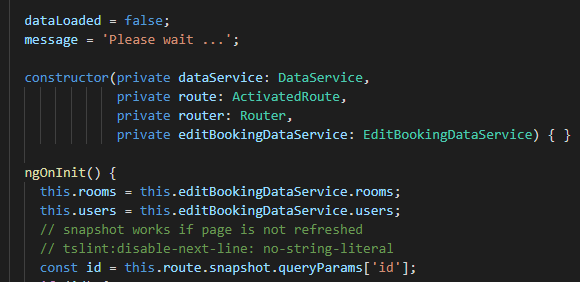


Remove the navigation line from the app.module.ts class for the EditBookingLoadComponent

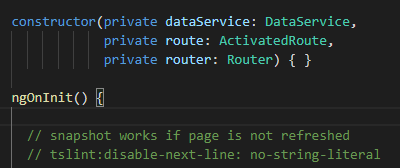


Remove routes and references from the edit-booking-component.ts class

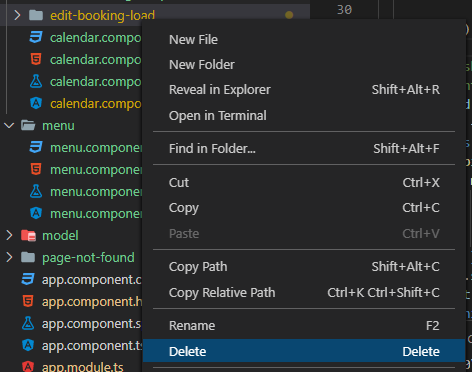
Before:



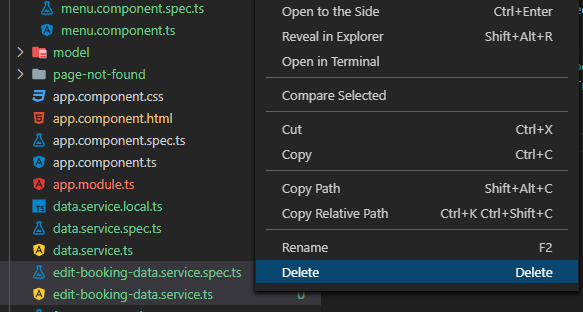
After:



Delete the edit-booking-load components



Delete the edit-booking-data.service files



Remove the import from the app.module.ts file

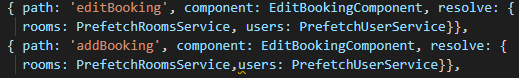


Remove the import from the edit-booking.component.ts file

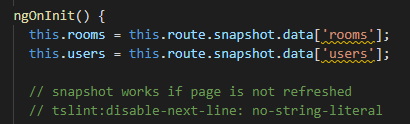


Now it compiles!

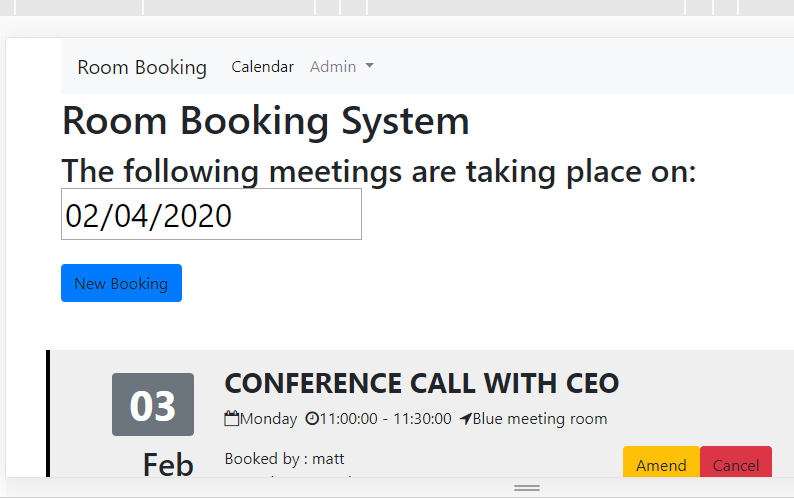
Go to the app-module.ts file to add additional resolving params to the routes



