

Etape 1:

The screenshot displays the Firebase console interface. On the left is a dark sidebar with the 'Firebase' logo and a 'Project Overview' section containing links for DEVELOP, STABILITY, ANALYTICS, and GROW. The main content area has a blue background with the text 'Welcome to Firebase! Get started here.' and three circular icons for iOS, Android, and web. Below these are instructions to 'Add Firebase to your [platform] app'. At the bottom, there's a 'Discover Firebase' section with three colorful cards.

Firebase

Test2 ▾

Go to docs 🔔 👤

Project Overview ⚙️

DEVELOP
Authentication, Database, Storage, H...

STABILITY
Crashlytics, Crash Reporting, Perfor...

ANALYTICS
Dashboard, Events, Audiences, Attrib...

GROW
Predictions, Notifications, Remote C...

Spark
Free \$0/month **UPGRADE**

Welcome to Firebase!

Get started here.

iOS
Add Firebase to your iOS app


Android
Add Firebase to your Android app

</>
Add Firebase to your web app

Discover Firebase



```
<script src="https://www.gstatic.com/firebasejs/4.9.0/firebase.js"></script>
<script>
  // Initialize Firebase
  // TODO: Replace with your project's customized code snippet
  var config = {
    apiKey: "<API_KEY>",
    authDomain: "<PROJECT_ID>.firebaseapp.com",
    databaseURL: "https://<DATABASE_NAME>.firebaseio.com",
    storageBucket: "<BUCKET>.appspot.com",
    messagingSenderId: "<SENDER_ID>",
  };
  firebase.initializeApp(config);
</script>
```



```
export const environment = {  
  firebase: {  
    apiKey: "YOUR_API_KEY",  
    authDomain: "YOUR_AUTH_DOMAIN",  
    databaseURL: "YOUR_DATABASE_URL",  
    projectId: "YOUR_PROJECT_ID",  
    storageBucket: "YOUR_STORAGE_BUCKET",  
    messagingSenderId: "YOUR_SENDER_ID"  
  }  
};
```



Etape 2 : Installation du AngularFire2

```
npm install firebase angularfire2 --save
```

