

Start new blank Ionic Project

Install firebase angularFire

<https://github.com/angular/angularfire>

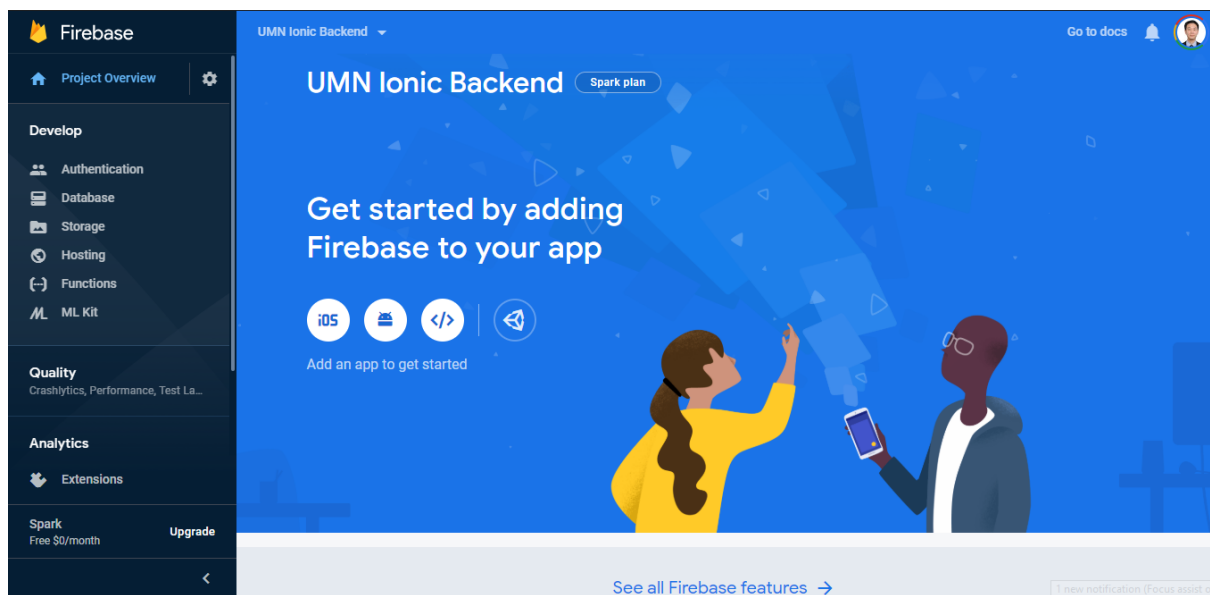
```
D:\ionic-course-umn\ionic-backend> npm install firebase @angular/fire --save
```

Buat page

```
D:\ionic-course-umn\ionic-backend> ionic g page pages/todoDetails
```

Buat service todo

Buat Project Baru di Firebase



Copas bagian ini

2

## Add Firebase SDK

Copy and paste these scripts into the bottom of your <body> tag, but before you use any Firebase services:

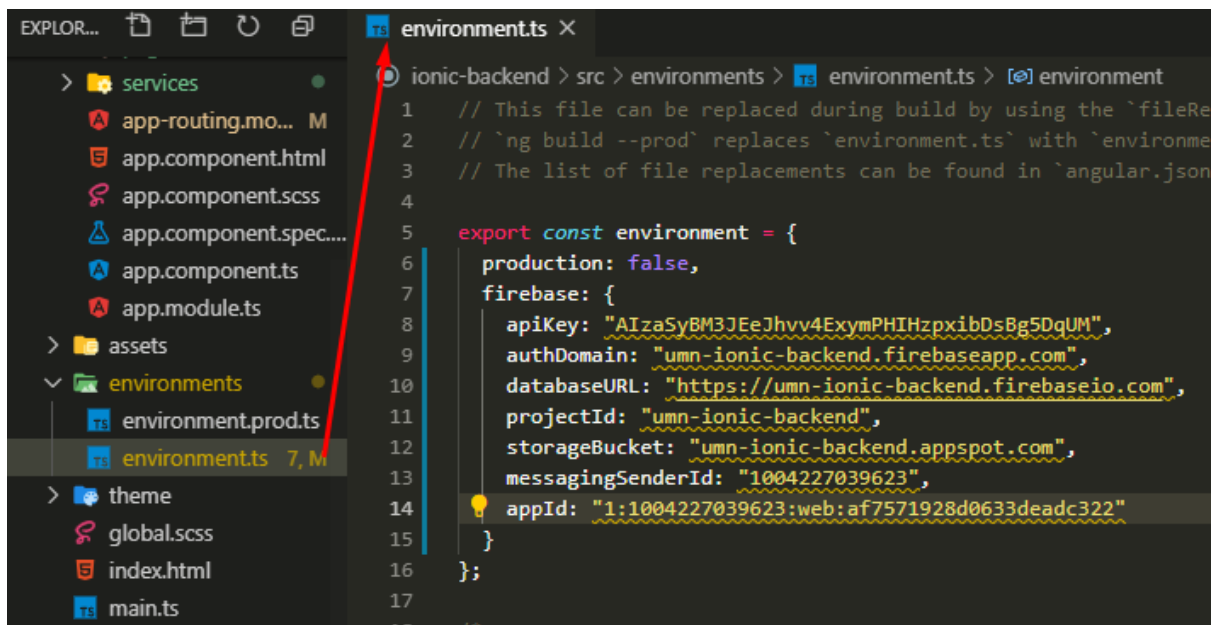
```
<!-- The core Firebase JS SDK is always required and must be listed first -->
<script src="https://www.gstatic.com/firebasejs/7.2.2/firebase-app.js"></script>

<!-- TODO: Add SDKs for Firebase products that you want to use
      https://firebase.google.com/docs/web/setup#available-libraries -->

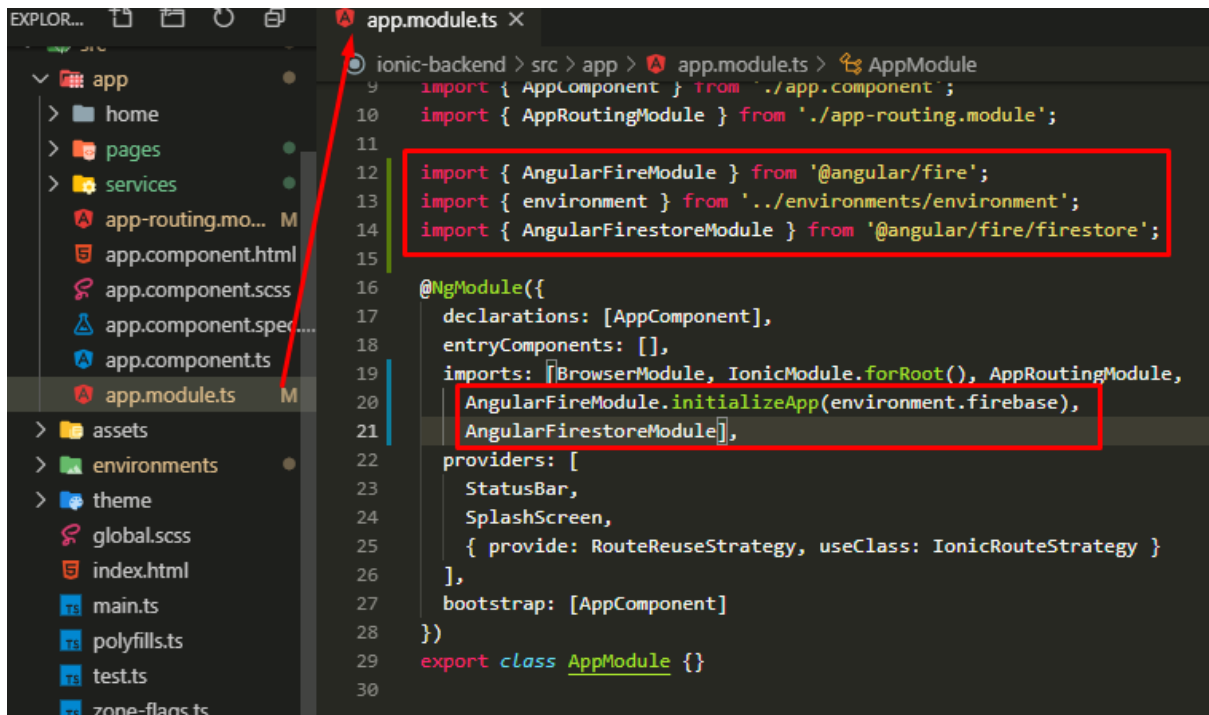
<script>
  // Your web app's Firebase configuration
  var firebaseConfig = {
    apiKey: "AIzaSyBM3JEeJhvv4ExymPHIHpxibDsBg5DqUM",
    authDomain: "umn-ionic-backend.firebaseio.com",
    databaseURL: "https://umn-ionic-backend.firebaseio.com",
    projectId: "umn-ionic-backend",
    storageBucket: "umn-ionic-backend.appspot.com",
    messagingSenderId: "1004227039623",
    appId: "1:1004227039623:web:af7571928d0633deadc322"
  };
  // Initialize Firebase
  firebase.initializeApp(firebaseConfig);
</script>
```

