

# HAYAKOKI





# Book your flight now!

One Way

Round Trip

Departure

Arrival

Choose a date \*

5/4/2021

Travellers \*

1

Fly

## Places to check out!



# WHO IS IT FOR?



USER



ADMIN

# HOW DOES IT WORK?

# FRONTEND

Project



- > Hayakuki ~/Projects/Angular/Hayakuki
- > e2e
- > node\_modules library root

- > src

- > app

- > adminconsole
- > authorization
- > blog
- > booking
- > home
- > img

- > Material

- > searchlist

- > Services

- > test

- > User

- > app.component.html
- > app.component.scss
- > app.component.ts
- > app.module.ts
- > app-routing.module.ts

- > assets

- > environments

- > favicon.ico

- > index.html

```
onewaybooking.component.ts × onewaybooking.component.html × flightsearch.component.ts ×
18 ↑ templateUrl: './flightsearch.component.html',
19 ↑ styleUrls: ['./flightsearch.component.css', '../../../../../node_modules/ngx-bootstrap/bs-callout/bs-callout.css'],
20 }
21
22 export class FlightsearchComponent implements OnInit {
23   minDate;
24   curDate = new Date();
25   minpass = 1;
26   nums: number[] = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
27   // @ts-ignore
28   tripType = 'oneway';
29   deptCtrl = new FormControl();
30   arrivalCtrl = new FormControl();
31   filteredArrival: Observable<Airport[]>;
32   filteredDept: Observable<Airport[]>;
33   // @ts-ignore
34   apFetchServer: object;
35   airports: Airport[] = [];
36   constructor( private odqds: OnewayquerydataService, private route: Router,
37                 private airportService: AirportfetchService, private rte: ActivatedRoute ) {
38     this.minDate = new Date();
39     this.airportService.getAllAirports().subscribe( next: e =>
40       {
```

Project

+ -

Hayakuki ~/Projects/Angular/Hayakuki

- e2e
- node\_modules library root
- src
  - app
    - adminconsole
      - adminflight
        - adminflight.component.css
        - adminflight.component.html
        - adminflight.component.ts
        - adminflight.service.ts
      - adminflightcreate
        - adminflightcreate.component.css
        - adminflightcreate.component.html
        - adminflightcreate.component.ts
      - adminflightedit
        - adminflightedit.component.css
        - adminflightedit.component.html
        - adminflightedit.component.ts
    - authorization
      - authguard.guard.ts
      - authorization.service.ts
      - signup.service.ts
      - user-details.service.ts
    - blog

onewaybooking.component.ts × onewaybooking.component.html × flightsearch.component.ts ×

```
1 import { Injectable } from '@angular/core';
2 import {HttpClient} from '@angular/common/http';
3 import {Observable} from 'rxjs';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class AdminflightService {
9   baseUrl = 'https://localhost:5001/api/FlightAdmin/';
10  constructor( private http: HttpClient) { }
11  getAllFlights(): Observable<object> {
12    return this.http.get(this.baseUrl);
13  }
14  getOneFlight(Fno: string): Observable<object>{
15    return this.http.get( url: this.baseUrl + Fno);
16  }
17  postFlight( data: any): Observable<object> {
18    return this.http.post(this.baseUrl, data);
19  }
20  putFlight(data: any, Fno: string): Observable<object>{
21    return this.http.put( url: this.baseUrl + Fno, data);
22  }
23  delFlight(Fno: string): Observable<object>{
```

Project

The screenshot shows a file explorer on the left and a code editor on the right. The file explorer displays a project structure under 'Project':

- blog
  - blog-post
    - blog-post.component.css
    - blog-post.component.html
    - blog-post.component.ts
  - blogcreate
    - blogcreate.component.css
    - blogcreate.component.html
    - blogcreate.component.ts
  - blogedit
    - blogedit.component.css
    - blogedit.component.html
    - blogedit.component.ts
  - blog
    - blog.component.css
    - blog.component.html
    - blog.component.ts
  - booking
    - onewaybooking
      - onewaybooking.component.css
      - onewaybooking.component.html
      - onewaybooking.component.ts
    - roundtripbooking
      - roundtripbooking.component.css
      - roundtripbooking.component.html
      - roundtripbooking.component.ts

onewaybooking.component.ts × onewaybooking.component.html × flightsearch.component.ts ×

```
9  selector: 'app-roundtripbooking',
10 templateUrl: './roundtripbooking.component.html',
11 styleUrls: ['./roundtripbooking.component.css', '../../../../../node_mod
12 })
13 export class RoundtripbookingComponent implements OnInit {
14   passengerform: FormGroup; subscription: Subscription; selectedFlight: Flight;
15   constructor(private fb: FormBuilder, private roundBooking: Roundbookin
16   this.subscription = this.roundBooking.currentMessage.subscribe( ne
17   this.airportFetch.getOneAirport(this.selectedFlight.from).subscribe(
18   this.airportFetch.getOneAirport(this.selectedFlight.to).subscribe(
19   console.log(this.selectedFlight);
20   this.passengerform = this.fb.group( controlsConfig: {
21     quantities: this.fb.array( controlsConfig: [] ) ,
22   });
23   const initialValue = this.selectedFlight.passengers;
24   for (let i = 0; i < initialValue; i++) {
25     this.quantities().push(this.newQuantity());
26   }
27 }
28 ngOnInit(): void {
29 }
30
31 quantities(): FormArray {
```

