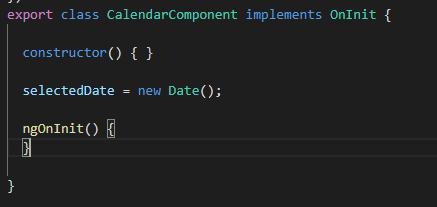
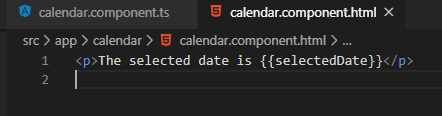
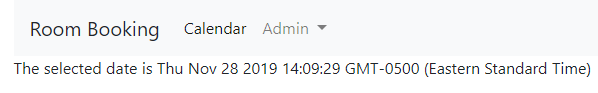
**Displaying Data with Pipes:**

Displaying Dates in a particular format in Angular



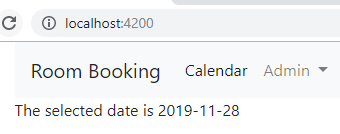
Now, show the date

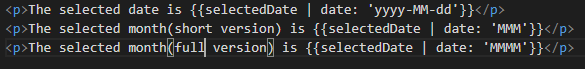


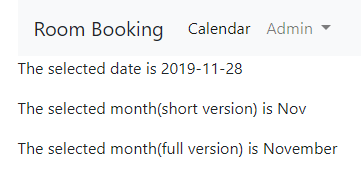


Now use Pipes:



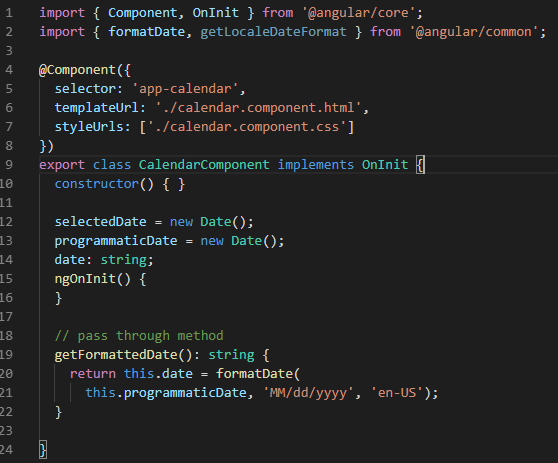






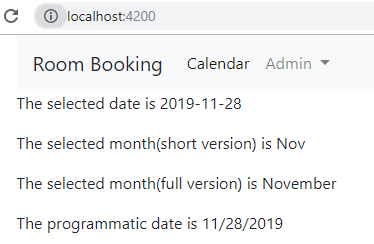
<http://angular.io/guide/pipes>

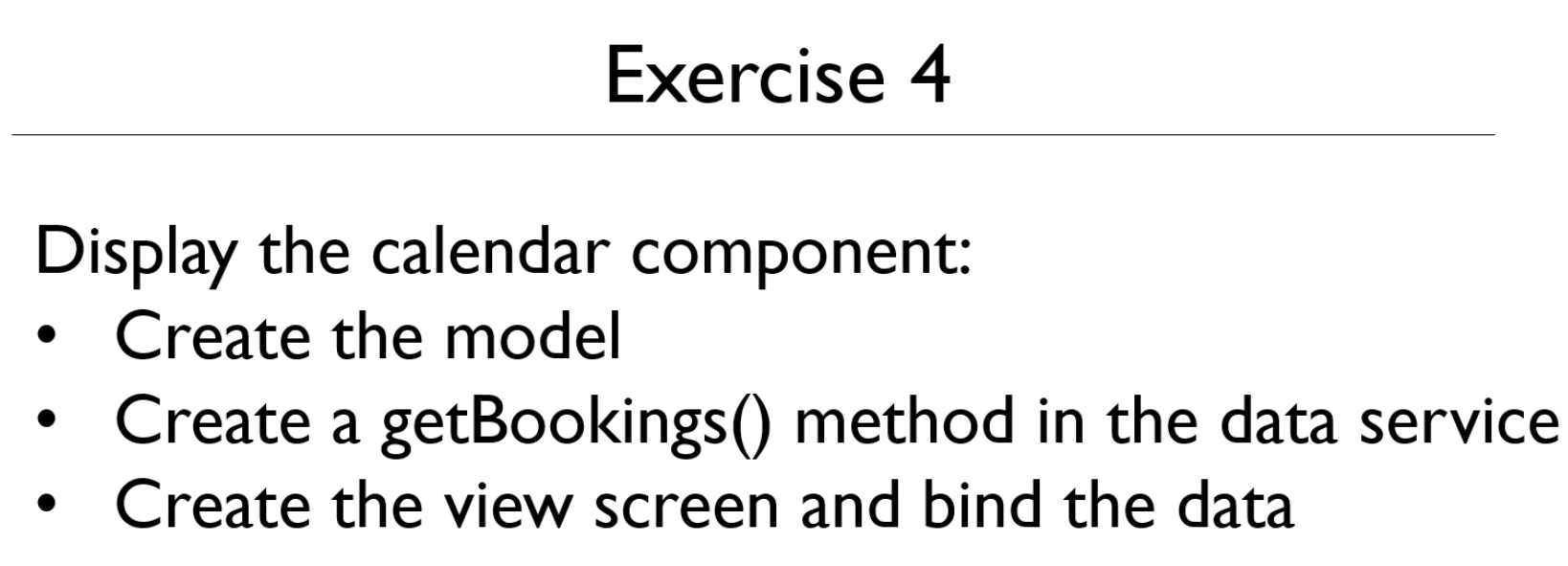
Programmatic Date



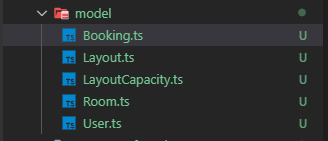


Displaying the Calendar Screens:

