Step 1 create Angular Project , say to routing to yes while creating project

// fetching all data from backend step

step 2 create class with below command

ng g class user

keep all variable same as your pojo in SB

Step 3

create compnent with below command

ng g c allUser

step4

initializw your class in compnent to find data in form of data

u:User[];

bind class variable using \*NgFor in component.html file

step 5

generate service file using below command

ng g s user

go in the app.module.ts file and import httpclient module

go in service file wtire fun which return all user and it give you user array

example

export class AddflightService {

private basrURL="http://localhost:9094/allFlights";

constructor(private http:HttpClient) { }

getAllFlightsList():Observable<AddFlight[]>{

return this.http.get<AddFlight[]>(`${this.basrURL}`);

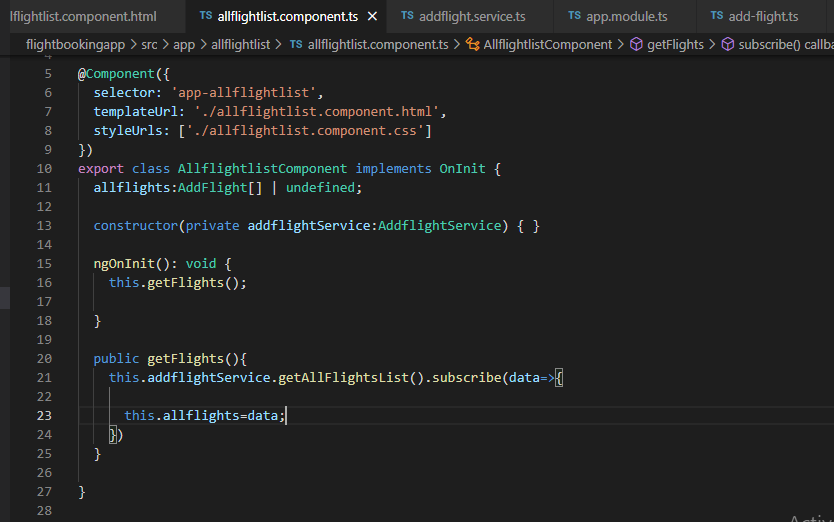
}

}

step 6

go in component.ts file, inside constructor inject service class

and call service fun there



export class AllflightlistComponent implements OnInit {

allflights:AddFlight[] | undefined;

constructor(private addflightService:AddflightService) { }

ngOnInit(): void {

this.getFlights();

}

public getFlights(){

this.addflightService.getAllFlightsList().subscribe(data=>{

this.allflights=data;

})

}

now run the project

ng serve

=============================================

routing

for routing go in routing file define write some router like below as per your componet

const routes: Routes = [

// { path: 'index', component:IndexComponent},

// { path: '', redirectTo:'index',pathMatch:'full'},

{ path: 'login', component: LoginComponent },

{ path: 'logout', component: LogoutComponent},

{ path: 'userregistration', component: UregistrationComponent},

{ path: 'home', component: HomeComponent},

{ path: 'allflights', component: AllflightlistComponent},

{ path: 'addflight', component: AddflightComponent},

{ path: 'userlogin', component: UserloginComponent}

];

then go app.component.html file and write

<router-outlet></router-outlet>

==========================================

PostMaping Steps

two way data binding

ng g c createUser

create one fun in user.ts component file

addFlight:AddFlight=new AddFlight();

onSubmit(){

}

call fun in html

<form class="justify-content-center" (ngSubmit)="onSubmit()">

now use 2 way data binding by

use object reference and variable to bind which declared in class

//addFlight:AddFlight=new AddFlight();

<div class="form-group">

<label for="nonbusinessseats">Non-businessseats</label>

<input type="number" class="form-control" name="nonbusinessseats" id="nonbusinessseats" [(ngModel)]="addFlight.nonBusinessSeats" aria-describedby="nonbusinessseats" placeholder="Enter non-business seats">

</div>

<div class="form-group">

<label for="cost">Cost</label>

<input type="number" class="form-control" name="cost" id="cost" aria-describedby="cost" [(ngModel)]="addFlight.cost" placeholder="Enter ticket cost">