Learn more about Firebase for web: [Get Started](#), [Web SDK API Reference](#), [Samples](#)

## Set environment



Tambahkan kode berikut



```
ionic-backend > src > app > app.module.ts > AppModule
9 import { AppComponent } from './app.component';
10 import { AppRoutingModuleModule } from './app-routing.module';
11
12 import { AngularFireModule } from '@angular/fire';
13 import { environment } from '../environments/environment';
14 import { AngularFirestoreModule } from '@angular/fire/firestore';
15
16 @NgModule({
17   declarations: [AppComponent],
18   entryComponents: [],
19   imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModuleModule,
20     AngularFireModule.initializeApp(environment.firebase),
21     AngularFirestoreModule],
22   providers: [
23     StatusBar,
24     SplashScreen,
25     { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
26   ],
27   bootstrap: [AppComponent]
28 })
29 export class AppModule {}
30
```

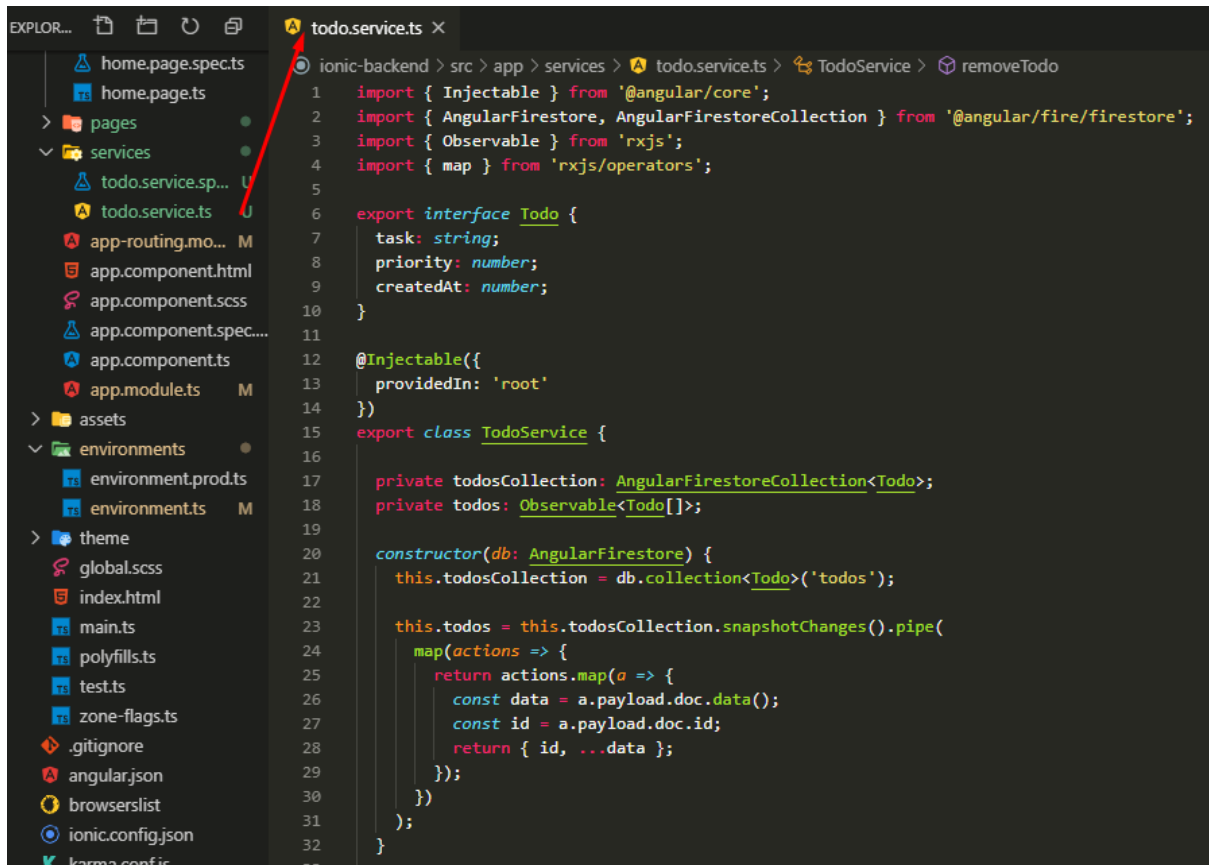
Ubah routing menjadi



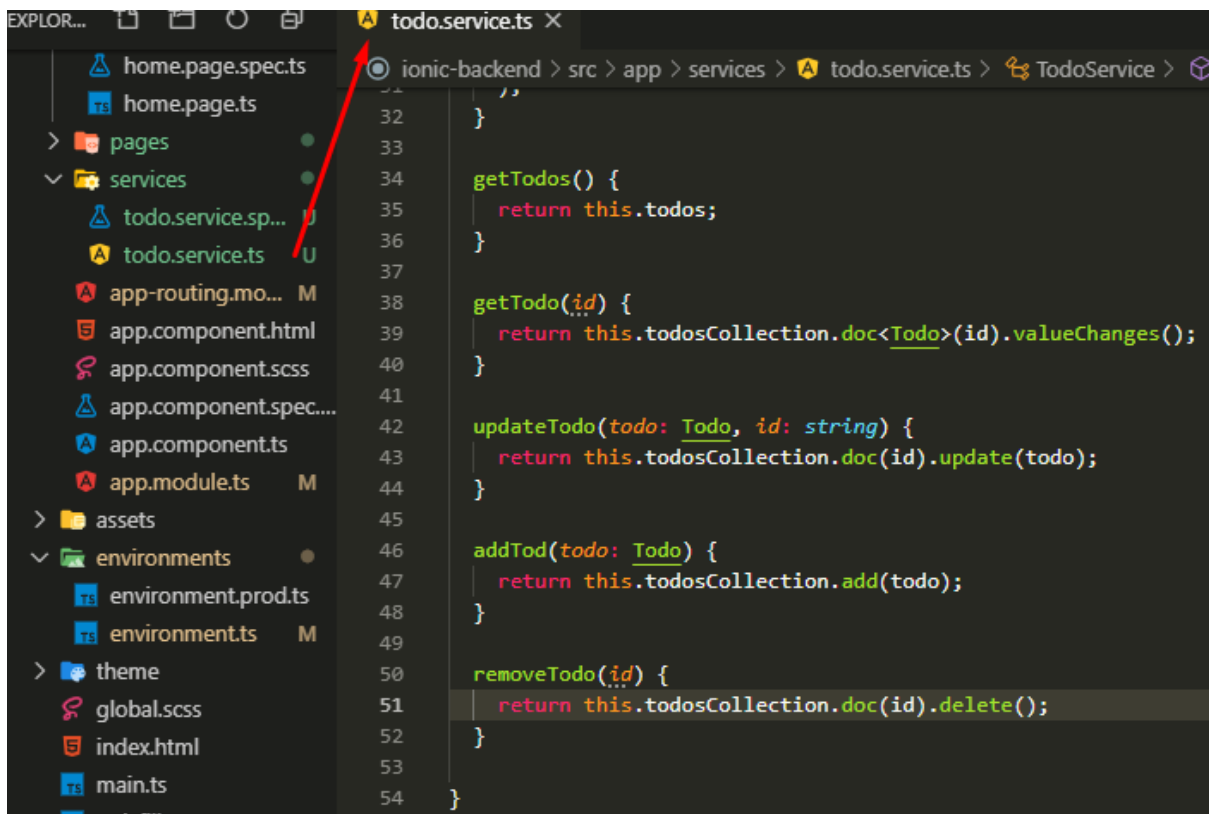
```
ionic-backend > src > app > app-routing.module.ts > routes
1 import { NgModule } from '@angular/core';
2 import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   { path: '', redirectTo: 'home', pathMatch: 'full' },
6   { path: 'home', loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)},
7   { path: 'details', loadChildren: './pages/todo-details/todo-details.module#TodoDetailsPageModule' },
8   { path: 'details/:id', loadChildren: './pages/todo-details/todo-details.module#TodoDetailsPageModule' },
9 ];
10
11 @NgModule({

