How to Build Ionic 7 Firebase CRUD App with Angular

Published: December 1, 2023 — by Digamber Singh in Ionic

Ionic Angular Crud operations tutorial; A complete step by step tutorial on creating Ionic Angular CRUD operation mobile app using Firebase.

In this Ionic Firebase Angular tutorial, we will learn how to build CRUD (Create, Read, Update, Delete) app.

In this doctor's appointment CRUD mobile application, we will learn how to store app data in the Firebase Real-time NoSQL Database.

Firebase is a NoSQL Realtime database based on the document-oriented model and helps in building top-notch mobile and web applications.

Firebase stores and sync web and mobile app data quite smoothly. It offers tons of excellent features such as:

- Cloud storage
- Realtime update
- Easy A/B Testing
- Analytics Monitoring
- Authentication support
- Easy Server management
- Real-time communication
- Offline access to WEB SDK
- Hosting and cloud storage
- Push notifications support
- Straightforward app hosting
- Google Cloud IoT tools Integration support

This doctor's appointment Ionic Firebase CRUD app allows users to make an appointment.

User will be able to view all the appointments, update and delete the appointments under one roof.

We will need the following packages, frameworks, and tools to get started with this tutorial.

- Node
- Ionic
- Angular
- Text Editor
- Firebase Account
- Firebase NPM Library
- Command Line Tool or Terminal

Table of Contents

Generate Ionic Project

Run the following command to generate a brand new Ionic / Angular app.

```
ionic start ionic-firebase-crud blank --type=angular
```

Get inside the project.

```
cd ionic-firebase-crud
```

Run the below command to add the native core package.

```
npm install @ionic-native/core --legacy-peer-deps
```

Disable Strict Type Errors

To avoid TypeScript compiling issues, we just need to open **tsconfig.json** file:

First, set below property under **“compilerOptions”** property.

```
"compilerOptions": {
    "strictPropertyInitialization": false,
    "skipLibCheck": true
    ...
}
```

Secondly, add given props under **“angularCompilerOptions”**.

```
"angularCompilerOptions": {
    "strictTemplates": false,
    ...
}
```

Configure Firebase in Ionic and Angular

In order to make the Firebase work in our Ionic app, we must have a Firebase config key.

To get the config keys, go to console.firebase.google.com and register an account and get the Firebase config keys.

If you are a beginner, then check out my detailed article on Setting up Firebase Database in Ionic/Angular.

Install Firebase Module in Ionic Angular

Install the Firebase module in Ionic Angular project. Run the following command in the terminal.

```
npm install @angular/fire firebase@9.16.0 --legacy-peer-deps
```

Insert Firebase Config Keys in Ionic Angular

Insert the Firebase config keys in (production environment) **environment.ts** file.

```
export const environment = {
  production: false,
  firebaseConfig: {
    apiKey: "<your-api-key>",
    authDomain: "<your-auth-domain>",
    databaseURL: "<your-database-url>",
    projectId: "<your-cloud-firestore-project>",
    storageBucket: "<your-storage-bucket>",
    messagingSenderId: "<your-sender-id>",
    appId: "<your-app-id>",
    measurementId: "<your-measurement-id>"
  }
};
```

Import and Register Firebase Services in Ionic

Now we will import and register Firebase services and environment object in the **app.module.ts** file.

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouteReuseStrategy } from '@angular/router';
import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
import { AppComponent } from './app.component';
import { AppRoutingModule } from './app-routing.module';
// Firebase
import { AngularFireModule } from '@angular/fire/compat';
import { AngularFireAuthModule } from '@angular/fire/compat/auth';
import { AngularFireStorageModule } from '@angular/fire/compat/storage';
import { AngularFirestoreModule } from '@angular/fire/compat/firestore';
import { AngularFireDatabaseModule } from '@angular/fire/compat/database';
import { environment } from '../environments/environment';
@NgModule({
  declarations: [AppComponent],
  imports: [
    BrowserModule,
    IonicModule.forRoot(),
    AppRoutingModule,
    AngularFireModule.initializeApp(environment.firebaseConfig),
    AngularFireAuthModule,
    AngularFirestoreModule,
    AngularFireStorageModule,
    AngularFireDatabaseModule,
  ],
  providers: [{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy }],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

In this tutorial, we need only **AngularFireDatabaseModule** to be imported and registered in the AppModule class.