</div>

steps for post mapping

go in service.ts write fun

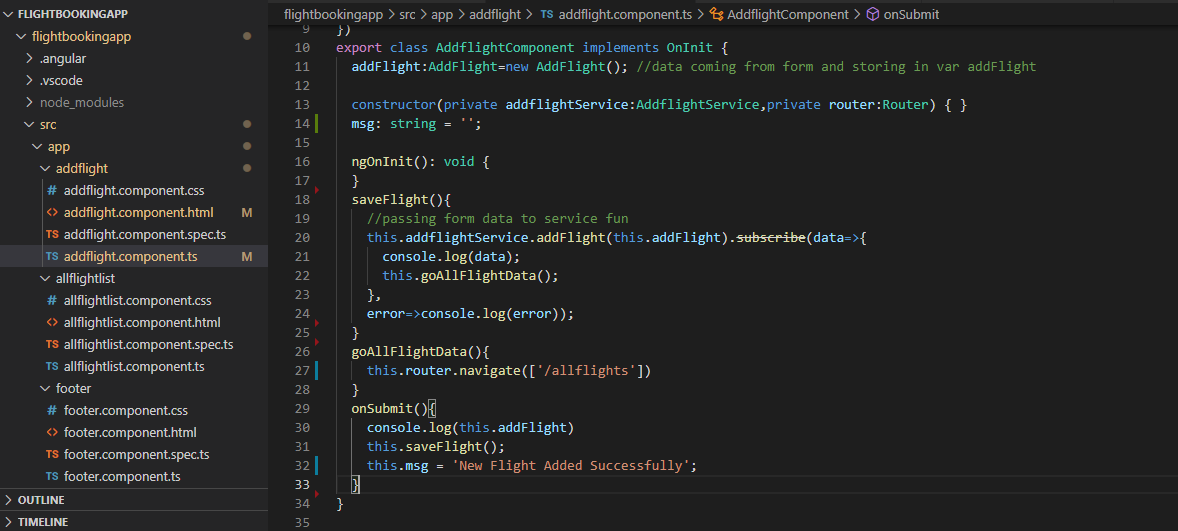
// for postmapping

addFlight( addFlight:AddFlight):Observable<Object>{

return this.httpClient.post(`${this.basrURL}`,addFlight)