```

## Todo service

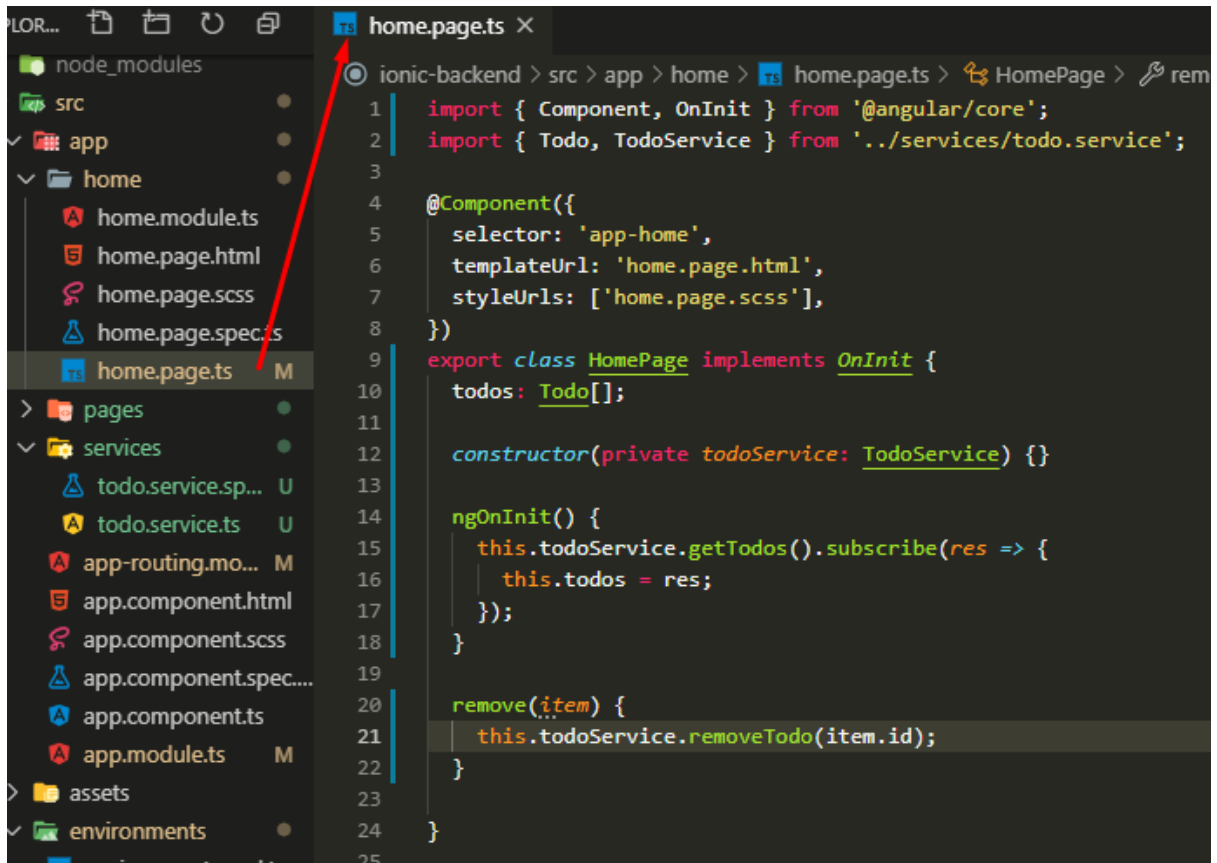


```
EXPLORER... ionic-backend > src > app > services > todo.service.ts > TodoService > removeTodo
1 import { Injectable } from '@angular/core';
2 import { AngularFireStore, AngularFireStoreCollection } from '@angular/fire/firestore';
3 import { Observable } from 'rxjs';
4 import { map } from 'rxjs/operators';
5
6 export interface Todo {
7   task: string;
8   priority: number;
9   createdAt: number;
10 }
11
12 @Injectable({
13   providedIn: 'root'
14 })
15 export class TodoService {
16
17   private todosCollection: AngularFireStoreCollection<Todo>;
18   private todos: Observable<Todo[]>;
19
20   constructor(db: AngularFireStore) {
21     this.todosCollection = db.collection<Todo>('todos');
22
23     this.todos = this.todosCollection.snapshotChanges().pipe(
24       map(actions => {
25         return actions.map(a => {
26           const data = a.payload.doc.data();
27           const id = a.payload.doc.id;
28           return { id, ...data };
29         });
30       })
31     );
32   }
33 }
```



```
EXPLORER... ionic-backend > src > app > services > todo.service.ts > TodoService >
31
32   },
33 }
34
35 getTodos() {
36   return this.todos;
37 }
38
39 getTodo(id) {
40   return this.todosCollection.doc<Todo>(id).valueChanges();
41 }
42
43 updateTodo(todo: Todo, id: string) {
44   return this.todosCollection.doc(id).update(todo);
45 }
46
47 addTod(todo: Todo) {
48   return this.todosCollection.add(todo);
49 }
50
51 removeTodo(id) {
52   return this.todosCollection.doc(id).delete();
53 }
54 }
```

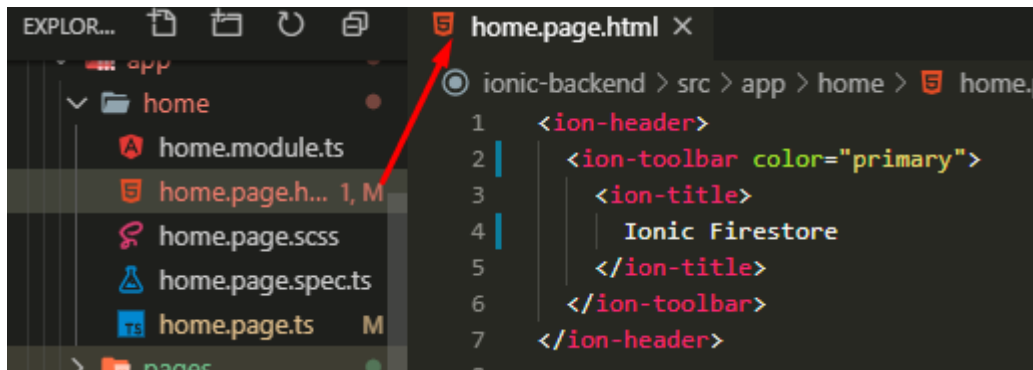
## Homepage.ts



```
ionic-backend > src > app > home > home.page.ts > HomePage > remove

1  import { Component, OnInit } from '@angular/core';
2  import { Todo, TodoService } from '../services/todo.service';
3
4  @Component({
5    selector: 'app-home',
6    templateUrl: 'home.page.html',
7    styleUrls: ['home.page.scss'],
8  })
9  export class HomePage implements OnInit {
10    todos: Todo[];
11
12    constructor(private todoService: TodoService) {}
13
14    ngOnInit() {
15      this.todoService.getTodos().subscribe(res => {
16        this.todos = res;
17      });
18    }
19
20    remove(item) {
21      this.todoService.removeTodo(item.id);
22    }
23
24  }
25
```