Setting Up Routes

In order to navigate in our Ionic Firebase app we need to configure routes. Run the following command to create pages in Ionic.

```
ng generate page make-appointment
ng generate page edit-appointment
```

To show the appointment list we will use the home page, the `make-appointment` page will be used to make the appointments.

In the `edit-appointment` page we will edit the appointments.

Open the `app/app-routing.module.ts` file and add the following code to configure the routes in app.

```
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
const routes: Routes = [
  {
    path: 'home',
    loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)
  },
  {
    path: '',
    redirectTo: 'home',
    pathMatch: 'full'
  },
  {
    path: 'make-appointment',
    loadChildren: () => import('./make-appointment/make-appointment.module').then
  },
  {
    path: 'edit-appointment/:id',
    loadChildren: () => import('./edit-appointment/edit-appointment.module').then
  },
];
@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

Next, open the `app.component.html` file and add the `ion-router-outlet` along with Ionic bottom tabs to activate the nav items.

```
<ion-app>
  <ion-router-outlet></ion-router-outlet>
</ion-app>
<ion-tabs>
  <ion-tab-bar slot="bottom">
    <ion-tab-button routerLinkActive="tab-selected" routerLink="/home" class="ios
      <ion-icon name="list-box"></ion-icon>
      <ion-label>Appointments</ion-label>
    </ion-tab-button>
    <ion-tab-button routerLinkActive="tab-selected" routerLink="/make-appointment
      <ion-icon name="add-circle"></ion-icon>
      <ion-label>Book Appointment</ion-label>
    </ion-tab-button>
  </ion-tab-bar>
</ion-tabs>
```

The **routerLinkActive=""** tag ads the active class on the selected navigation item in Ionic/Angular.

The **routerLink =""** tag helps in navigation from one page to another page in Ionic/Angular.

Create CRUD Operation with AngularFire

Next, create **shared/** folder, create **Appointment.ts** file, also add the given code into the file.

```
export class Appointment {
    $key: string;
    name: string;
    email: string
    mobile: number;
}
```

Execute command to create `shared/appointment` file inside of it.

```
ng generate service shared/appointment
```

Then, Include the following code inside **shared/appointment.service.ts**:

```
import { Injectable } from '@angular/core';
import { Appointment } from '../shared/Appointment';
import {
  AngularFireDatabase,
  AngularFireList,
  AngularFireObject,
} from '@angular/fire/compat/database';
@Injectable({
  providedIn: 'root',
})
export class AppointmentService {
  bookingListRef: AngularFireList<any>;
  bookingRef: AngularFireObject<any>;
  constructor(private db: AngularFireDatabase) {}
  // Create
  createBooking(apt: Appointment) {
    return this.bookingListRef.push({
      name: apt.name,
      email: apt.email,
      mobile: apt.mobile,
    });
  }
  // Get Single
  getBooking(id: string) {
    this.bookingRef = this.db.object('/appointment/' + id);
    return this.bookingRef;
  }
  // Get List
  getBookingList() {
    this.bookingListRef = this.db.list('/appointment');
    return this.bookingListRef;
  }
  // Update
  updateBooking(id: any, apt: Appointment) {
    return this.bookingRef.update({
      name: apt.name,
      email: apt.email,
      mobile: apt.mobile,
    });
  }
  // Delete
  deleteBooking(id: string) {
    this.bookingRef = this.db.object('/appointment/' + id);
    this.bookingRef.remove();
  }
}
```

Adding Item in Firebase Database

We will show you how to add items to the Firebase Realtime database but first to work get the user information we need to create a form so to work with forms you must import and register **ReactiveFormsModule** in **make-appointment.module.ts** file.

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
@NgModule({
  imports: [
    ReactiveFormsModule,
    FormsModule,
  ]
})
export class MakeAppointmentPageModule { }
```

Next, add the following code in **make-appointment.page.ts** file.