}

Go to component to call service .ts fun



Below image show to print success message if record added



==========================

Update in NG

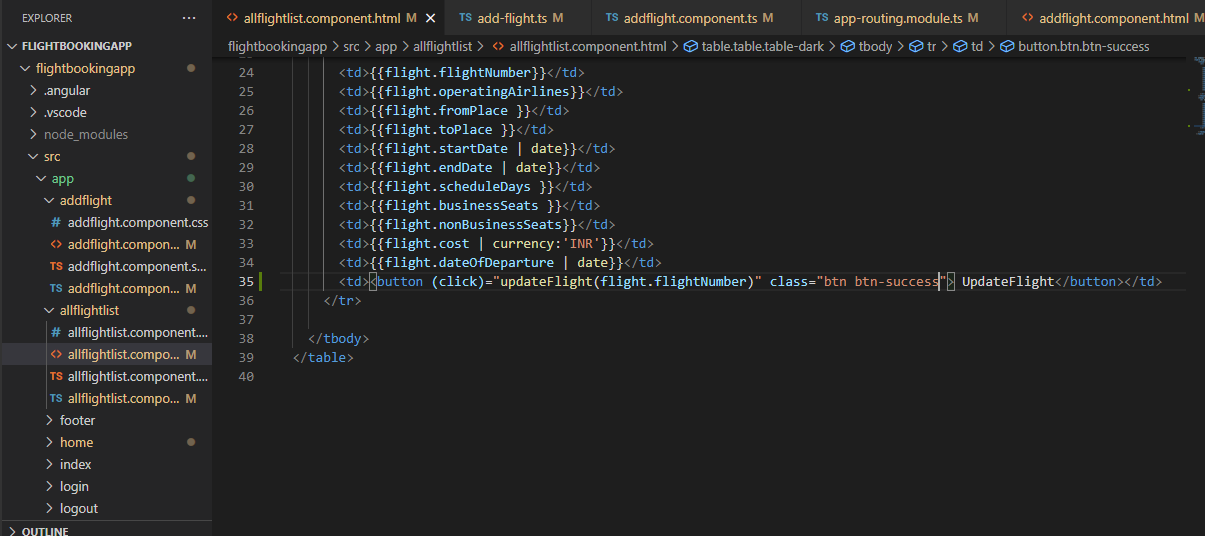
Create component

ng g c updateflight

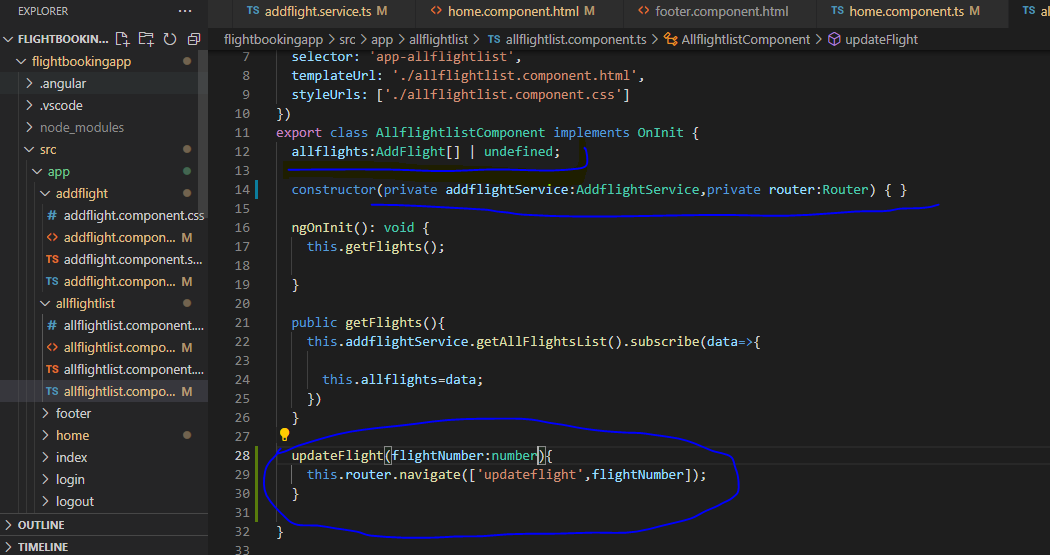
Go in routing module and add routing path