## Home.page.html



```
EXPLOR... ionic-backend > src > app > home > home.page.html > home.page.html

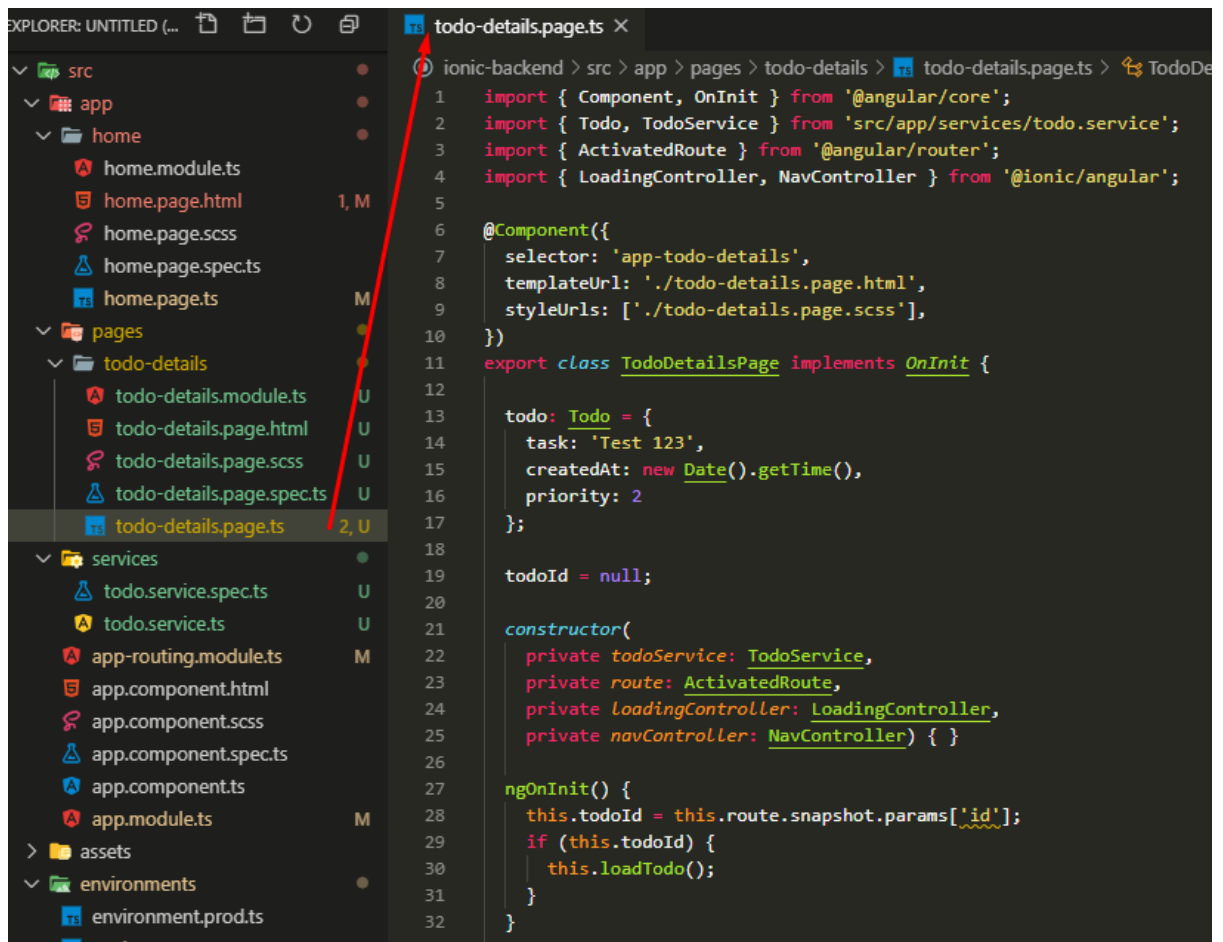
1  <ion-header>
2    <ion-toolbar color="primary">
3      <ion-title>
4        Ionic Firestore
5      </ion-title>
6    </ion-toolbar>
7  </ion-header>
8
```

EXPLOR... home.module.ts home.page.html x

ionic-backend > src > app > home > home.page.html > ion-content > ion-fab

```
9 <ion-content>
10
11 <ion-list>
12   <ng-container *ngIf="!todos || todos.length == 0">
13     <div *ngFor="let n of [0,1,2]" padding>
14       <ion-skeleton-text></ion-skeleton-text>
15     <p>
16       <ion-skeleton-text class="fake-skeleton"></ion-skeleton-text>
17     </p>
18   </div>
19 </ng-container>
20
21 <ion-item-sliding *ngFor="let item of todos">
22   <ion-item lines="inset" button [routerLink]="['/details', item.id]">
23     <ion-label>
24       {{ item.task }}
25       <p>{{ item.createdAt | date:'short' }}</p>
26     </ion-label>
27     <ion-note slot="end" color="primary">{{ item.priority }}</ion-note>
28   </ion-item>
29
30   <ion-item-options side="end">
31     <ion-item-option (click)="remove(item)" color="secondary">
32       <ion-checkbox><ion-icon name="checkmark" slot="end"></ion-icon>
33     </ion-item-option>
34   </ion-item-options>
35 </ion-item-sliding>
36 </ion-list>
37
38 <ion-fab vertical="bottom" horizontal="end" slot="fixed">
39   <ion-fab-button routerLink="/details" routerDirection="forward">
40     <ion-icon name="add"></ion-icon>
41   </ion-fab-button>
42 </ion-fab>
43 </ion-content>
```

## Todo Details page



The screenshot shows an IDE with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'src', 'app', 'home', 'pages', 'services', 'assets', and 'environments'. The 'pages' folder is expanded, showing 'todo-details' with files like 'todo-details.module.ts', 'todo-details.page.html', 'todo-details.page.scss', 'todo-details.page.spec.ts', and 'todo-details.page.ts'. The 'todo-details.page.ts' file is selected, and its code is displayed in the editor. A red arrow points from the file explorer to the code editor.

```
EXPLORER: UNTITLED (...)
```

```
src
├── app
│   ├── home
│   │   ├── home.module.ts
│   │   ├── home.page.html
│   │   ├── home.page.scss
│   │   ├── home.page.spec.ts
│   │   └── home.page.ts
│   ├── pages
│   │   ├── todo-details
│   │   │   ├── todo-details.module.ts
│   │   │   ├── todo-details.page.html
│   │   │   ├── todo-details.page.scss
│   │   │   ├── todo-details.page.spec.ts
│   │   │   └── todo-details.page.ts
│   ├── services
│   │   ├── todo.service.spec.ts
│   │   ├── todo.service.ts
│   │   ├── app-routing.module.ts
│   │   ├── app.component.html
│   │   ├── app.component.scss
│   │   ├── app.component.spec.ts
│   │   ├── app.component.ts
│   │   └── app.module.ts
│   ├── assets
│   └── environments
│       ├── environment.prod.ts
│       └── environment.ts
```

```
ionic-backend > src > app > pages > todo-details > todo-details.page.ts > TodoDe
1  import { Component, OnInit } from '@angular/core';
2  import { Todo, TodoService } from 'src/app/services/todo.service';
3  import { ActivatedRoute } from '@angular/router';
4  import { LoadingController, NavController } from '@ionic/angular';
5
6  @Component({
7    selector: 'app-todo-details',
8    templateUrl: './todo-details.page.html',
9    styleUrls: ['./todo-details.page.scss'],
10  })
11  export class TodoDetailsPage implements OnInit {
12
13    todo: Todo = {
14      task: 'Test 123',
15      createdAt: new Date().getTime(),
16      priority: 2
17    };
18
19    todoId = null;
20
21    constructor(
22      private todoService: TodoService,
23      private route: ActivatedRoute,
24      private loadingController: LoadingController,
25      private navCtrl: NavController) { }
26
27    ngOnInit() {
28      this.todoId = this.route.snapshot.params['id'];
29      if (this.todoId) {
30        this.loadTodo();
31      }
32    }
33  }
```

EDITOR: UNTITLED (...)