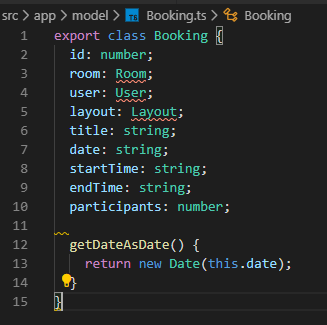




Create the model Booking.ts

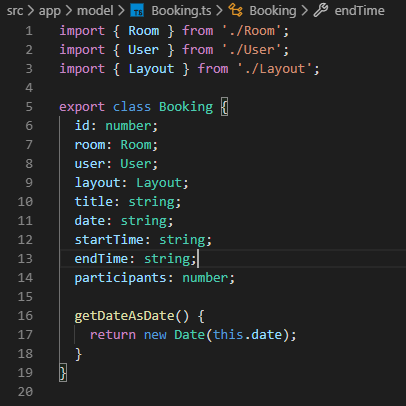


Copy the starting workspace model and edit for typescript

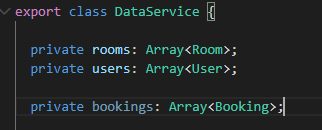


Need to fix imports

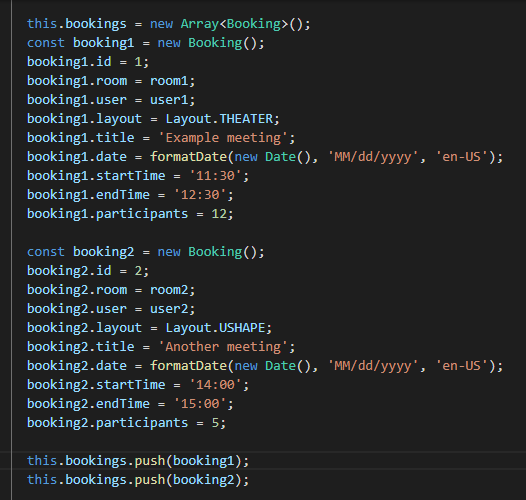
To import first highlight, Ctrl + . then Enter



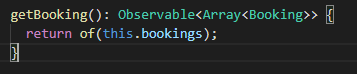
Create Dummy Array of Data in the DataService



Data:



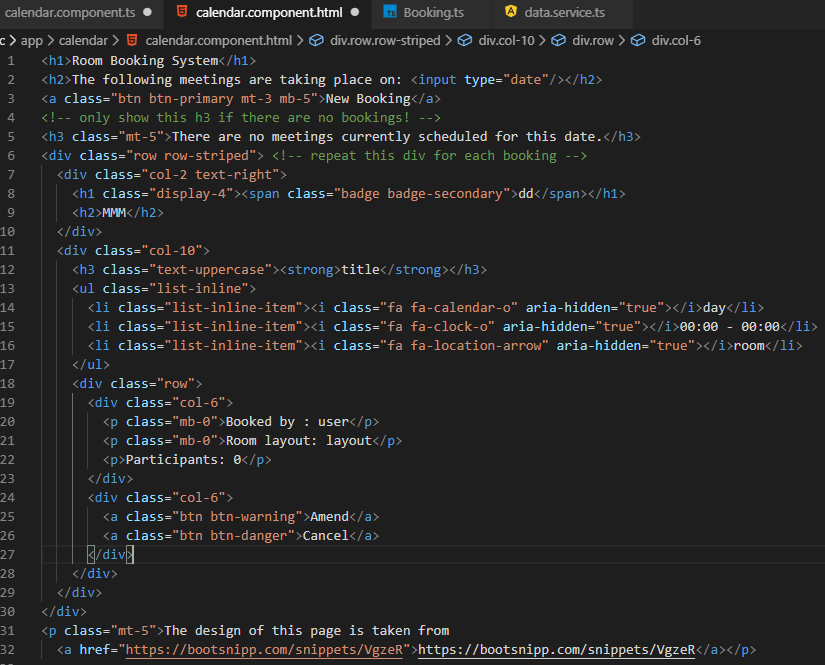
Method for obtaining the booking



That completes the data service

Let us now create the view

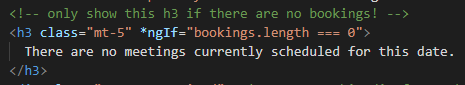
So, in the Calendar component paste the following



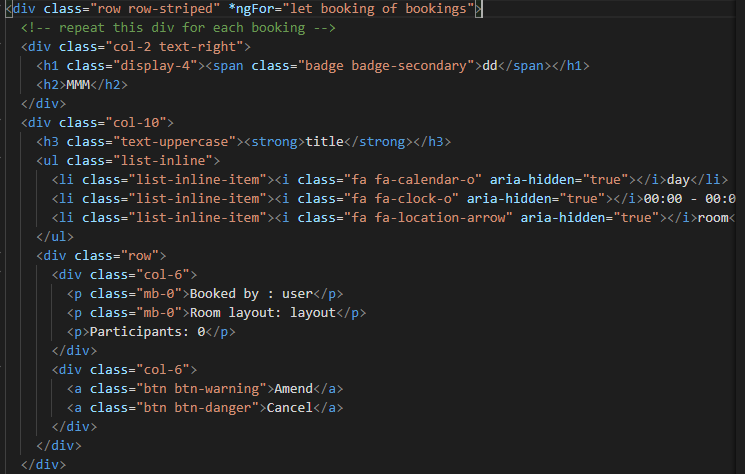
Let us create a reference to the bookings data in the calendar.component.ts file