Fetching data by id

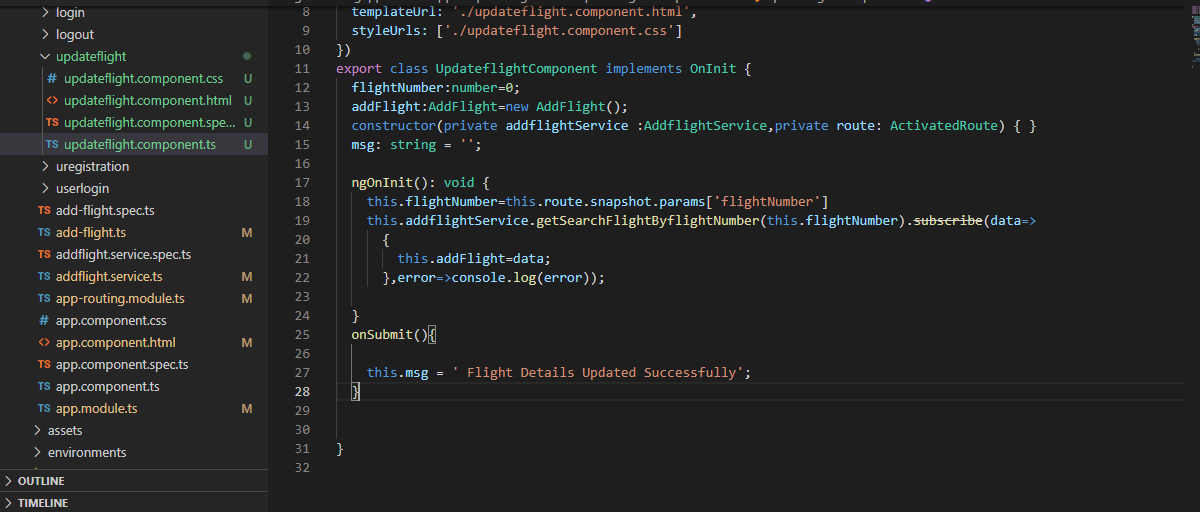
Alllist.html component file



Alllist.ts component file

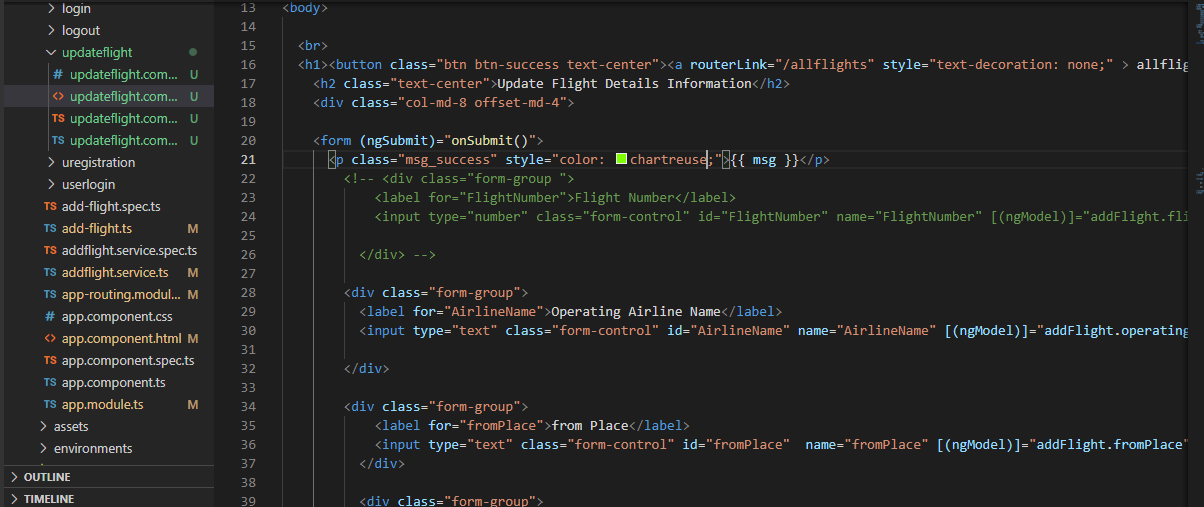


Updates ts file



Binding data

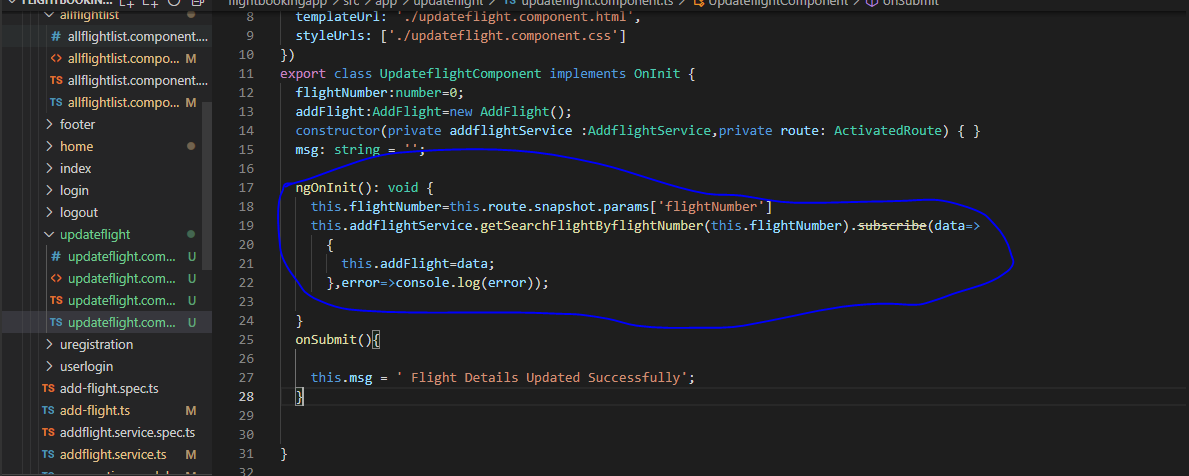
