Firebase CRUD in Angular

- Firebase Database
- Angularfire
- Firebase CRUD in Angular

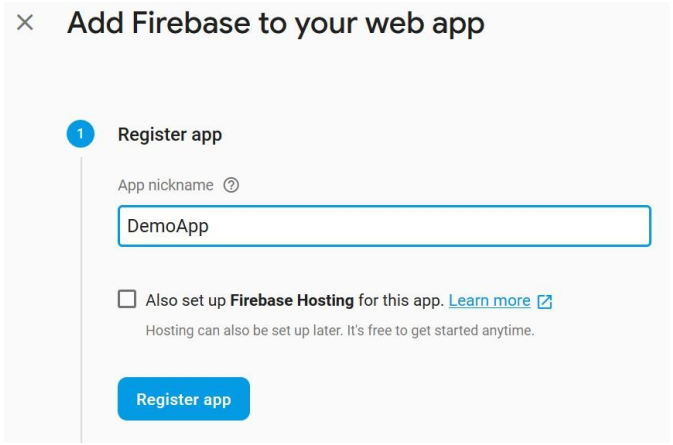
Firebase

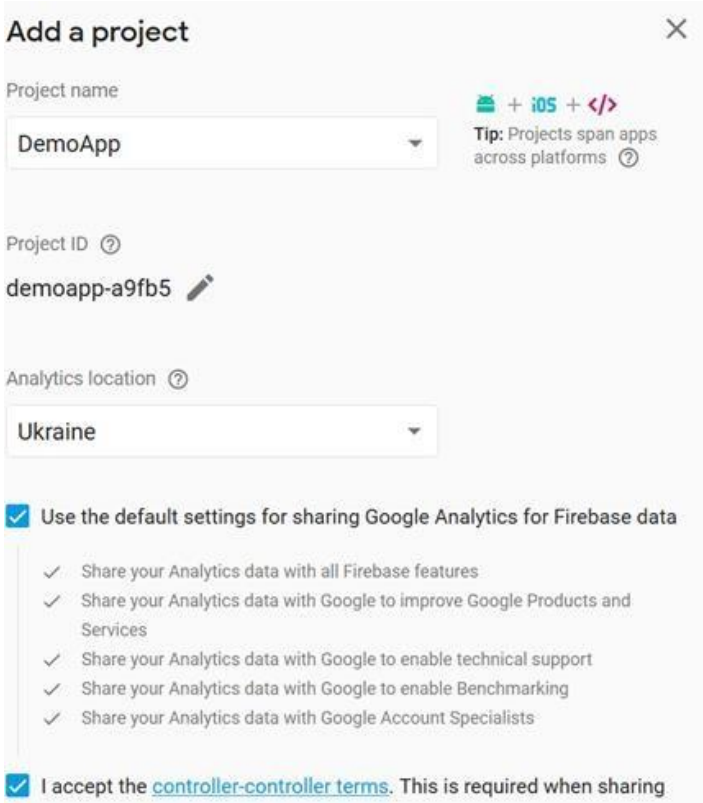
Firebase is a mobile and web application development platform

firebase.google.com

Firebase Database Example

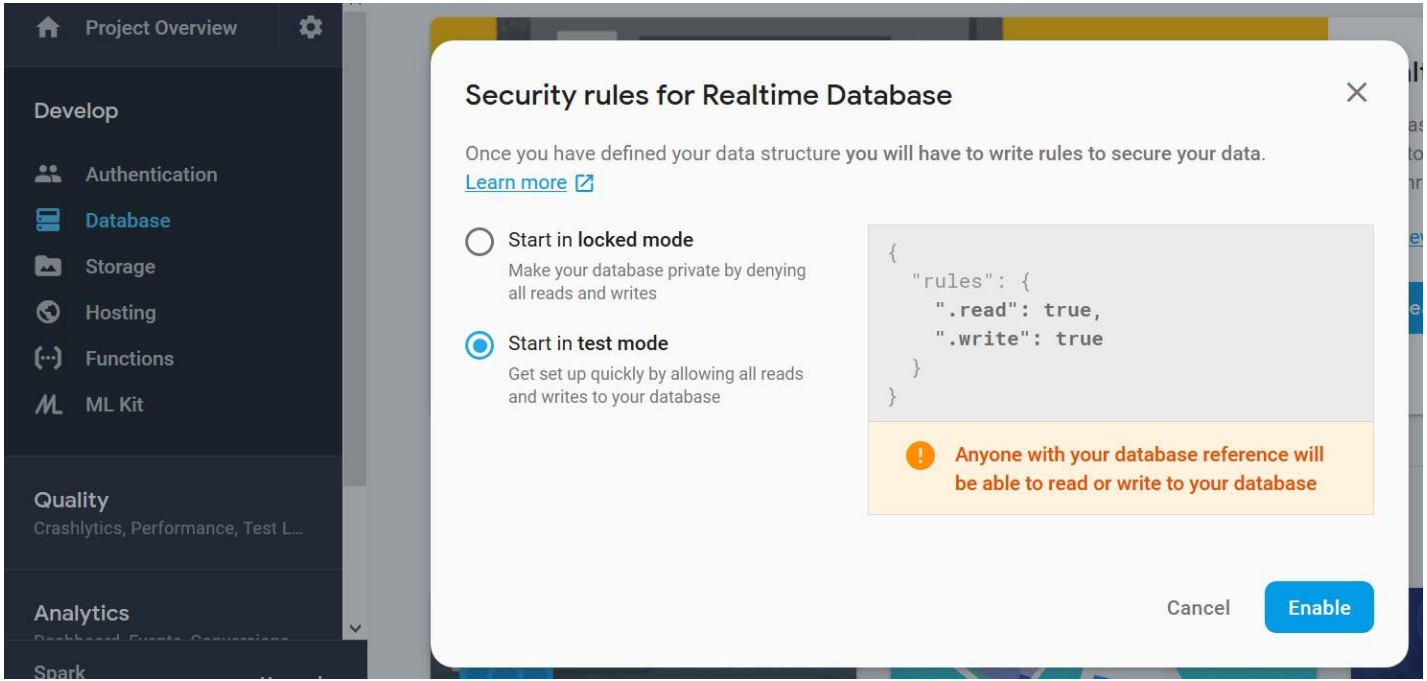
Firebase Database





Realtime database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client.