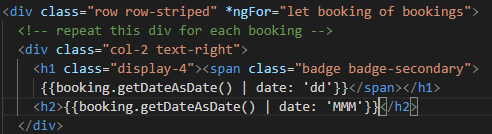
Now, bind the data to the html



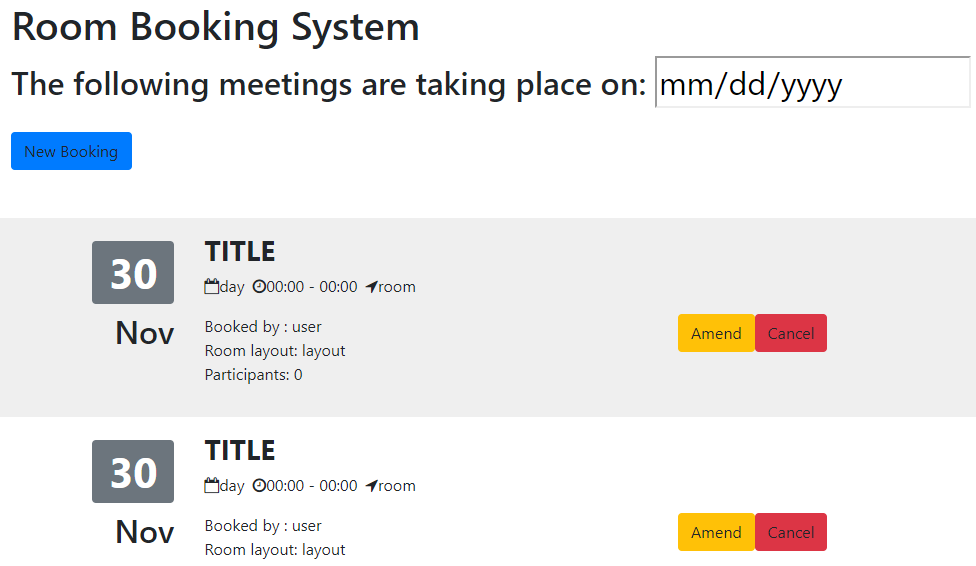
We need to loop of bookings



Add the date



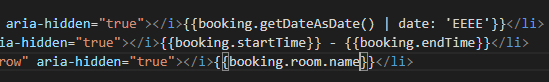
Test this



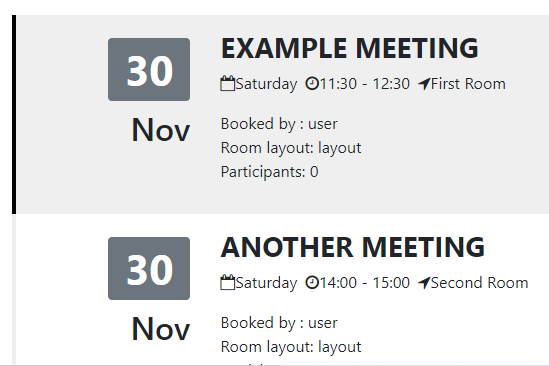
Add the Booking Title



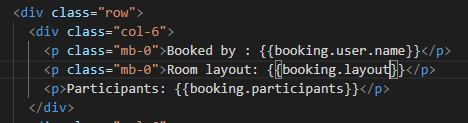
Add the day, times and room



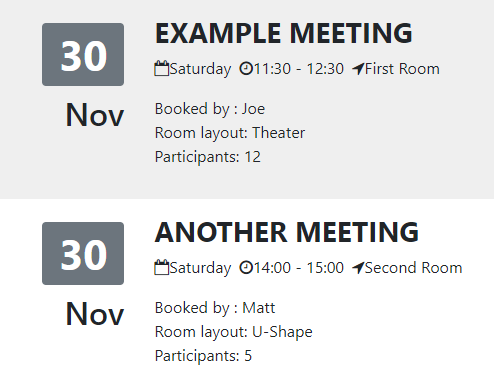
Test it



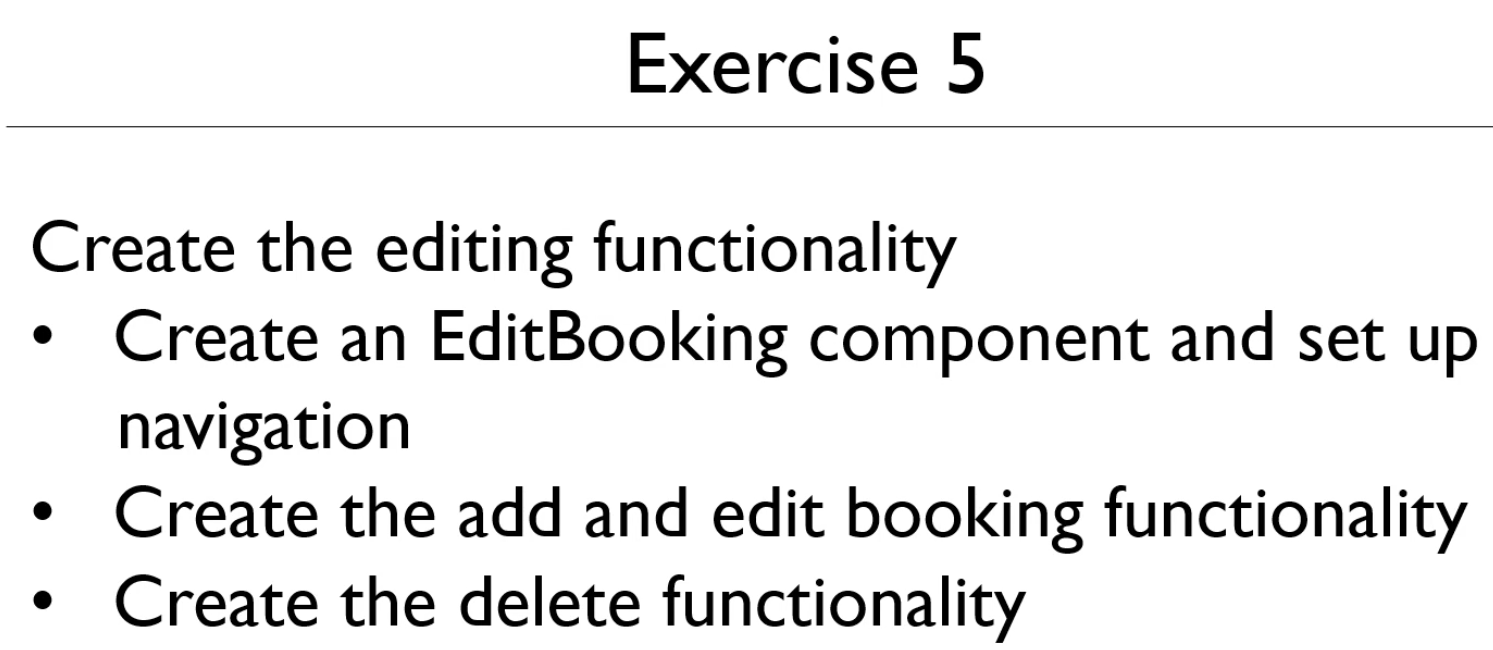
Add user, layout and participants



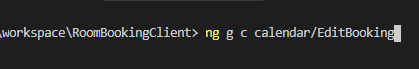
Test it

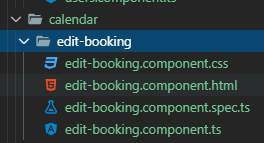


Next the Edit Screen: Exercise 5

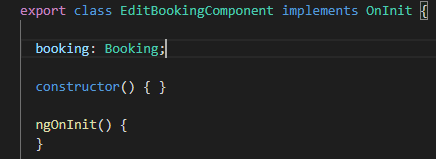