Project

- home
  - flightsearch
    - flightsearch.component.css
    - flightsearch.component.html
    - flightsearch.component.ts
  - stories
    - stories.component.css
    - stories.component.html
    - stories.component.ts
  - home.component.css
  - home.component.html
  - home.component.ts- img
- Material
- searchlist
  - onewaylist
    - onewaylist.component.css
    - onewaylist.component.html
    - onewaylist.component.ts
  - roundtriplist
    - roundonward
    - roundreturn
    - roundselect.service.ts
    - roundtriplist.component.css
    - roundtriplist.component.html
    - roundtriplist.component.ts

onewaybooking.component.ts

```
9  selector: 'app-roundtripbooking',
10  templateUrl: './roundtripbooking.component.html',
11  styleUrls: ['./roundtripbooking.component.css', '../../../../../node_modu
12 }
13 export class RoundtripbookingComponent implements OnInit {
14   passengerform: FormGroup; subscription: Subscription; selectedFlight:
15   constructor(private fb: FormBuilder, private roundBooking: Roundbookin
16   this.subscription = this.roundBooking.currentMessage.subscribe( next
17   this.airportFetch.getOneAirport(this.selectedFlight.from).subscribe(
18   this.airportFetch.getOneAirport(this.selectedFlight.to).subscribe( r
19   console.log(this.selectedFlight);
20   this.passengerform = this.fb.group( controlsConfig: {
21     quantities: this.fb.array( controlsConfig: [] ) ,
22   });
23   const initialValue = this.selectedFlight.passengers;
24   for (let i = 0; i < initialValue; i++) {
25     this.quantities().push(this.newQuantity());
26   }
27 }
28 ngOnInit(): void {
29 }
30
31 quantities(): FormArray {
32   return this.passengerform.get('quantities') as FormArray;
```

Project Commit Pull Requests

The screenshot shows a code editor with the following project structure on the left:

- Services
  - airportfetch.service.ts
  - flightsearch.service.ts
  - managebooking.service.ts
  - onewaybooking.service.ts**
  - onewayquerydata.service.ts
  - roundbooking.service.ts
  - roundquerydata.service.ts
- test
- User
  - bookings
    - bookings.component.css
    - bookings.component.html
    - bookings.component.ts
    - dialogview.html
  - login
    - login.component.css
    - login.component.html
    - login.component.ts
  - profile
    - profile.component.css
    - profile.component.html
    - profile.component.ts
  - signup
    - signup.component.css
    - signup.component.html
    - signup.component.ts

The current file being edited is `onewaybooking.service.ts`. The code is as follows:

```
28     travelDate: '',
29     bookingAmount: 0
30   };
31   bookingactive = false;
32   baseUrl = 'https://localhost:5001/api/Ticket/';
33   private messageSource = new BehaviorSubject(this.oneWayQuery);
34   currentMessage = this.messageSource.asObservable();
35   constructor( private route: Router, private http: HttpClient, private ud
36               private snackBar: MatSnackBar) { }
37   // tslint:disable-next-line:typedef
38   changeMessage(data: any, flight: any){
39     this.oneWayQuery.from = data.from;
40     this.oneWayQuery.to = data.to;
41     this.oneWayQuery.dot = data.dot;
42     this.oneWayQuery.passengers = data.passengers;
43     this.oneWayQuery.flight = flight;
44     this.messageSource.next(this.oneWayQuery);
45     console.log(this.oneWayQuery);
46     this.bookingactive = true;
47     this.initiateBooking();
48   }
49   checkIfBookingInProgress(): void{
50     this.initiateBooking();
51   }
52   initiateBooking(): void{
```

# BACKEND

## AirportFetch

**GET** /api/AirportFetch

**GET** /api/AirportFetch/{id}

## Blog

**GET** /api/Blog

**POST** /api/Blog

**GET** /api/Blog/{id}

**PUT** /api/Blog/{id}

**DELETE** /api/Blog/{id}

## BookingID

**GET** /api/BookingID/{id}

**POST** /api/BookingID

## Bookings

**GET** /api/Bookings/{id}

**PUT** /api/Bookings/{id}

**DELETE** /api/Bookings/{id}

## FlightAdmin

**GET** /api/FlightAdmin

**POST** /api/FlightAdmin

**GET** /api/FlightAdmin/{id}

**PUT** /api/FlightAdmin/{id}

**DELETE** /api/FlightAdmin/{id}

## FlightQuery

**GET** /api/FlightQuery/{id}

**POST** /api/FlightQuery

## Session

**GET** /api/Session/{id}

**DELETE** /api/Session/{id}

**POST** /api/Session

## Ticket

**GET** /api/Ticket

**POST** /api/Ticket

**GET** /api/Ticket/{id}

**PUT** /api/Ticket/{id}

**DELETE** /api/Ticket/{id}

## User

**GET** /api/User

**POST** /api/User

**DELETE** /api/Ticket/{id}

## User

**GET** /api/User

**POST** /api/User

**GET** /api/User/{id}

**PUT** /api/User/{id}

**DELETE** /api/User/{id}

## UserBlogs

**GET** /api/UserBlogs/{id}

## UserLogin

**POST** /api/UserLogin

## UsernameCheckCotroller

**GET** /api/UsernameCheckCotroller/{id}

Now, On to the real  
thing!