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { FormGroup, FormBuilder } from '@angular/forms';
import { AppointmentService } from './../shared/appointment.service';
@Component({
  selector: 'app-make-appointment',
  templateUrl: './make-appointment.page.html',
  styleUrls: ['./make-appointment.page.scss'],
})
export class MakeAppointmentPage implements OnInit {
  bookingForm: FormGroup;
  constructor(
    private aptService: AppointmentService,
    private router: Router,
    public fb: FormBuilder
  ) {}
  ngOnInit() {
    this.bookingForm = this.fb.group({
      name: [''],
      email: [''],
      mobile: [''],
    });
  }
  formSubmit() {
    if (!this.bookingForm.valid) {
      return false;
    } else {
      return this.aptService
        .createBooking(this.bookingForm.value)
        .then((res) => {
          console.log(res);
          this.bookingForm.reset();
          this.router.navigate(['/home']);
        })
        .catch((error) => console.log(error));
    }
  }
}
```

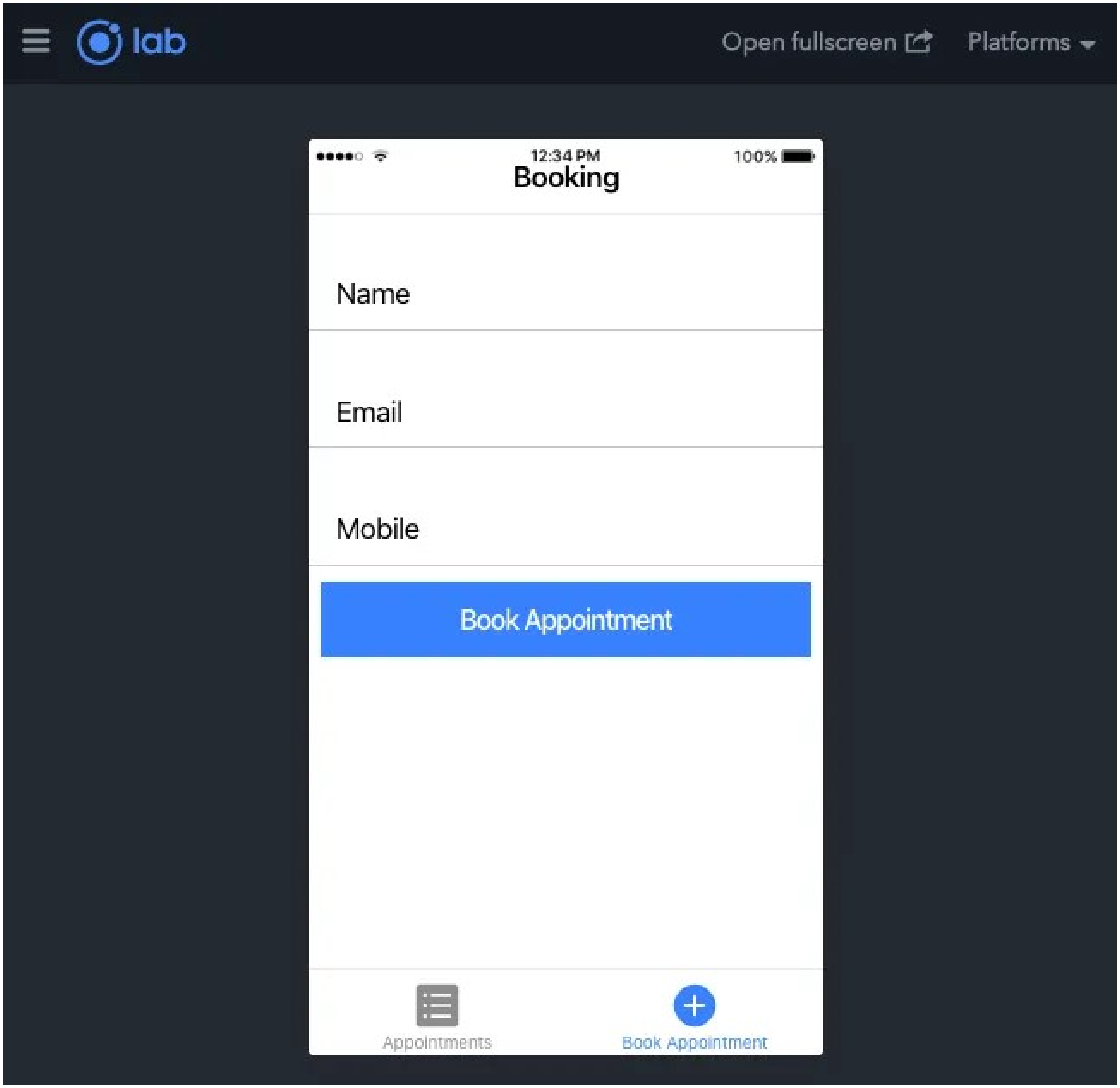
To set up the form first import FormGroup and FormBuilder Reactive Forms services in Ionic. Define a variable with FormGroup inject the FormBuilder in the constructor and create a FormControl inside the **this.bookingForm** using the FormBuilder object.

Next, **AppointmentService** class, inject inside the constructor. Submit the form values by accessing the AppointmentService CRUD api and redirect the user to the appointments list page using Angular Router service.

Next, Use the *ngFor directive to create the appointment form using Ionic UI components. Open `make-appointment.page.html` file and add the following code.