Create a EditBooking Component:

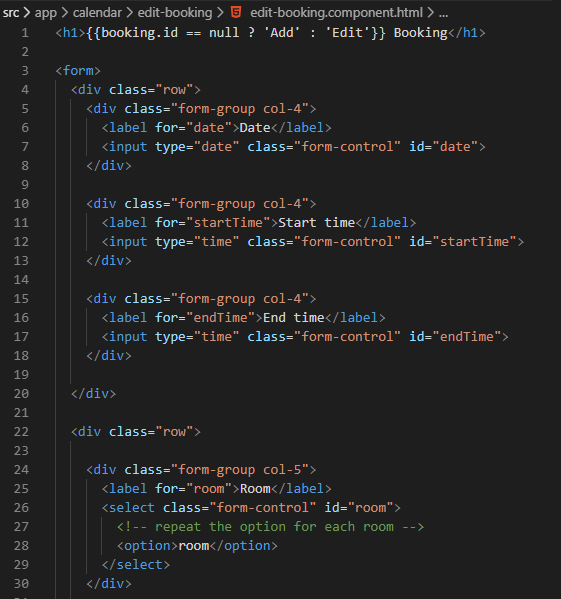


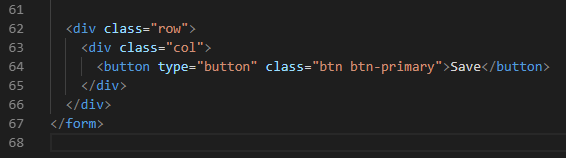


Add a booking element to the class

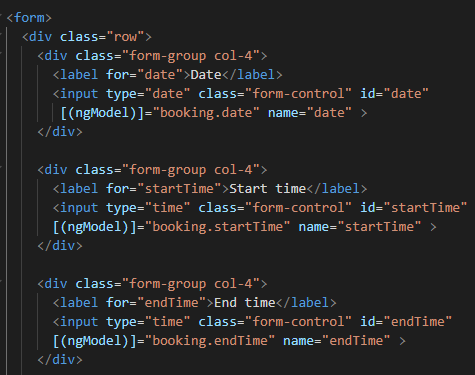


Add the html

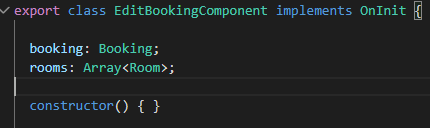




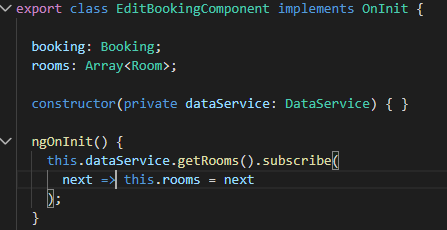
Start binding elements



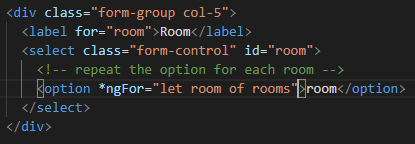
Create a backing object rooms



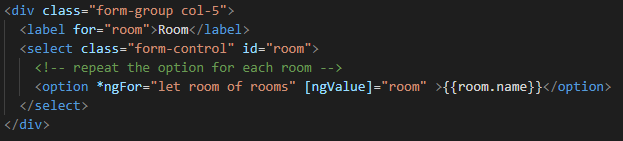
Source of the data is the dataService



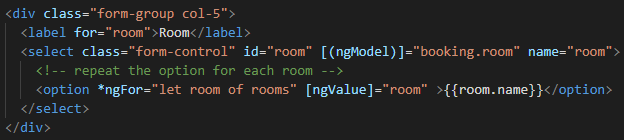
Now, loop through the rooms



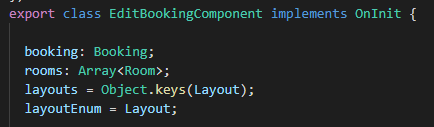
Binding room and outputting the room name

