

# Configuración del Token para hacer Login



## Configuración del Token para hacer Login

En esta lección vamos a obtener el token para poder hacer login desde la aplicación.

En primer lugar, obtenemos los datos de configuración de la aplicación de firebase (deben ser los de su propio proyecto):

The screenshot shows the Firebase console interface for a project named "tienda-online". A yellow box labeled "1" highlights the "Descripción general" (General description) button. A red box labeled "2" highlights the "Configuración del proyecto" (Project configuration) button. The "Authentication" section is currently selected. A warning message at the top right states: "El acceso con redirecciónamiento de origen cruzado dejará de funcionar el 24 de junio de 2024." (Cross-origin redirection access will stop working on June 24, 2024.)

This screenshot shows the same project configuration page as above. A red arrow points to the "Tus apps" (Your apps) section, which displays a message: "No hay apps en tu proyecto" (There are no apps in your project). Below this, there are five circular icons representing different platforms: iOS, Android, Web (highlighted with a yellow box and red arrow), macOS, and Linux. At the bottom right of this section is a "Borrar proyecto" (Delete project) button.



Estos son los datos que vamos a utilizar posteriormente:

Si ya usas npm y un agrupador de módulos como Webpack o Rollup, puedes ejecutar el siguiente comando para instalar la versión más reciente del SDK (más información):

```
$ npm install firebase
```

Luego, inicializa Firebase y comienza a usar los SDK de los productos que quieras utilizar.

```
// Import the functions you need from the SDKs you need
import { initializeApp } from "firebase/app";
// TODO: Add SDKs for Firebase products that you want to use
// https://firebase.google.com/docs/web/setup#available-libraries

// Your web app's Firebase configuration
const firebaseConfig = {
  apiKey: "AIzaSyAkvTWeTofqdV8mIEXjHn6ukHuDTQJi5M",
  authDomain: "tienda-online-7f4e5.firebaseioapp.com",
  databaseURL: "https://tienda-online-7f4e5-default-rtdb.firebaseio.com",
  projectId: "tienda-online-7f4e5",
  storageBucket: "tienda-online-7f4e5.appspot.com",
  messagingSenderId: "955021264959",
  appId: "1:955021264959:web:65a55e886bc0141a25508a"
};

// Initialize Firebase
const app = initializeApp(firebaseConfig);
```

Instalamos el módulo de firebase en nuestra aplicación:

```
npm install firebase --save
```

Posteriormente creamos un servicio para configurar el acceso a firebase y el manejo de login:

```
ng g s firebase --skip-tests
```

También vamos a crear un servicio para el manejo de login:

```
ng g s login --skip-tests
```

Modificamos ahora los archivos.

Código firebase.service.ts:

```
import { Injectable } from '@angular/core';
// Import the functions you need from the SDKs you need
import { initializeApp } from 'firebase/app';
import { Auth, getAuth } from 'firebase/auth';
import { Firestore, getFirestore } from 'firebase/firestore';

@Injectable({
  providedIn: 'root',
})
export class FirebaseService {

  firebaseConfig = {
    apiKey: 'AIzaSyAkaVTWeTofqdV0mIEXjHn6ukHuDTQJi5M',
    authDomain: 'tienda-online-7f4e5.firebaseio.com',
    databaseURL: 'https://tienda-online-7f4e5-default-rtdb.firebaseio.com',
    projectId: 'tienda-online-7f4e5',
    storageBucket: 'tienda-online-7f4e5.appspot.com',
    messagingSenderId: '955021264959',
    appId: '1:955021264959:web:65a55e886bc0141a25508a',
  };

  public auth: Auth;
  public firebase: Firestore;

  constructor() {
    const app = initializeApp(this.firebaseioConfig);
    this.auth = getAuth(app);
    this.firebaseio = getFirestore(app);
  }
}
```

Código login.service.ts:

```
import { Injectable } from '@angular/core';
```

```
import { Router } from '@angular/router';
import { FirebaseService } from './firebase.service';
import { signInWithEmailAndPassword } from 'firebase/auth';

@Injectable({
  providedIn: 'root'
})
export class LoginService {

  token: string = '';

  constructor(
    private router: Router,
    private firebaseService: FirebaseService) {
  }

  login(email: string, password: string) {
    const auth = this.firebaseio.auth;

    signInWithEmailAndPassword(auth, email, password)
      .then(() => {
        auth.currentUser?.getIdToken().then((token) => {
          this.token = token;
          this.router.navigate(['/']);
        });
      })
      .catch((error) => {
        console.error('Error al iniciar sesión:', error);
      });
  }

  getIdToken() {
    return this.token;
  }
}
```

Código login.component.ts:

```
import { Component } from '@angular/core';
import { FormsModule, NgForm } from '@angular/forms';
import { LoginService } from '../login.service';

@Component({
  selector: 'app-login',
```

```

standalone: true,
imports: [FormsModule],
templateUrl: './login.component.html',
styleUrl: './login.component.css',
})
export class LoginComponent {

constructor(private loginService: LoginService){}

login(form: NgForm){
  const email = form.value.email;
  const password = form.value.password;
  this.loginService.login(email, password);
}
}

```

Modificamos el servicio de datos para enviar el token una vez que hemos hecho login correctamente.

Código datos.service.ts:

```

import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { Producto } from './producto/producto.model';
import { Observable } from 'rxjs';
import { LoginService } from './login.service';

@Injectable({
  providedIn: 'root',
})
export class DataService {
  url = 'https://tienda-online-7f4e5-default-rtdb.firebaseio.com/';

  constructor(
    private httpClient: HttpClient,
    private loginService: LoginService
  ) {}

  cargarProductos(): Observable<{ [llave: string]: Producto }> {
    const token = this.loginService.getIdToken();
    return this.httpClient.get<{ [llave: string]: Producto }>(this.url +
    'datos.json?auth=' + token);
  }

  guardarProducto(producto: Producto): Observable<any> {

```

```
        return this.httpClient.post(`.${this.url}datos.json`, producto);
    }

modificarProducto(producto: Producto, llave: string): Observable<any> {
    const url_modificar = `.${this.url}datos/${llave}.json`;
    return this.httpClient.put(url_modificar, producto);
}

eliminarProducto(llave: string): Observable<any> {
    const url_eliminar = `.${this.url}datos/${llave}.json`;
    return this.httpClient.delete(url_eliminar);
}
}
```

Entramos a la página de login:

<http://localhost:4200/login>



Nos debe redirigir al listado de productos y visualizarlo correctamente.



Con esto ya hemos configurado el login y recuperado el token que debemos utilizar para validar las peticiones http a la base de datos de firebase. En la siguiente lección terminaremos de configurar las acciones de agregar, modificar o eliminar productos.

Saludos!

Ing. Ubaldo Acosta

Fundador de [GlobalMentoring.com.mx](http://www.globalmentoring.com.mx)