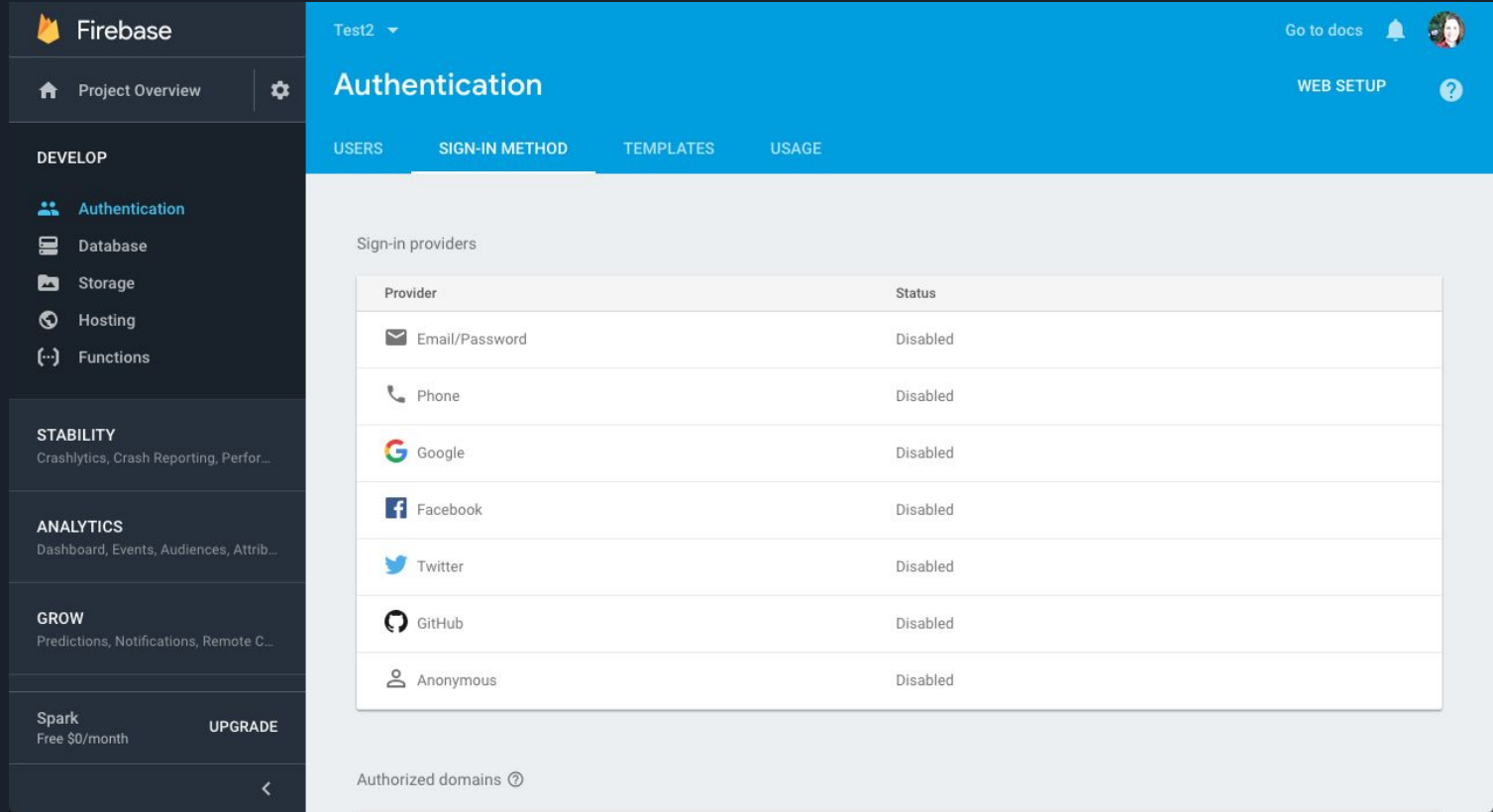


Etape 3 : Update AppModule

```
import { AngularFireModule } from 'angularfire2';
import { AngularFirestoreModule } from 'angularfire2/firestore';
import { AngularFireAuthModule } from 'angularfire2/auth';
import { environment } from '../environments/environment';

imports: [
  AngularFireModule.initializeApp(environment.firebase),
  AngularFirestoreModule, // imports firebase/firestore, only needed for database
  features
  AngularFireAuthModule, // imports firebase/auth, only needed for auth features
]
```

Etape 4 : Configurer les méthodes de connexion Firebase



The screenshot displays the Firebase Authentication console interface. On the left is a dark sidebar with navigation options: Project Overview, DEVELOP (Authentication, Database, Storage, Hosting, Functions), STABILITY, ANALYTICS, GROW, and Spark. The main content area has a blue header with 'Test2' and 'Authentication' tabs. Below the header are sub-tabs: USERS, SIGN-IN METHOD (selected), TEMPLATES, and USAGE. The 'Sign-in providers' section contains a table listing various providers and their status.

Provider	Status
Email/Password	Disabled
Phone	Disabled
Google	Disabled
Facebook	Disabled
Twitter	Disabled
GitHub	Disabled
Anonymous	Disabled

Authorized domains ⓘ



Etape 5 : Préparation du service

```
import { AngularFireAuth } from 'angularfire2/auth';  
import * as firebase from 'firebase/app';
```

```
.
```

```
.
```

```
constructor(public afAuth: AngularFireAuth ) {  
  
}
```



A. Authentification avec email & password

1. Register Component : Création du Form Group :

```
this.registerForm = new FormGroup({  
    'email' : new FormControl(),  
    'password' : new FormControl()  
})
```




A. Authentification avec email & password

2. Register Template : Formulaire en HTML :

```
<form [formGroup]="registerForm">
  <div class="form-group">
    <label>Email address</label>
    <input type="email" formControlName="email" class="form-control" required>
  </div>
  <div class="form-group">
    <label>Password</label>
    <input type="password" class="form-control" formControlName="password" required>
    <label class="error">{{errorMessage}}</label>
    <label class="success">{{successMessage}}</label>
  </div>
  <button type="submit" (click)="registerAction(registerForm.value)" class="btn
btn-default">Register</button>
</form>
```



A. Authentification avec email & password

3. Register Component : RegisterAction :

```
registerAction(value) {  
  this.authService.doRegister(value)  
    .then(res => {  
    console.log(res);  
    this.errorMessage = "";  
    this.successMessage = "Your account has been created";  
  }, err => {  
    console.log(err);  
    this.errorMessage = err.message;  
    this.successMessage = "";  
  })  
}
```