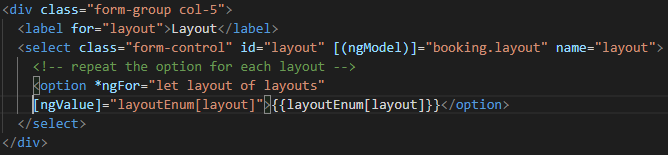


Bind the select to actual booking room

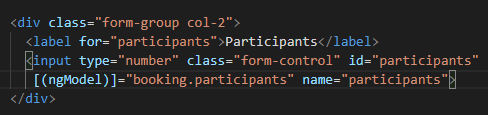


Do the same for the Layout

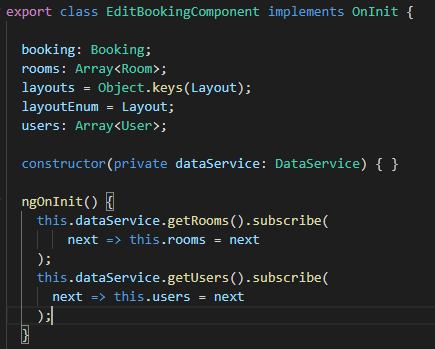


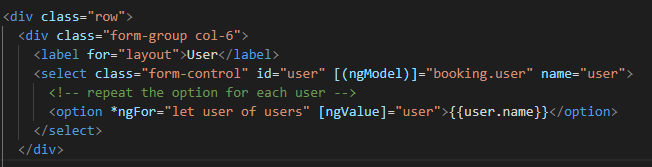


Let us do the participants

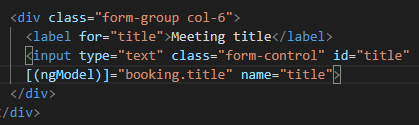


Now the loop for the users

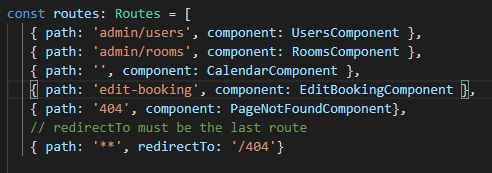




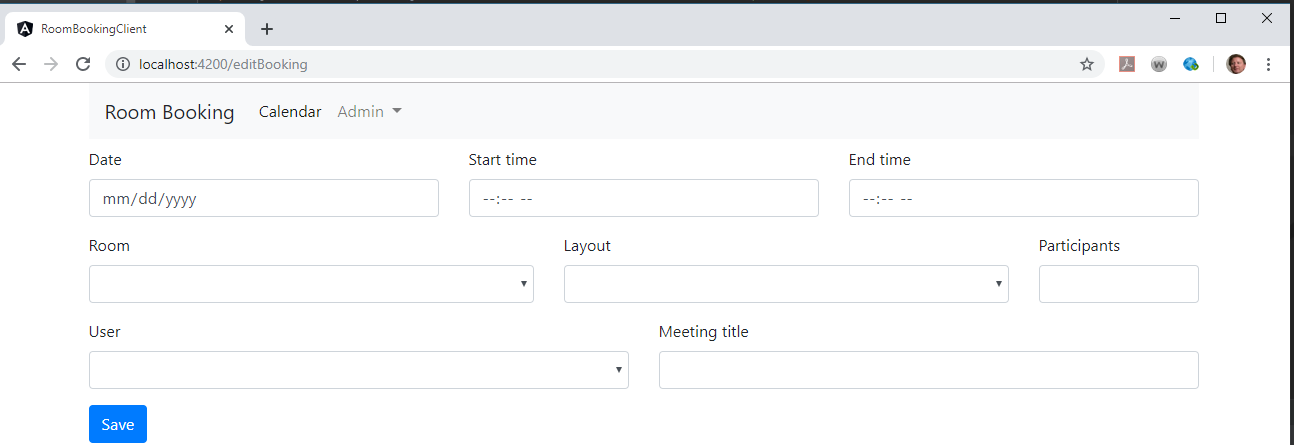
Now, meeting title



Now, set up the navigation

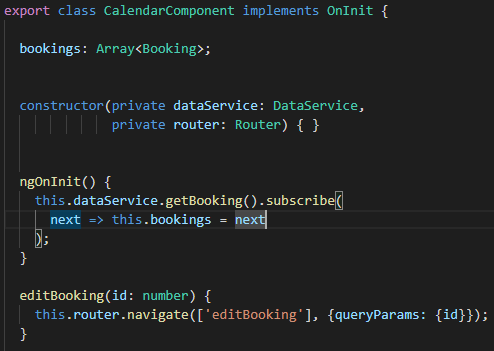


Test it

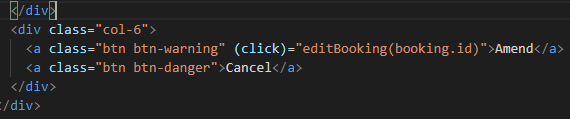


Still more work to clean up console errors

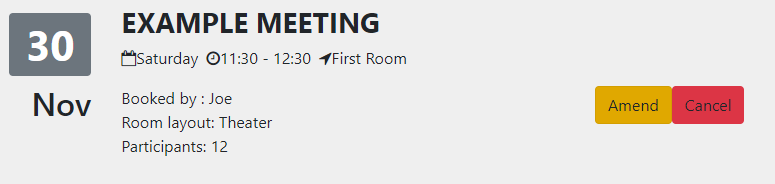
Complete the routing off the button click