```
<ion-header>
  <ion-toolbar class="ios hydrated">
    <ion-title class="ios title-ios hydrated">Booking</ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>
  <ion-list lines="full">
    <form [formGroup]="bookingForm" (ngSubmit)="formSubmit()">
      <ion-item>
        <ion-label position="floating">Name</ion-label>
        <ion-input formControlName="name" type="text" required></ion-input>
      </ion-item>
      <ion-item>
        <ion-label position="floating">Email</ion-label>
        <ion-input formControlName="email" type="text" required>
        </ion-input>
      </ion-item>
      <ion-item>
        <ion-label position="floating">Mobile</ion-label>
        <ion-input formControlName="mobile" type="text" required>
        </ion-input>
      </ion-item>
```

```
        <ion-row>
          <ion-col>
            <ion-button type="submit" color="primary" shape="full" expand="block">I
          </ion-col>
        </ion-row>
      </form>
    </ion-list>
  </ion-content>
```


lab
Open fullscreen
Platforms
12:34 PM
100%
Booking
Name
Email
Mobile
Book Appointment
Appointments
Book Appointment

Delete Item from List of Data

To display the list import AppointmentService and then inject in the constructor.

Get the appointment list via **getBookingList()** function.

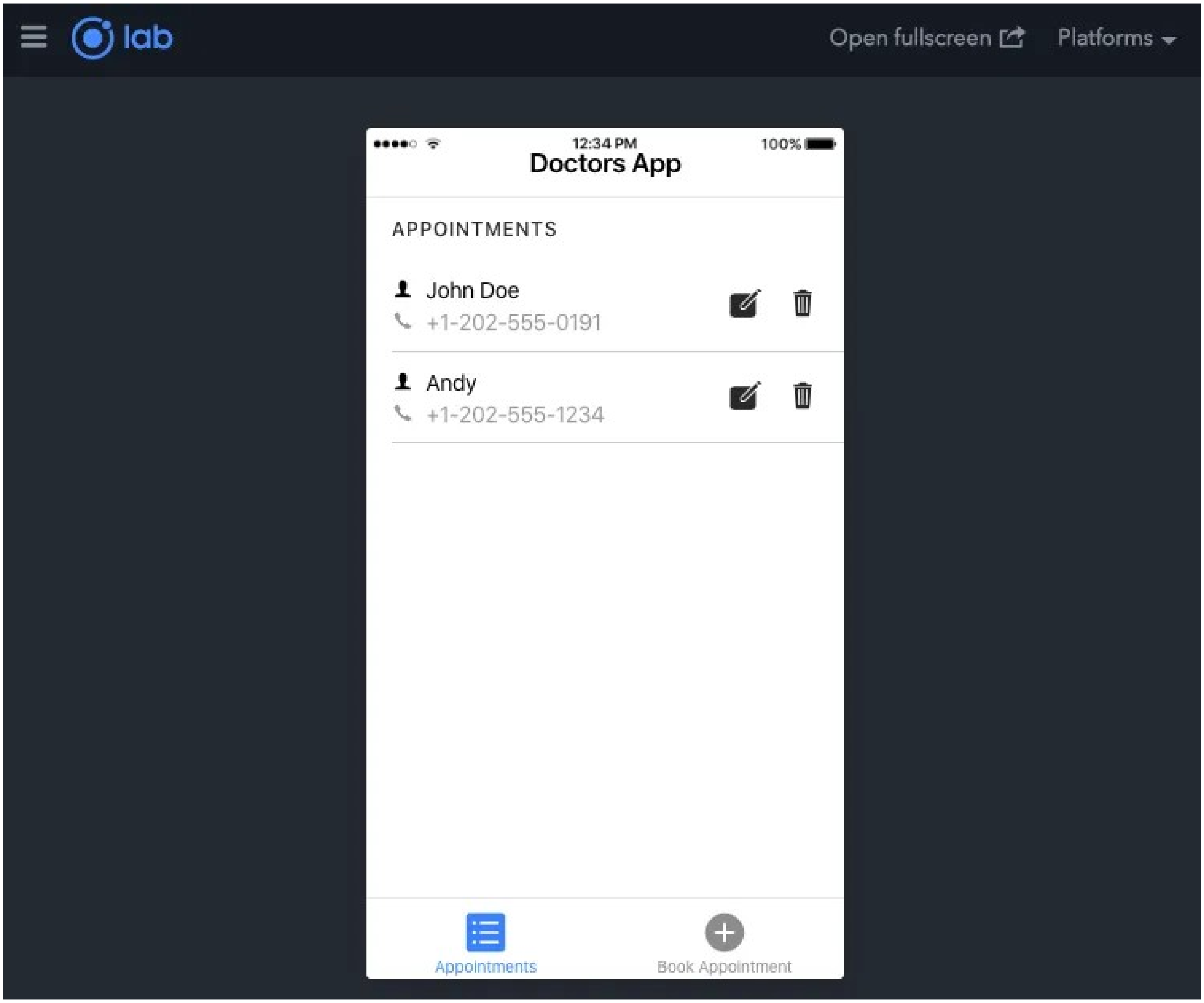
To remove the item from the database we are using **deleteBooking(id)** method and passing the item id or key in it.

Open `home.page.ts` file and add the following code.