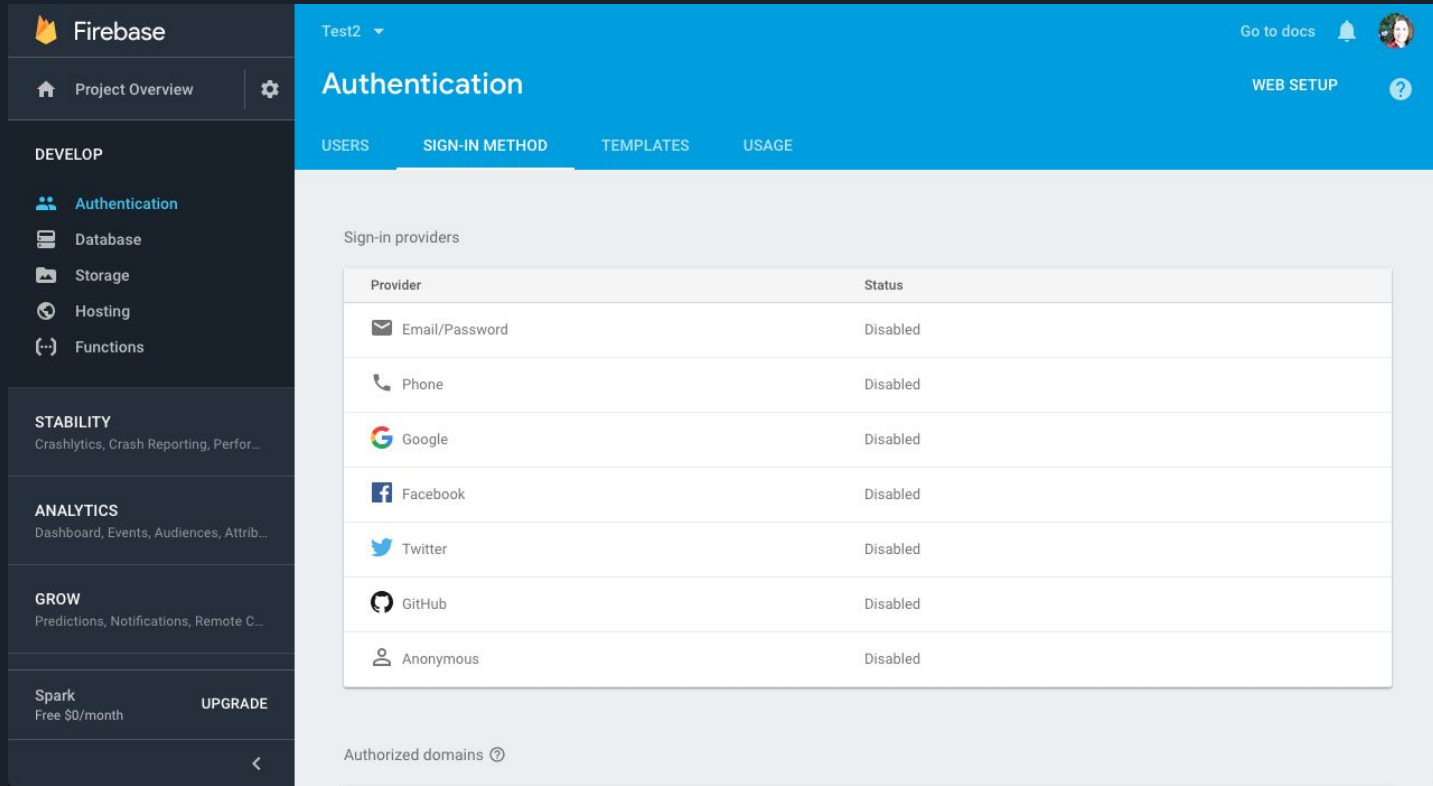
A. Authentification avec email & password

4. AuthService

```
doRegister(value) {  
  return new Promise<any>((resolve, reject) => {  
    firebase.auth().createUserWithEmailAndPassword(value.email,  
value.password)  
      .then(res => {  
        resolve(res);  
      }, err => reject(err))  
  })  
}
```