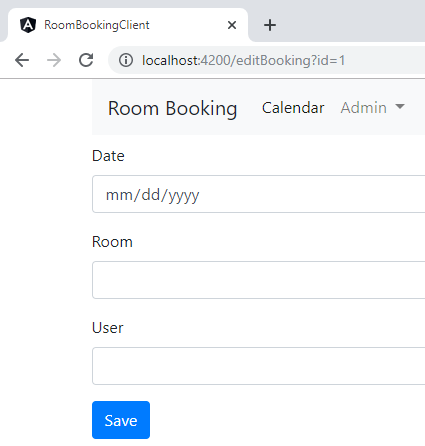


Bind the click event to the Amend button



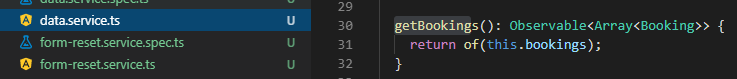
Test this by clicking amend

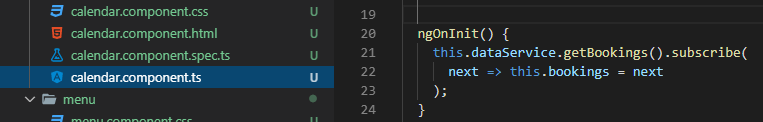




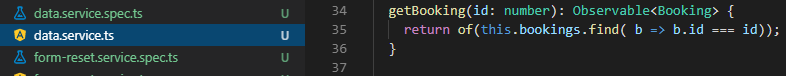
Let us grab the id by getting a reference to the activated route

The getBooking() that was already present had to be refactored to getBookings();

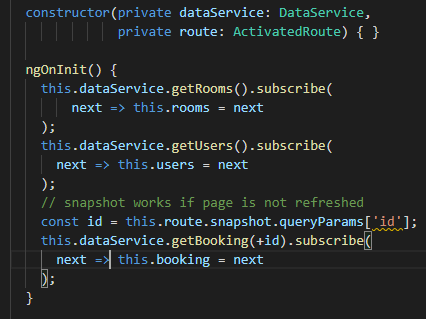




Now use getBooking() for the new method

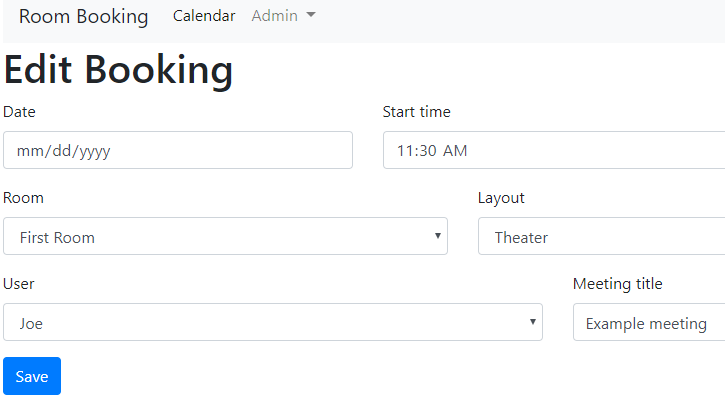


Back to the editBooking class



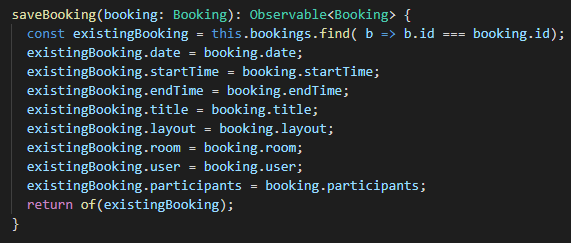
Test it

On clicking the Amend on the first Booking - data is populated correctly

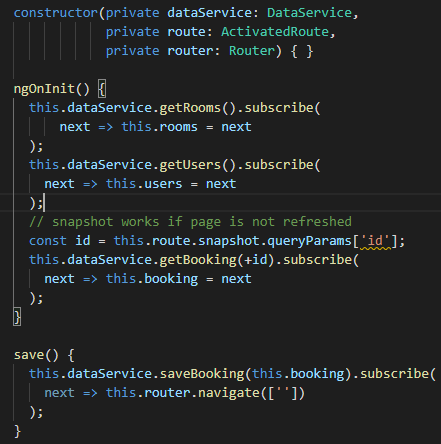


Implement the Save button

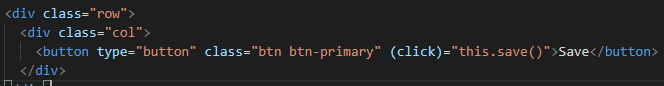
In the dataService



Create the pass through method in the EditBooking class

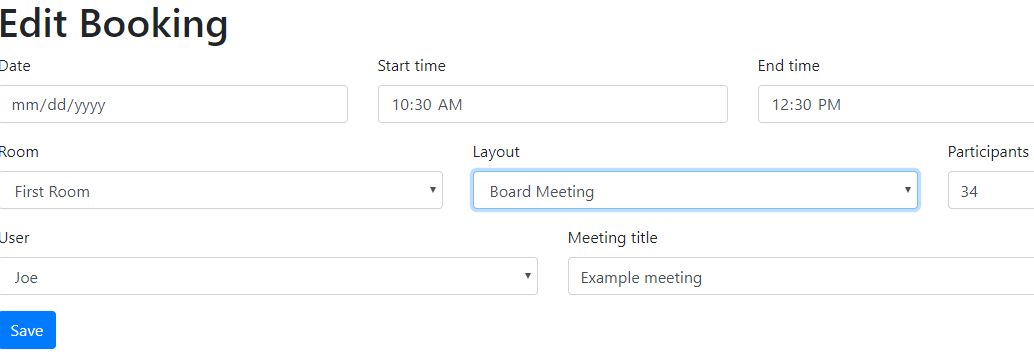


Bind the save() to the button Save In the editBooking html file

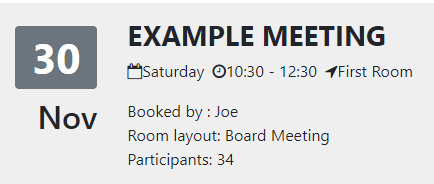


Test this by editing a booking

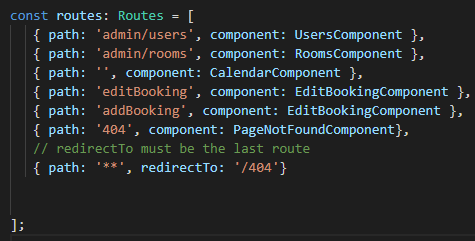
Changed startTime from 11:30, Layout from Theatre, and participants from 12.