```
import { Component, OnInit } from '@angular/core';
import { Appointment } from '../shared/Appointment';
import { AppointmentService } from './../shared/appointment.service';
@Component({
  selector: 'app-home',
  templateUrl: 'home.page.html',
  styleUrls: ['home.page.scss'],
})
export class HomePage implements OnInit {
  Bookings: any = [];
  constructor(private aptService: AppointmentService) {}
  ngOnInit() {
    this.fetchBookings();
    let bookingRes = this.aptService.getBookingList();
    bookingRes.snapshotChanges().subscribe((res) => {
      this.Bookings = [];
      res.forEach((item) => {
        let a: any = item.payload.toJSON();
        a['$key'] = item.key;
        this.Bookings.push(a as Appointment);
      });
    });
  }
  fetchBookings() {
    this.aptService
      .getBookingList()
      .valueChanges()
      .subscribe((res) => {
        console.log(res);
      });
  }
  deleteBooking(id: any) {
    console.log(id);
    if (window.confirm('Do you really want to delete?')) {
      this.aptService.deleteBooking(id);
    }
  }
}
```

To display items open **home.page.html** file and insert the following code in it.

```
<ion-header>
  <ion-toolbar>
    <ion-title>
      Doctors App
    </ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>
  <ion-list class="ios list-ios hydrated">
    <ion-list-header class="ios hydrated">
      Appointments
    </ion-list-header>
    <ion-item *ngFor="let booking of Bookings" class="user-list">
      <ion-label>
        <h5>
          <ion-icon name="person"></ion-icon> {{booking.name}}
        </h5>
        <p>
          <ion-icon name="call"></ion-icon> {{booking.mobile}}
        </p>
      </ion-label>
      <div class="item-note" item-end>
        <button ion-button clear [routerLink]="['/edit-appointment/', booking.$ke
          <ion-icon name="create" style="zoom:2.0"></ion-icon>
        </button>
        <button ion-button clear (click)="deleteBooking(booking.$key)">
          <ion-icon name="trash" style="zoom:2.0"></ion-icon>
        </button>
      </div>
    </ion-item>
  </ion-list>
```



Edit Item in Ionic

To edit the item import AppointmentService and then inject into the constructor class. Import the **ReactiveFormsModule** in `edit-appointment.module.ts` file.

```
import { FormsModule, ReactiveFormsModule } from '@angular/forms';
@NgModule({
  imports: [
    ReactiveFormsModule,
    FormsModule,
  ]
})
export class MakeAppointmentPageModule { }
```

Next, Insert the following code in `edit-appointment.page.ts` file.

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from "@angular/router";
import { FormGroup, FormBuilder } from "@angular/forms";
import { AppointmentService } from './../shared/appointment.service';
@Component({
  selector: 'app-edit-appointment',
  templateUrl: './edit-appointment.page.html',
  styleUrls: ['./edit-appointment.page.scss'],
})
export class EditAppointmentPage implements OnInit {
  updateBookingForm: FormGroup;
  id: any;
  constructor(
    private aptService: AppointmentService,
    private actRoute: ActivatedRoute,
    private router: Router,
    public fb: FormBuilder
  ) {
    this.id = this.actRoute.snapshot.paramMap.get('id');
    this.aptService.getBooking(this.id).valueChanges().subscribe(res => {
      this.updateBookingForm.setValue(res);
    });
  }
  ngOnInit() {
    this.updateBookingForm = this.fb.group({
      name: [''],
      email: [''],
      mobile: ['']
    })
    console.log(this.updateBookingForm.value)
  }
  updateForm() {
    this.aptService.updateBooking(this.id, this.updateBookingForm.value)
      .then(() => {
        this.router.navigate(['/home']);
      })
      .catch(error => console.log(error));
  }
}
```