src

- app
  - home
    - home.module.ts
    - home.page.html 1, M
    - home.page.scss
    - home.page.spec.ts
    - home.page.ts M
  - pages
    - todo-details
      - todo-details.module.ts U
      - todo-details.page.html U
      - todo-details.page.scss U
      - todo-details.page.spec.ts U
      - todo-details.page.ts 2, U
  - services
    - todo.service.spec.ts U
    - todo.service.ts U
  - app-routing.module.ts M
  - app.component.html
  - app.component.scss
  - app.component.spec.ts
  - app.component.ts
  - app.module.ts M- assets
- environments
  - environment.prod.ts
  - environment.ts M

ionic-backend > src > app > pages > todo-details > todo-details.page.ts > TodoDetails

```
34 async loadTodo() {
35   const loading = await this.loadingController.create({
36     message: 'Loading Todo...'
37   });
38   await loading.present();
39
40   this.todoService.getTodo(this.todoId).subscribe(res => {
41     loading.dismiss();
42     this.todo = res;
43   });
44 }
45
46 async saveTodo() {
47   const loading = await this.loadingController.create({
48     message: 'Saving Todo...'
49   });
50   await loading.present();
51
52   if (this.todoId) {
53     this.todoService.updateTodo(this.todo, this.todoId).then(() => {
54       loading.dismiss();
55       this.navController.navigateBack('home');
56     });
57   } else {
58     this.todoService.addTodo(this.todo).then(() => {
59       loading.dismiss();
60       this.navController.navigateBack('home');
61     });
62   }
63 }
64
65 }
```



Coba di browser

The screenshot shows a web browser with two tabs. The left tab is the Firebase console, displaying the 'Database' section for 'UMN Ionic Backend'. It shows a collection named 'todos' with a document ID '9bM1r3sF2K2G07Ud6SYc'. The document contains the following data:

Field	Value
createdAt	1572382309714
priority	1
task	"Beli Bubur Ayam"

The right tab shows a local Ionic application running on 'localhost:8100/home'. It displays a list of items from the 'Ionic Firestore' database:

Item	Count
Test 123 10/30/19, 3:48 AM	2
Beli Bubur Ayam 10/30/19, 3:51 AM	1

A blue circular button with a white plus sign is visible in the bottom right corner of the browser window.

With PHP & MySQL

New Project Blank

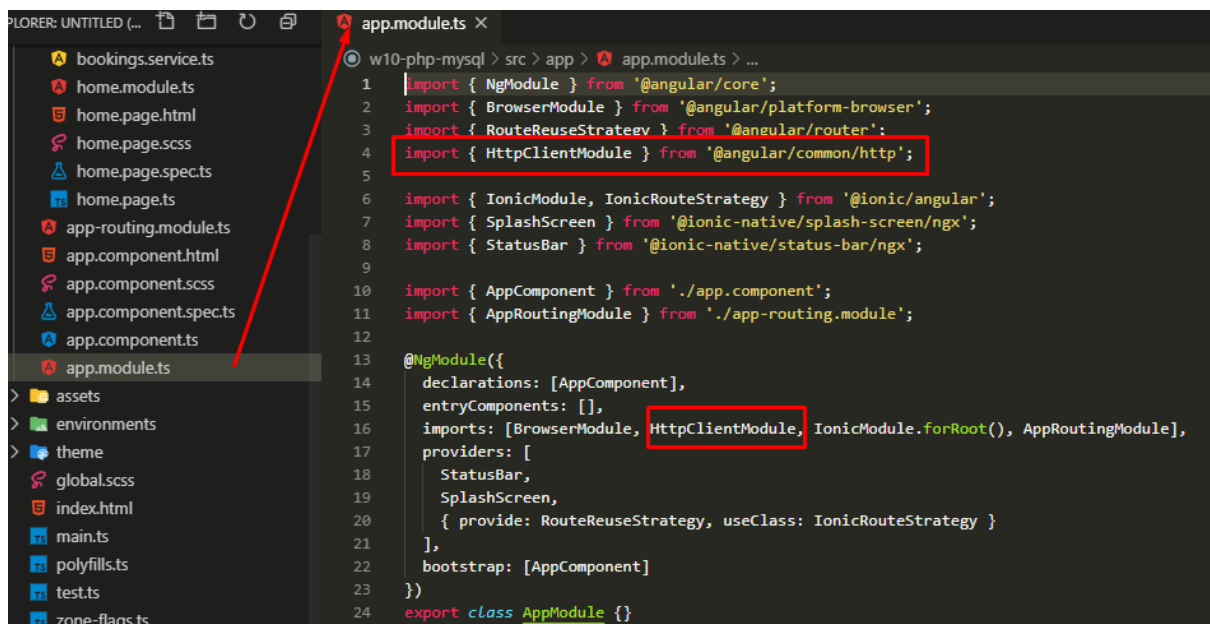
Generate page

```
D:\ionic-course-umn\w10-php-mysql> ionic generate page home/new-booking
```

Buat service

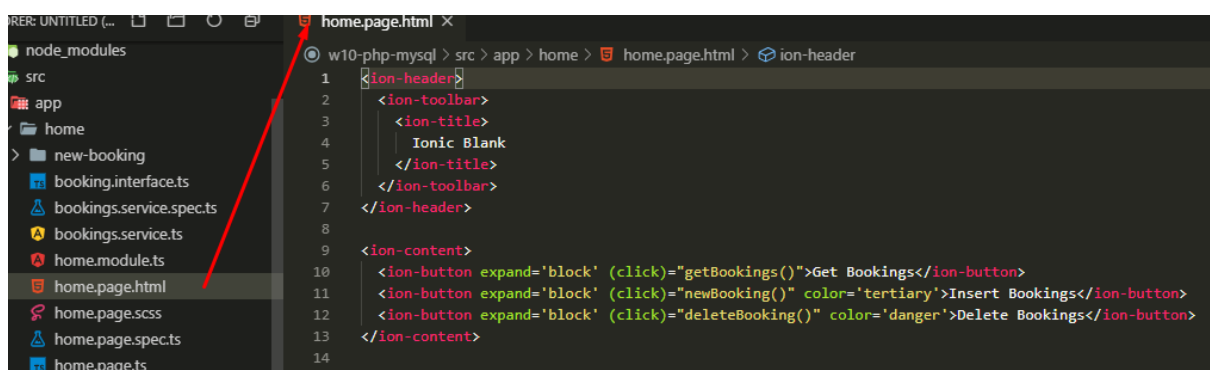
```
D:\ionic-course-umn\w10-php-mysql> ionic generate service home/bookings
```

App.module.ts



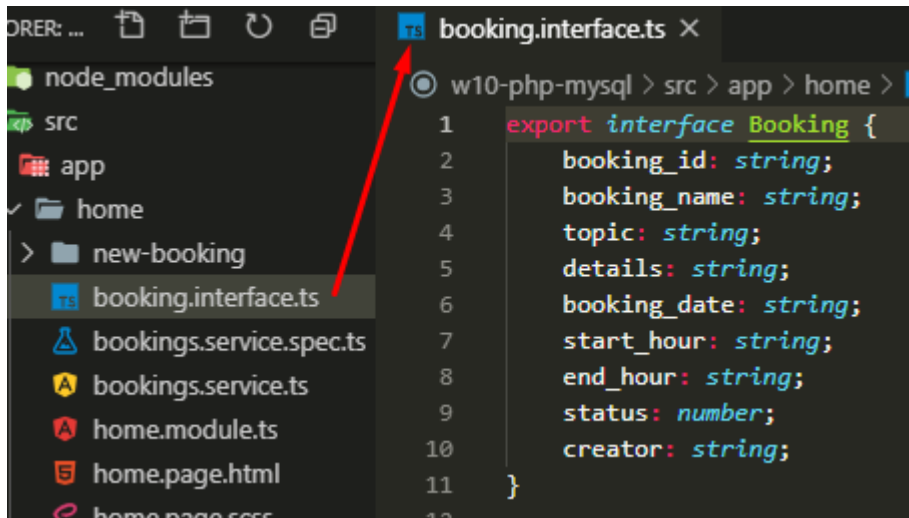
```
app.module.ts
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { RouteReuseStrategy } from '@angular/router';
4 import { HttpClientModule } from '@angular/common/http';
5
6 import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
7 import { SplashScreen } from '@ionic-native/splash-screen/ngx';
8 import { StatusBar } from '@ionic-native/status-bar/ngx';
9
10 import { AppComponent } from './app.component';
11 import { AppRoutingModule } from './app-routing.module';
12
13 @NgModule({
14   declarations: [AppComponent],
15   entryComponents: [],
16   imports: [BrowserModule, HttpClientModule, IonicModule.forRoot(), AppRoutingModule],
17   providers: [
18     StatusBar,
19     SplashScreen,
20     { provide: RouteReuseStrategy, useClass: IonicRouteStrategy }
21   ],
22   bootstrap: [AppComponent]
23 })
24 export class AppModule {}
```