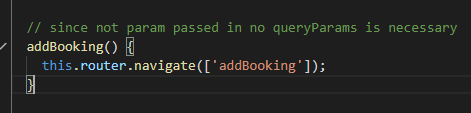
Changes were saved



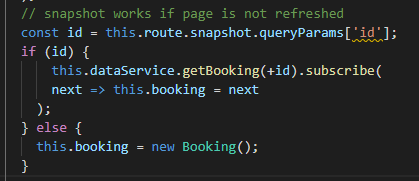
Now create a new URL for our New Booking button That is addBooking



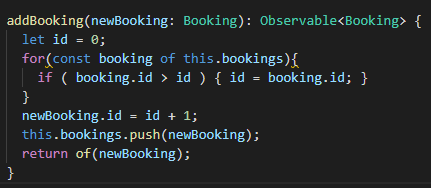
Now the click method in the calendar class



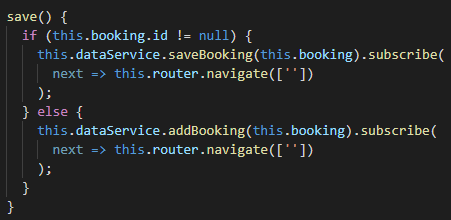
Need a null check on id



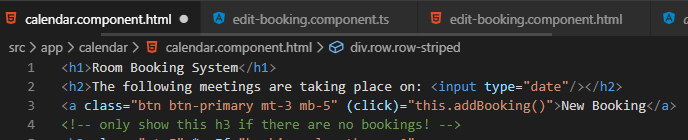
Create addBooking in the dataService



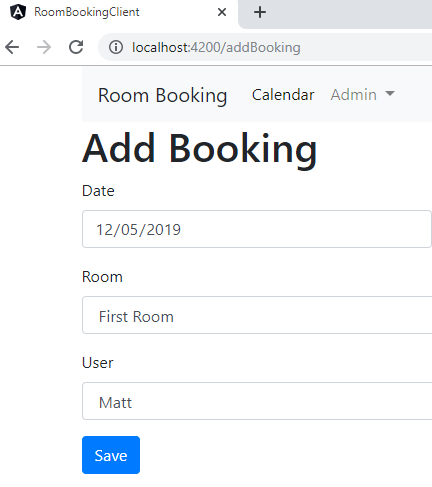
Fix the save() for id check

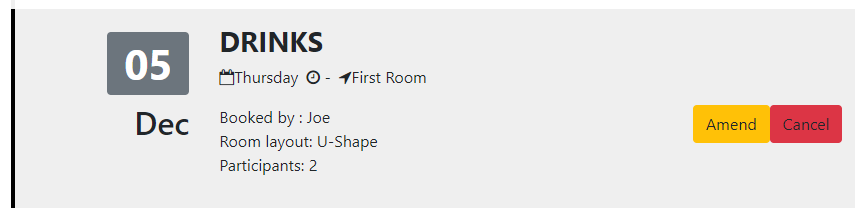


Bind the addBooking for the Click event



Test it Click new booking button

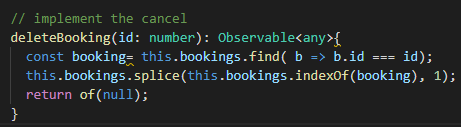




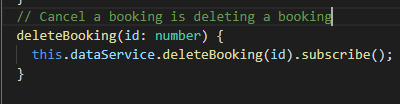
It worked!!