Update html file



Service file



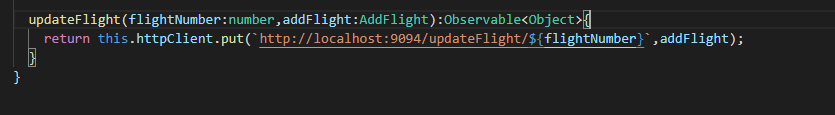
Call service fun in update.ts component file



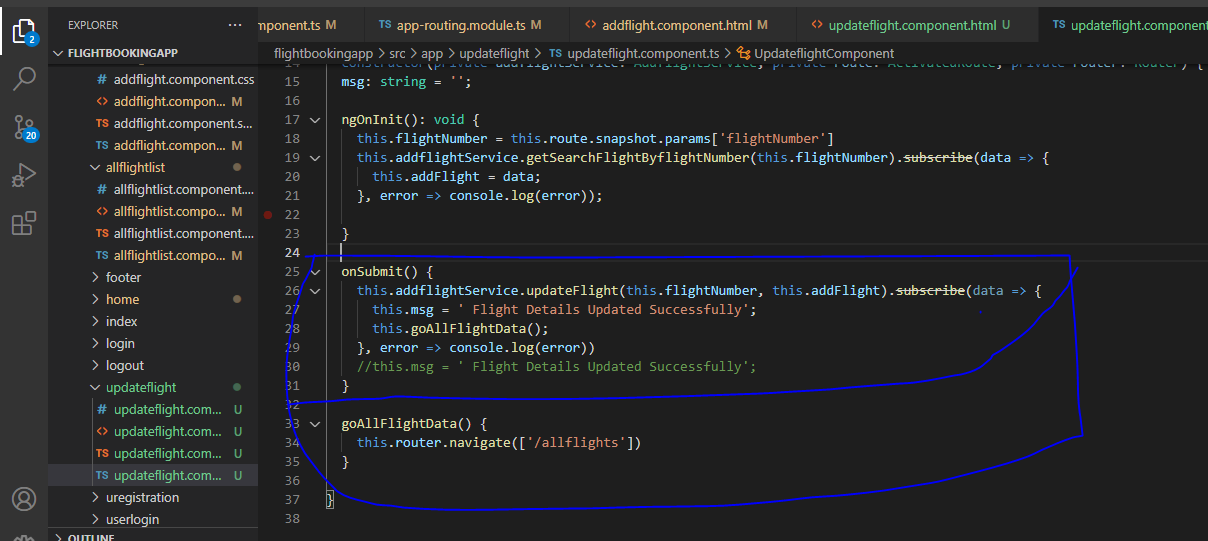
Check all data coming in update form or not