Firebase configuration

```
var firebaseConfig = {
  apiKey: "AIzaSyDePEB1ED8H7IpuGVzSa3xYiWRCxfrJw",
  authDomain: "demoapp-a9fb5.firebaseapp.com",
  databaseURL: "https://demoapp-a95.firebaseio.com",
  projectId: "demoapp-a9f5",
  storageBucket: "",
  messagingSenderId: "77244142497",
  appId: "1:772484142497:web:9a2c8dedf0ea54e"
};
```

Angular Part

Angular Part - Create a new Project

ng new FireApp

What is AngularFire?

The official library for Firebase and Angular

- **Observable based** - Use the power of RxJS, Angular, and Firebase.
- **Realtime bindings** - Synchronize data in realtime.
- **Authentication** - Log users in with a variety of providers and monitor authentication state.
- **Offline Data** - Store data offline automatically with AngularFirestore.
- **Server-side Render** - Generate static HTML to boost perceived performance or create static sites.
- **ngrx friendly** - Integrate with ngrx using AngularFire's action based APIs.
- **Manage binary data** - Upload, download, and delete binary files like images, videos, and other blobs.
- **Call server code** - Directly call serverless Cloud Functions with user context automatically passed.
- **Push notifications** - Register and listen for push notifications
- **Modular** - Include only what's needed.

Add Firebase to your project - @angular/fire

- npm install firebase @angular/fire --save
- npm i rxjs --save

- app.modules.ts
- import {AngularFireModule} from '@angular/fire';
- import {AngularFireDatabaseModule} from '@angular/fire/database';

app.module.ts – Part 1

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { FormsModule } from "@angular/forms";
import { HttpClientModule } from "@angular/common/http";
import {AngularFireModule} from '@angular/fire';
import {AngularFireDatabaseModule} from '@angular/fire/database';


export const firebaseConfig = {
};

@NgModule({
declarations: [
AppComponent
],
```

app.module.ts – Part 2

```
imports: [
    BrowserModule,
    FormsModule,
    HttpClientModule,
    AngularFireModule.initializeApp(firebaseConfig),
    AngularFireDatabaseModule,
AppRoutingModule
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }
```

app.component.css – Test visualization

- .box{
- width:
200px;
height: 200px;
- border: 1px solid black;
background: yellowgreen;
margin: 2px;
- float:left;
- }

- p{
- border: 1px dotted
black; margin: 2px;
- }

```
.control_panel, .container{
width: 800px;
float: bottom;
}
.submit{
border: 1px dotted grey;
margin: 2px;
padding: 2px;
display: inline-block;
}

div{
float: left;
}
```

app.component.ts – Properties

```
import { Component } from '@angular/core';
import { AngularFireDatabase, AngularFireList } from '@angular/fire/database';
import { map } from 'rxjs/operators';

@Component({
selector: 'app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
export class AppComponent {

todos: AngularFireList<any>;
keysTodos = [];
countTodos: number = 0;
todo: any;
```

app.component.ts – ngOnInit

```
ngOnInit() {

this.todos = this.db.list('/todos', ref =>
ref.limitToFirst(11));

this.todos.snapshotChanges().subscribe(tmp => {
this.keysTodos = tmp;
this.countTodos = tmp.length;

})
}
```

app.component.ts – getToDoById

```
getToDoById(todoId: string) {

let tmp = "/todos/" + todoId; //'-LhbUBQeoTRt1lgHKwDq'

this.todo = this.db.list(tmp)
.snapshotChanges()
.pipe(map(items => {

return items.map(a => {
const data = a.payload.val();
const key = a.payload.key;

console.log({ key, data });

return { key, data }; j
});
}));
}
```

app.component.ts – addTodo/deleteTodo

```
addTodo(value: string): void {
this.todos.push({ content: value, done: false});
}

deleteTodo(id: string): void {
let tmp = "/todos/" + id;
this.db.object(tmp).remove();
}
}
```

app.component.html – Toolbar

```
<div class="control_panel">
<input type="text" name="todo" id="todo" placeholder="Your ToDo"
#todo>

<input type="submit" value="Get Todo By Id"
class="submit"  (click)="getToDoById(todo.value);">

<input type="submit" value="Delete Todo" class="submit"
(click)="deleteTodo(todo.value);">

<input type="submit" value="Add Todo"
class="submit"  (click)="addTodo(todo.value);">
</div>
```

app.component.html – keysTodos/Count

```
<div class="control_panel">
<div *ngFor='let item of keysTodos; let i = index' class="container">
<div class="box">
      <p>Number:{{i}}</p>
      <p>Key: {{item.key}}</p>
      <p>Desc: {{item.payload.val().content}} </p>
      <p>Status: {{item.payload.val().done}} </p>
</div>
</div>
</div>


<div class="control_panel">Count: {{countTodos}}</div>
```