Home.page.html



```
home.page.html
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>
4       Ionic Blank
5     </ion-title>
6   </ion-toolbar>
7 </ion-header>
8
9 <ion-content>
10   <ion-button expand='block' (click)="getBookings()">Get Bookings</ion-button>
11   <ion-button expand='block' (click)="newBooking()" color='tertiary'>Insert Bookings</ion-button>
12   <ion-button expand='block' (click)="deleteBooking()" color='danger'>Delete Bookings</ion-button>
13 </ion-content>
14
```

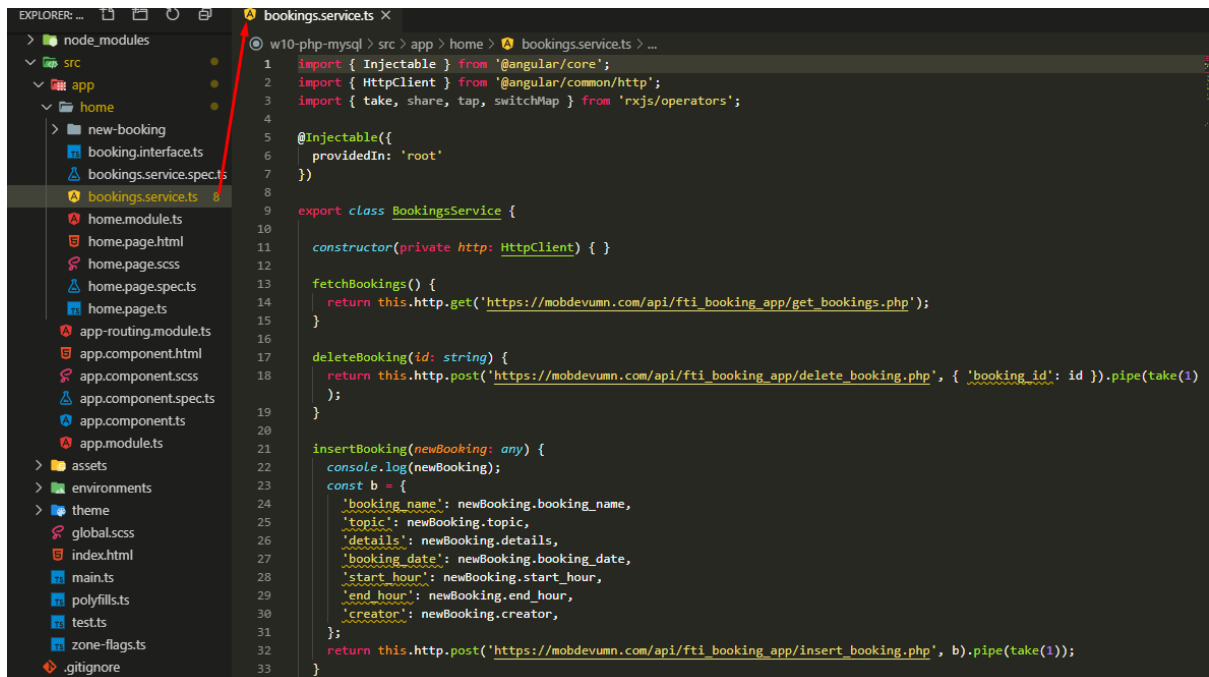
Buat file booking interface



The screenshot shows the VS Code interface with the Explorer sidebar on the left. The file explorer shows a project structure with folders 'node\_modules', 'src', 'app', and 'home'. Inside 'home', there is a 'new-booking' folder containing 'booking.interface.ts', 'bookings.service.spec.ts', 'bookings.service.ts', 'home.module.ts', 'home.page.html', and 'home.page.scss'. The 'booking.interface.ts' file is selected and highlighted. The main editor shows the content of 'booking.interface.ts', which defines an interface named 'Booking' with the following properties: 'booking\_id: string;', 'booking\_name: string;', 'topic: string;', 'details: string;', 'booking\_date: string;', 'start\_hour: string;', 'end\_hour: string;', 'status: number;', and 'creator: string;'. A red arrow points from the file explorer to the file name in the editor.

```
1 export interface Booking {
2     booking_id: string;
3     booking_name: string;
4     topic: string;
5     details: string;
6     booking_date: string;
7     start_hour: string;
8     end_hour: string;
9     status: number;
10    creator: string;
11 }
```

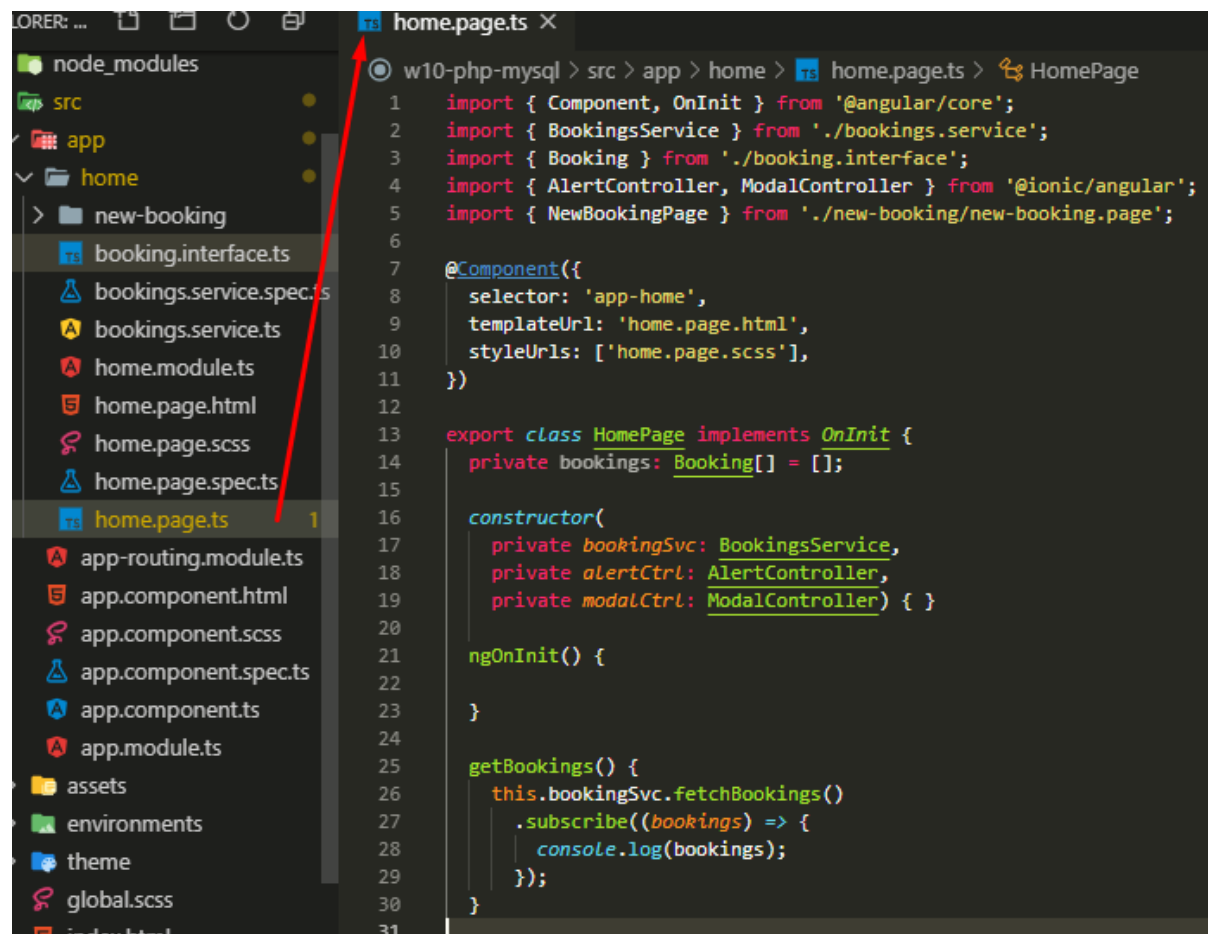
Bookings.service.ts



The screenshot shows the VS Code interface with the Explorer sidebar on the left. The file explorer shows the same project structure as the previous image. The 'bookings.service.ts' file is selected and highlighted. The main editor shows the content of 'bookings.service.ts', which implements the 'BookingService' class. The class is decorated with '@Injectable({providedIn: 'root'})'. It has a constructor that takes an 'HttpClient' as a parameter. The class implements three methods: 'fetchBookings()', 'deleteBooking(id: string)', and 'insertBooking(newBooking: any)'. The 'fetchBookings()' method returns a 'Promise<Booking[]>' by making a GET request to 'https://mobdevumn.com/api/fti\_booking\_app/get\_bookings.php'. The 'deleteBooking()' method returns a 'Promise<void>' by making a POST request to 'https://mobdevumn.com/api/fti\_booking\_app/delete\_booking.php' with the 'booking\_id' in the body. The 'insertBooking()' method returns a 'Promise<Booking>' by making a POST request to 'https://mobdevumn.com/api/fti\_booking\_app/insert\_booking.php' with the 'newBooking' object in the body. A red arrow points from the file explorer to the file name in the editor.