To show the edit form, add the following Ionic HTML code in `edit-appointment.page.html` file.

```
<ion-header>
  <ion-toolbar>
    <ion-title>
      Edit Appointment
    </ion-title>
  </ion-toolbar>
</ion-header>
<ion-content>
  <ion-list lines="full">
    <form [formGroup]="updateBookingForm" (ngSubmit)="updateForm()">
      <ion-item>
        <ion-label position="floating">Name</ion-label>
        <ion-input formControlName="name" type="text" required></ion-input>
      </ion-item>
      <ion-item>
        <ion-label position="floating">Email</ion-label>
        <ion-input formControlName="email" type="text" required>
        </ion-input>
      </ion-item>
      <ion-item>
        <ion-label position="floating">Mobile</ion-label>
        <ion-input formControlName="mobile" type="text" required>
        </ion-input>
      </ion-item>
      <ion-row>
        <ion-col>
          <ion-button type="submit" color="primary" shape="full" expand="block">l
        </ion-col>
      </ion-row>
    </form>
  </ion-list>
</ion-content>
```

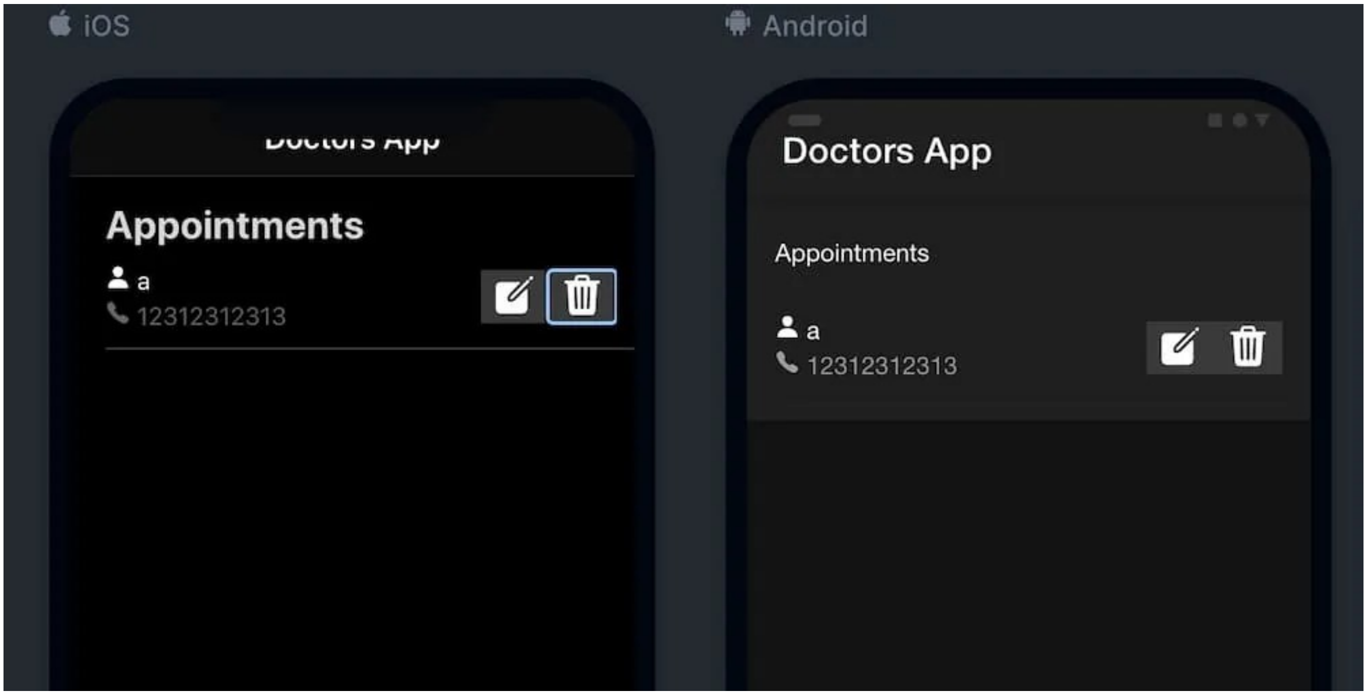
To start the app, you may try the following commands:

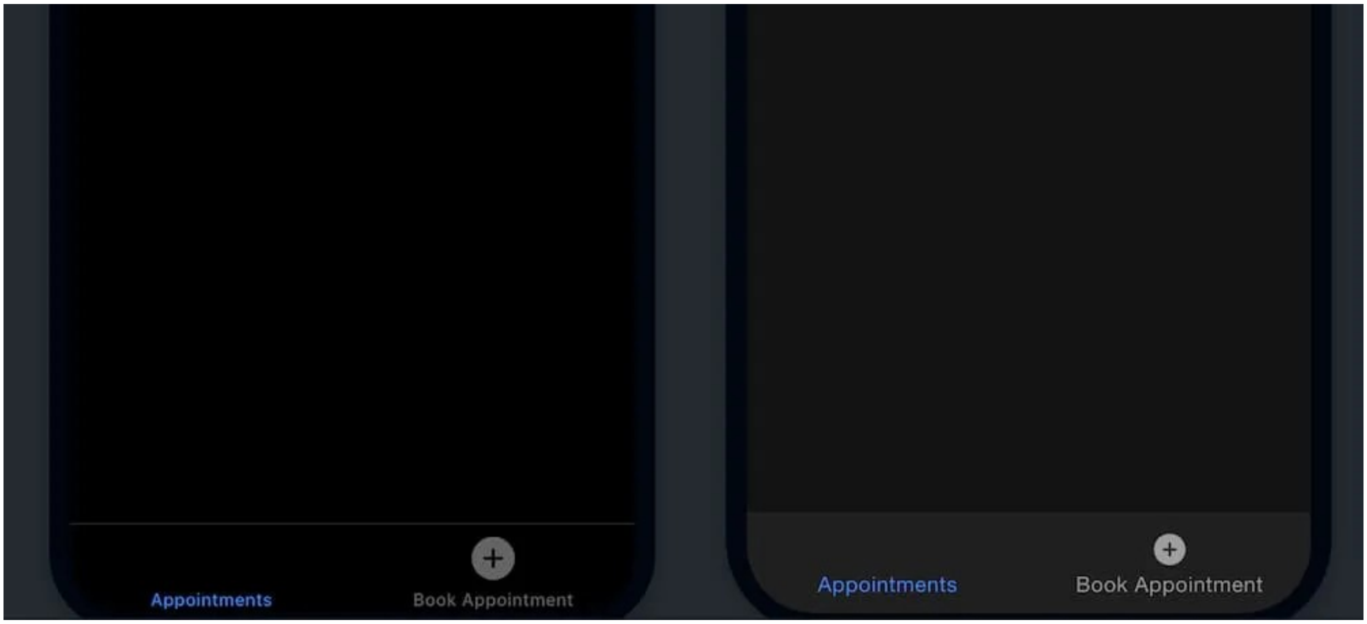
Run the following command to install lab mode.

```
npm i @ionic/lab --legacy-peer-deps
```

Start the Ionic Firebase CRUD project.

```
ionic serve -l
```





Conclusion

We have completed Ionic Firebase tutorial in which we learned how to create CRUD mobile app with Angular.

In this tutorial we focused on how to set Firebase in Ionic and Angular, how to define CRUD operations in Angular service.

Finally, we also checked out how to consume Firebase REST APIs to perform CRUD operation.

Digamber Singh

Digamber Singh is a freelance Front-end engineer, educator, and fitness enthusiast. He's spent over a decade working as a web developer for organizations like Emids, HeaderLabs, Ketchupp.

Follow me:

Comments

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Post Comment

HTML 5, CSS 3, JavaScript, PHP, React, Angular, Laravel — Aimed to offer custom coding solutions on almost every modern programming language.

Explore

About

Affiliate Disclosure

Privacy Policy

Disclaimer

Contact

Categories

Angular

Vue

React

Ionic

Laravel

Codeigniter

PHP