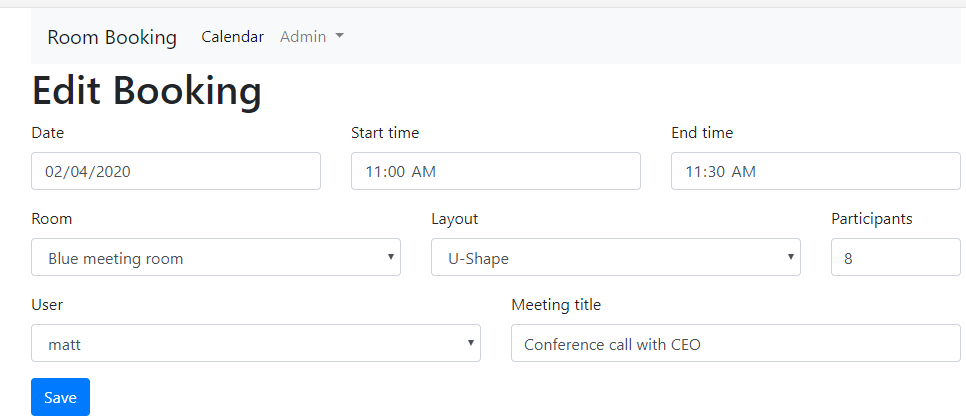
Now, in the edit-booking.component.ts file repopulate the rooms and users from the resolve route:



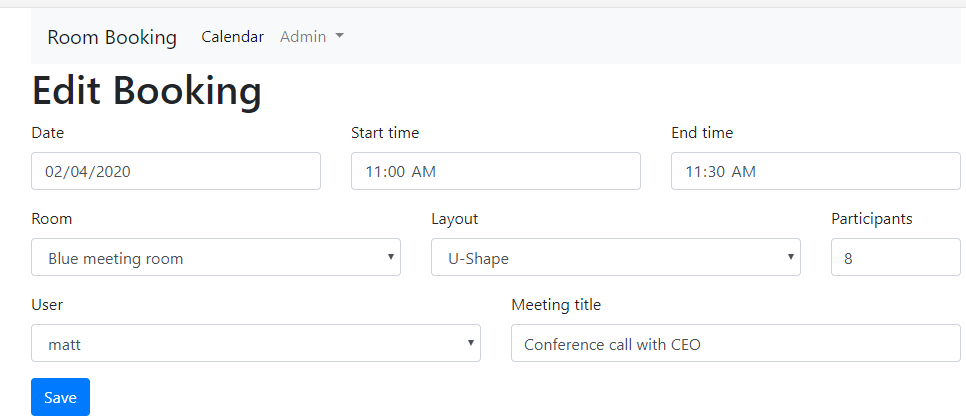
Test this:



Choose Amend



Now try the refresh button, to see if the page reloads correctly

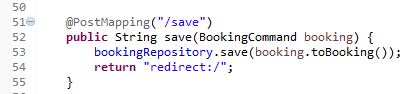


Reloads with no console errors, even with typing into the Meeting title field asdfasdfa

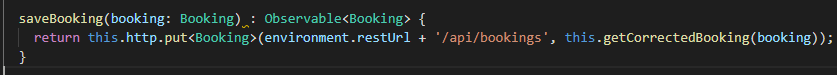
Implement the SAVE button:

We have done save on other pages so no read differences!!!

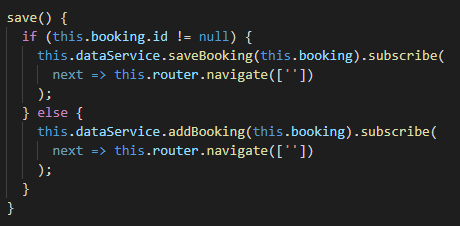
BookingController.java



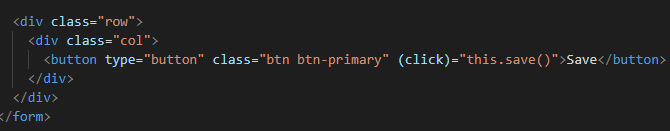
In the data.service.ts file



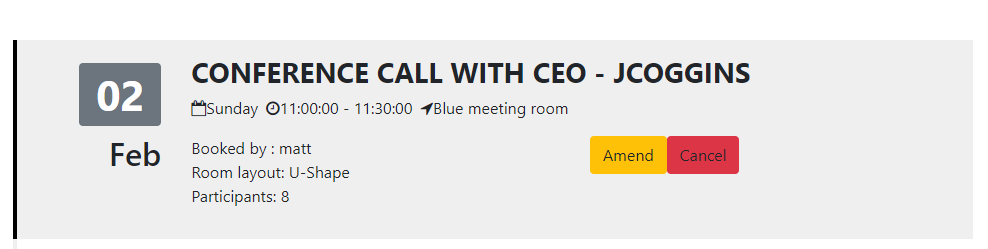
In the edit-booking.component.ts file



In the edit-booking.component.html file



Test it



It works