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient } from '@angular/common/http';
3 import { take, share, tap, switchMap } from 'rxjs/operators';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8
9 export class BookingsService {
10
11   constructor(private http: HttpClient) { }
12
13   fetchBookings() {
14     return this.http.get('https://mobdevumn.com/api/fti_booking_app/get_bookings.php');
15   }
16
17   deleteBooking(id: string) {
18     return this.http.post('https://mobdevumn.com/api/fti_booking_app/delete_booking.php', { 'booking_id': id }).pipe(take(1));
19   }
20
21   insertBooking(newBooking: any) {
22     console.log(newBooking);
23     const b = {
24       'booking_name': newBooking.booking_name,
25       'topic': newBooking.topic,
26       'details': newBooking.details,
27       'booking_date': newBooking.booking_date,
28       'start_hour': newBooking.start_hour,
29       'end_hour': newBooking.end_hour,
30       'creator': newBooking.creator,
31     };
32     return this.http.post('https://mobdevumn.com/api/fti_booking_app/insert_booking.php', b).pipe(take(1));
33   }
34 }
```

## [Home.page.ts](#)



```
1  import { Component, OnInit } from '@angular/core';
2  import { BookingsService } from '../bookings.service';
3  import { Booking } from '../booking.interface';
4  import { AlertController, ModalController } from '@ionic/angular';
5  import { NewBookingPage } from '../new-booking/new-booking.page';
6
7  @Component({
8    selector: 'app-home',
9    templateUrl: 'home.page.html',
10   styleUrls: ['home.page.scss'],
11 })
12
13 export class HomePage implements OnInit {
14   private bookings: Booking[] = [];
15
16   constructor(
17     private bookingSvc: BookingsService,
18     private alertCtrl: AlertController,
19     private modalCtrl: ModalController) { }
20
21   ngOnInit() {
22
23   }
24
25   getBookings() {
26     this.bookingSvc.fetchBookings()
27       .subscribe((bookings) => {
28         console.log(bookings);
29       });
30   }
31 }
```

EXPLORER: ...

node\_modules

src

app

home

new-booking

booking.interface.ts

bookings.service.spec.ts

bookings.service.ts

home.module.ts

home.page.html

home.page.scss

home.page.spec.ts

home.page.ts

app-routing.module.ts

app.component.html

app.component.scss

app.component.spec.ts

app.component.ts

app.module.ts

assets

environments

theme

global.scss

index.html

main.ts

polyfills.ts

test.ts

zone-flags.ts

home.page.ts

w10-php-mysql > src > app > home > home.page.ts > HomePage