==============================

1 Create update fun in service

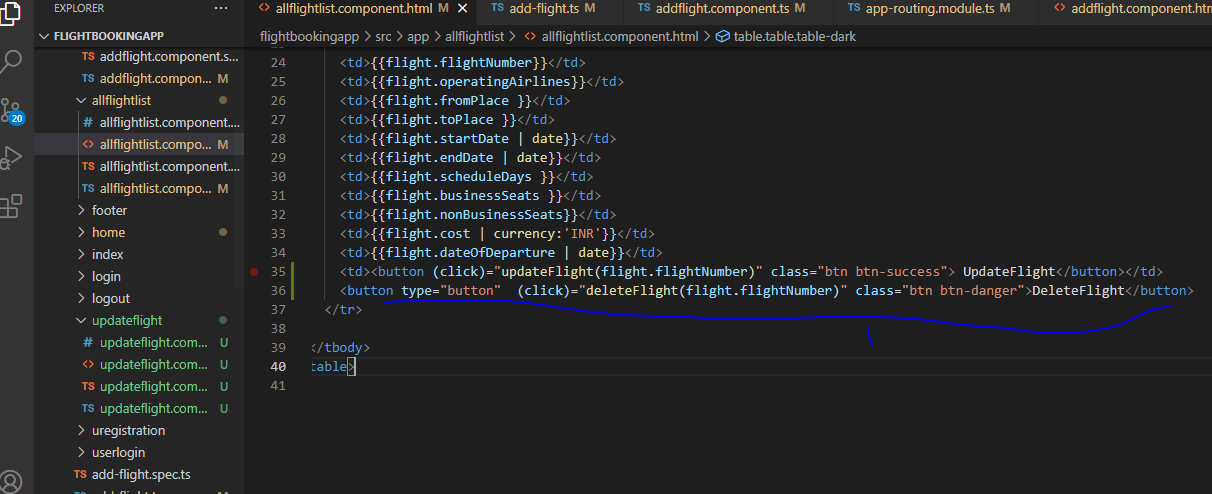


Call above fun in update.ts file componet



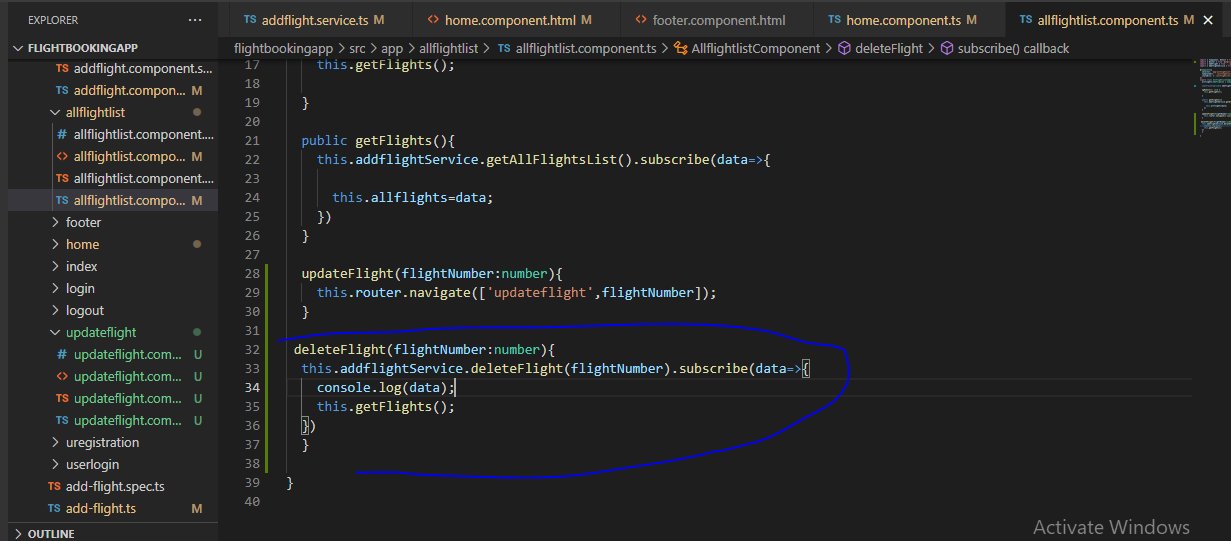
Delete Step in NG

1 All flightlist.html file



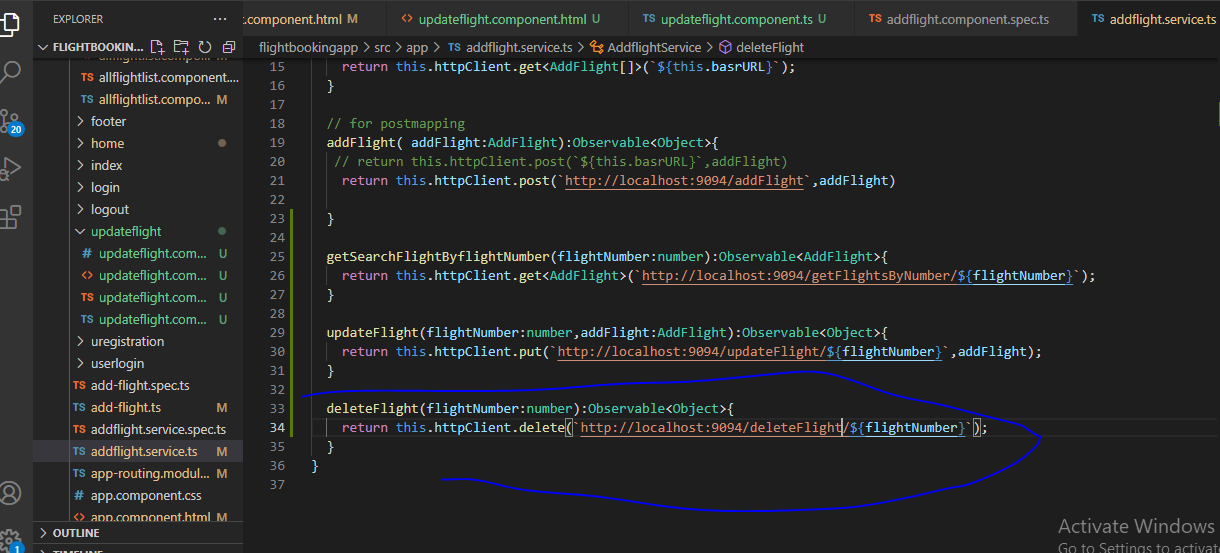