Implement the Cancel button

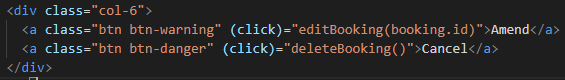
In the dataservice



Pass through method is in the calendar class

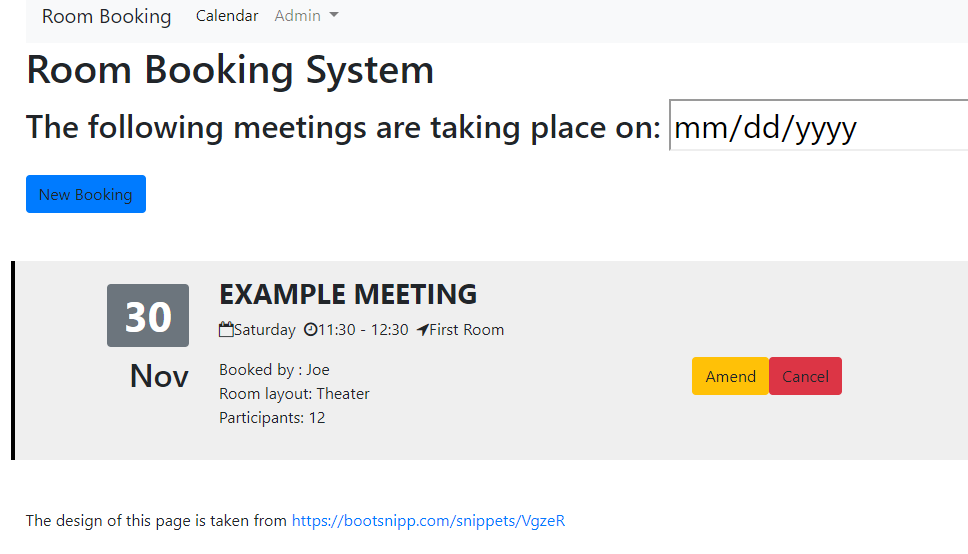


Databind in the html page



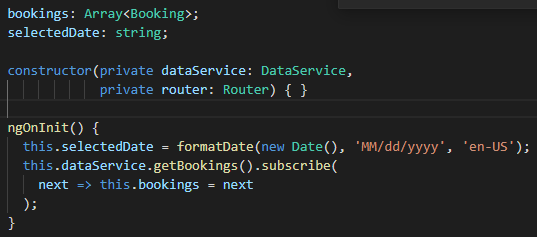
Test it

Default is 2 bookings successfully removed one

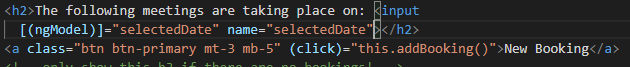


Now let us get the Calendar component to work

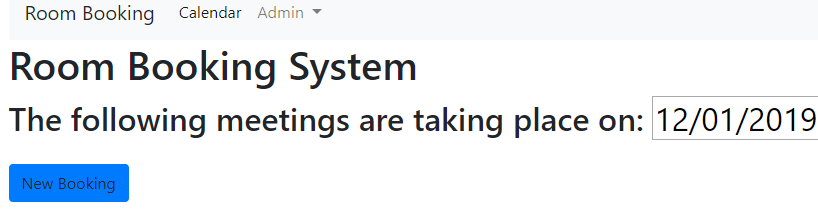
Let us create a selectedDate of type string



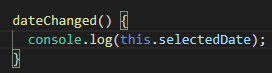
Bind the selectDate

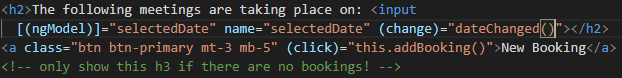


Test it

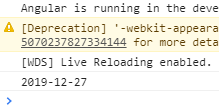


Update the date in the bookings if changed in the calendar





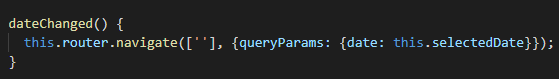
Test that date shows up in the console



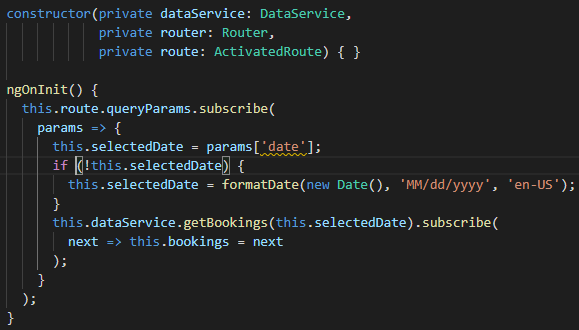
Redo dataservice getBookings() to handle dates



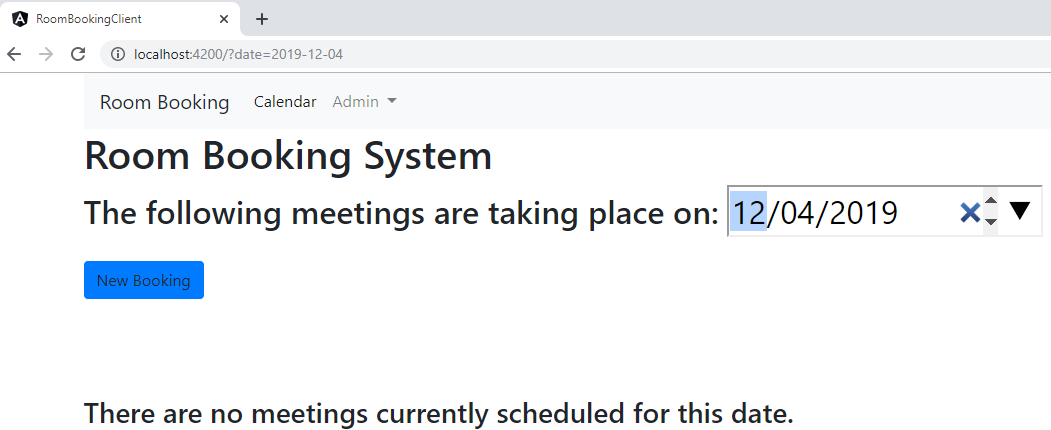
Now add navigation



Need the ActivatedRoute and refactor the methods



Test it



<h1>{{booking.id == null ? 'Add' : 'Edit'}} Booking</h1>

<form>

<div class="row">

<div class="form-group col-4">

<label for="date">Date</label>

<input type="date" class="form-control" id="date">

</div>

<div class="form-group col-4">

<label for="startTime">Start time</label>

<input type="time" class="form-control" id="startTime">

</div>

<div class="form-group col-4">

<label for="endTime">End time</label>

<input type="time" class="form-control" id="endTime">

</div>

</div>

<div class="row">

<div class="form-group col-5">

<label for="room">Room</label>

<select class="form-control" id="room">

<!-- repeat the option for each room -->

<option>room</option>

</select>

</div>

<div class="form-group col-5">

<label for="layout">Layout</label>

<select class="form-control" id="layout">

<!-- repeat the option for each layout -->

<option>layout</option>

</select>

</div>

<div class="form-group col-2">

<label for="participants">Participants</label>

<input type="number" class="form-control" id="participants">

</div>

</div>

<div class="row">

<div class="form-group col-6">

<label for="layout">User</label>

<select class="form-control" id="user">

<!-- repeat the option for each user -->

<option>user</option>

</select>

</div>

<div class="form-group col-6">

<label for="title">Meeting title</label>

<input type="text" class="form-control" id="title">

</div>

</div>

<div class="row">

<div class="col">

<button type="button" class="btn btn-primary">Save</button>

</div>

</div>

</form>