```
31
32
33 async presentAlertPrompt() {
34   const alert = await this.alertCtrl.create({
35     header: 'Delete a Booking',
36     inputs: [
37       {
38         name: 'bookingId',
39         type: 'text',
40         placeholder: 'Enter your booking ID'
41       },
42     ],
43     buttons: [
44       {
45         text: 'Cancel',
46         role: 'cancel',
47         cssClass: 'secondary',
48         handler: () => {
49           console.log('Confirm Cancel');
50         }
51       }, {
52         text: 'Ok',
53         handler: (data) => {
54           this.bookingSvc.deleteBooking(data.bookingId)
55             .subscribe(() => {
56               this.bookingSvc.fetchBookings()
57                 .subscribe((bookings) => {
58                   console.log(bookings);
59                 });
60               console.log("DELETED");
61             });
62         }
63       }
64     ]
65   });
66   await alert.present();
67 }
```

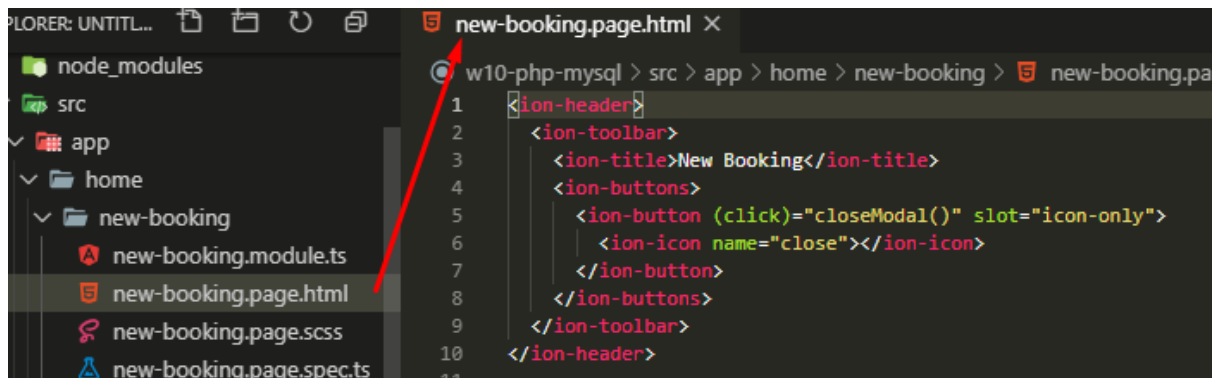
EXPLORER: ...

home.page.ts

w10-php-mysql > src > app > home > home.page.ts >

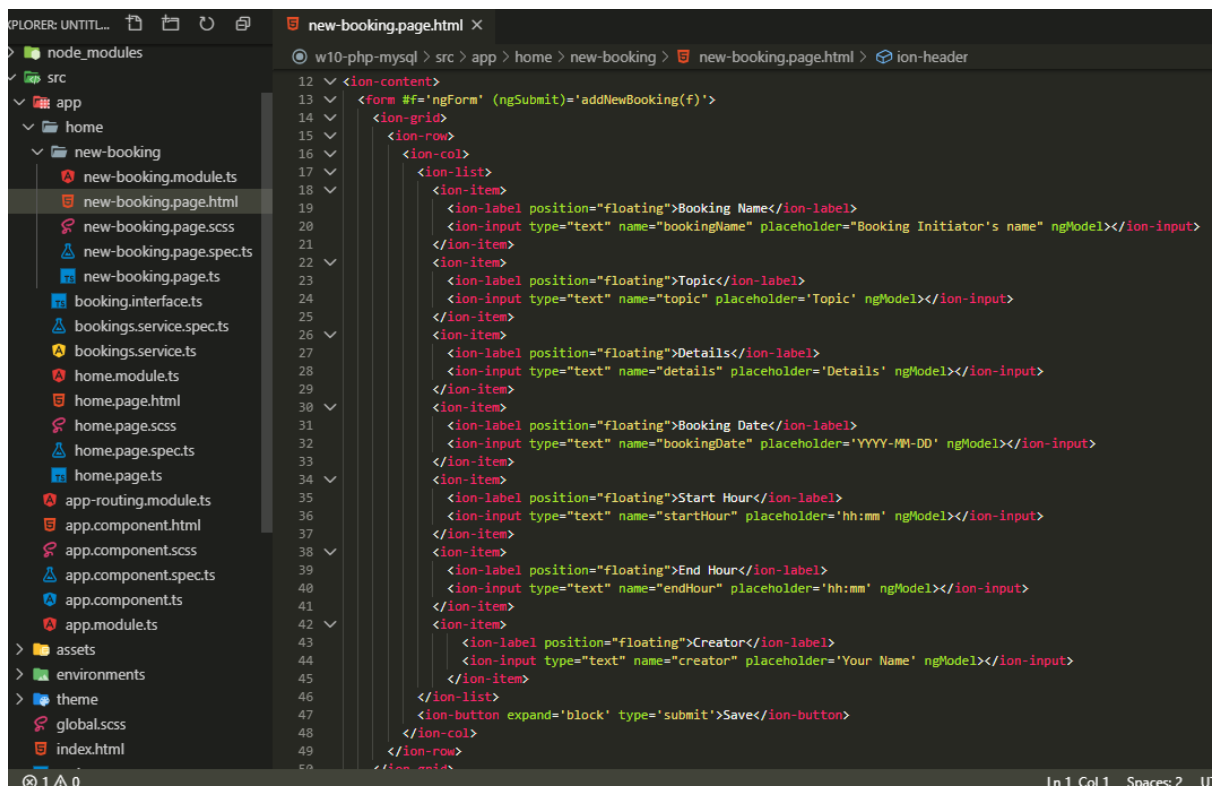
```
68
69
70 async presentModal() {
71   const modal = await this.modalCtrl.create({
72     component: NewBookingPage
73   });
74   return await modal.present();
75 }
76
77 newBooking() {
78   this.presentModal();
79 }
80
81 deleteBooking() {
82   this.presentAlertPrompt();
83 }
84 }
```

## New booking page



The screenshot shows the VS Code interface. On the left, the file explorer displays the project structure: `node_modules`, `src`, `app`, `home`, and `new-booking`. The `new-booking` folder is expanded, showing `new-booking.module.ts`, `new-booking.page.html` (selected with a red arrow), `new-booking.page.scss`, and `new-booking.page.spec.ts`. The main editor shows the content of `new-booking.page.html`, which is an Ionic header component.

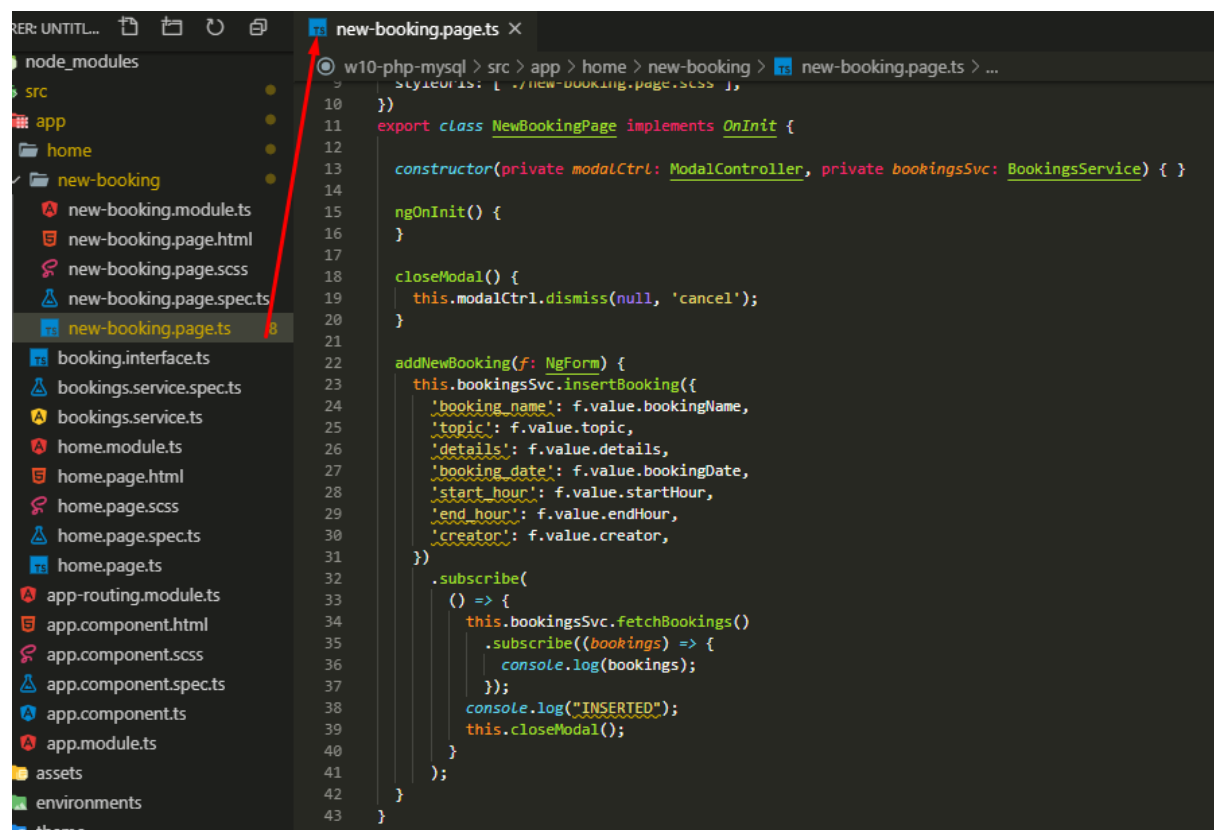
```
1 <ion-header>
2   <ion-toolbar>
3     <ion-title>New Booking</ion-title>
4     <ion-buttons>
5       <ion-button (click)="closeModal()" slot="icon-only">
6         <ion-icon name="close"></ion-icon>
7       </ion-button>
8     </ion-buttons>
9   </ion-toolbar>
10 </ion-header>
```



The screenshot shows the VS Code interface with the `new_booking.page.html` file open. The file explorer on the left shows the project structure, with `new_booking.page.html` selected. The main editor shows the content of the file, which is an Ionic page component. The `<ion-content>` section is expanded, showing a form with various input fields and a submit button.

```
12 <ion-content>
13   <form #f='ngForm' (ngSubmit)="addNewBooking(f)">
14     <ion-grid>
15       <ion-row>
16         <ion-col>
17           <ion-list>
18             <ion-item>
19               <ion-label position="floating">Booking Name</ion-label>
20               <ion-input type="text" name="bookingName" placeholder="Booking Initiator's name" ngModel></ion-input>
21             </ion-item>
22             <ion-item>
23               <ion-label position="floating">Topic</ion-label>
24               <ion-input type="text" name="topic" placeholder="Topic" ngModel></ion-input>
25             </ion-item>
26             <ion-item>
27               <ion-label position="floating">Details</ion-label>
28               <ion-input type="text" name="details" placeholder="Details" ngModel></ion-input>
29             </ion-item>
30             <ion-item>
31               <ion-label position="floating">Booking Date</ion-label>
32               <ion-input type="text" name="bookingDate" placeholder="YYYY-MM-DD" ngModel></ion-input>
33             </ion-item>
34             <ion-item>
35               <ion-label position="floating">Start Hour</ion-label>
36               <ion-input type="text" name="startHour" placeholder="hh:mm" ngModel></ion-input>
37             </ion-item>
38             <ion-item>
39               <ion-label position="floating">End Hour</ion-label>
40               <ion-input type="text" name="endHour" placeholder="hh:mm" ngModel></ion-input>
41             </ion-item>
42             <ion-item>
43               <ion-label position="floating">Creator</ion-label>
44               <ion-input type="text" name="creator" placeholder="Your Name" ngModel></ion-input>
45             </ion-item>
46           </ion-list>
47           <ion-button expand="block" type="submit">Save</ion-button>
48         </ion-col>
49       </ion-row>
50     </ion-grid>
```

## New-booking.page.ts



```
10  })
11  export class NewBookingPage implements OnInit {
12
13
14      constructor(private modalCtrl: ModalController, private bookingsSvc: BookingsService) { }
15
16      ngOnInit() {
17      }
18
19      closeModal() {
20          this.modalCtrl.dismiss(null, 'cancel');
21      }
22
23      addNewBooking(f: NgForm) {
24          this.bookingsSvc.insertBooking({
25              'booking_name': f.value.bookingName,
26              'topic': f.value.topic,
27              'details': f.value.details,
28              'booking_date': f.value.bookingDate,
29              'start_hour': f.value.startHour,
30              'end_hour': f.value.endHour,
31              'creator': f.value.creator,
32          })
33          .subscribe(
34              () => {
35                  this.bookingsSvc.fetchBookings()
36                      .subscribe((bookings) => {
37                          console.log(bookings);
38                      });
39                  console.log("INSERTED");
40                  this.closeModal();
41              }
42          );
43      }
```