2

All flightlist.Ts file

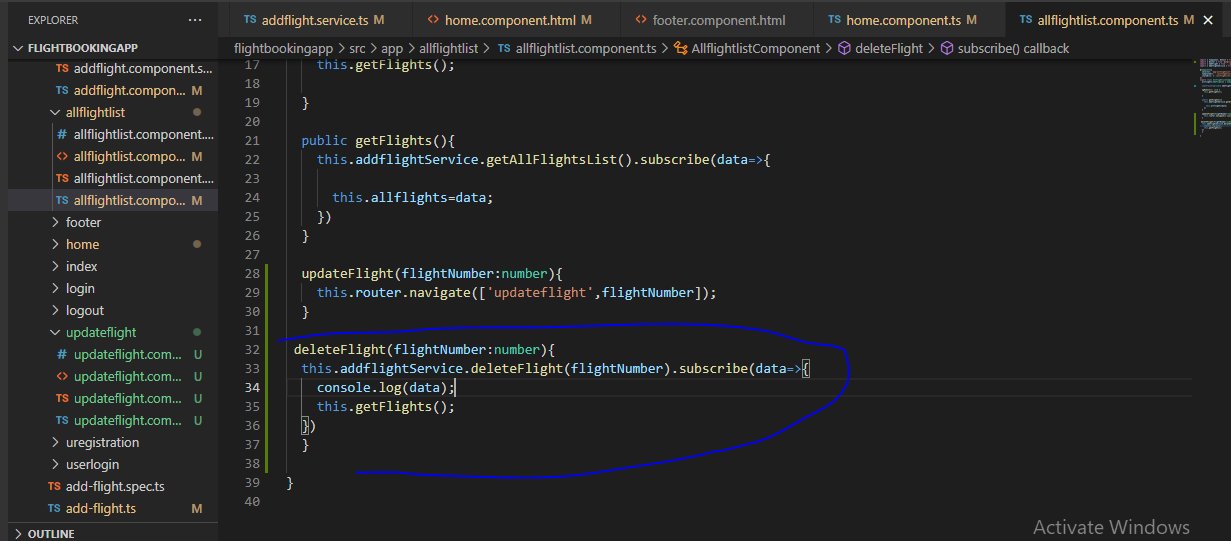


3 service file

Create delete fun



Allflightlist.ts



Ng g pipe ellipsis

ng g m admin //creating module

ng g m admin --routing ll creating routing module

ng g m admin/login // creating login folder inside admin