A. Authentification avec email & password

5. Activer La méthode de connexion avec email & password



The screenshot displays the Firebase Authentication console. The left sidebar contains the following sections:

- DEVELOP**
 - Authentication (selected)
 - Database
 - Storage
 - Hosting
 - Functions
- STABILITY**
Crashlytics, Crash Reporting, Perfor...
- ANALYTICS**
Dashboard, Events, Audiences, Attrib...
- GROW**
Predictions, Notifications, Remote C...
- Spark**
Free \$0/month **UPGRADE**

The main content area is titled 'Authentication' and includes tabs for 'USERS', 'SIGN-IN METHOD' (selected), 'TEMPLATES', and 'USAGE'. The 'Sign-in providers' section shows a table of providers:

Provider	Status
Email/Password	Disabled
Phone	Disabled
Google	Disabled
Facebook	Disabled
Twitter	Disabled
GitHub	Disabled
Anonymous	Disabled

At the bottom, there is a link for 'Authorized domains' with a help icon.



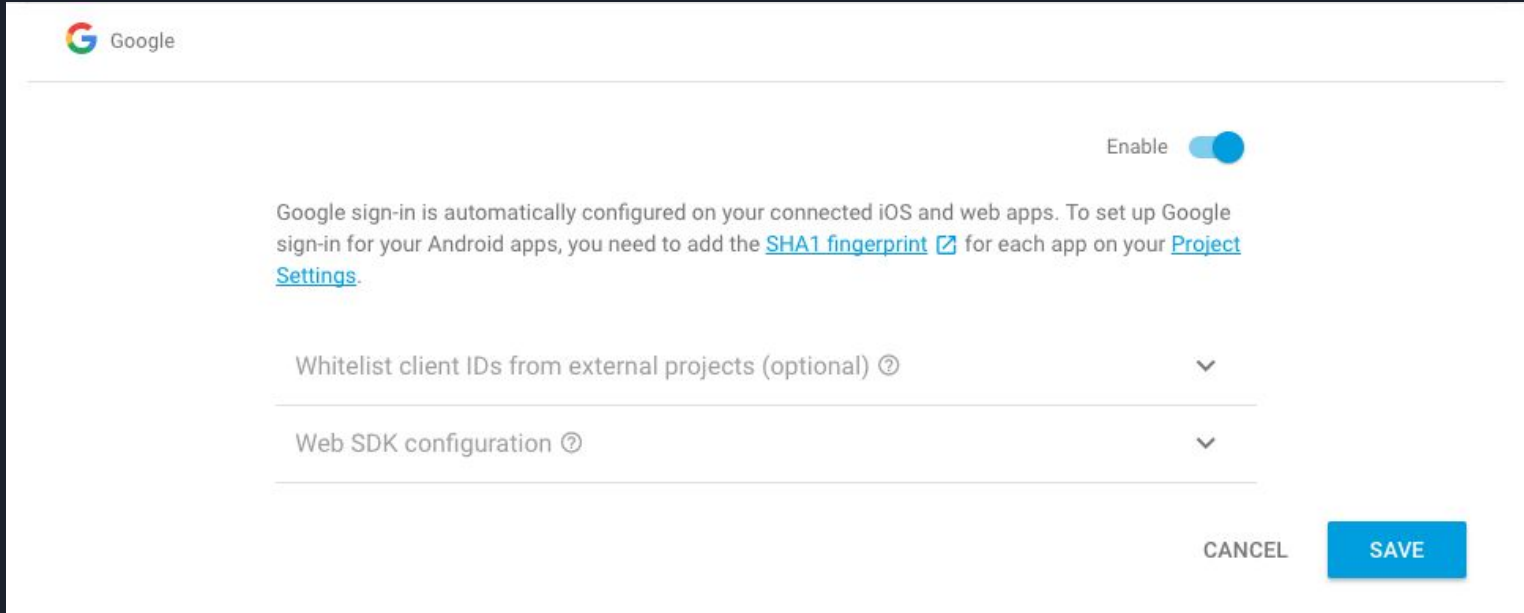
A. Authentification avec email & password

5. Login : Même que l'inscription

```
firebase.auth().signInWithEmailAndPassword(value.email, value.password)
```

B. Authentification avec Google Account

1. Setup fireBase



The screenshot shows the 'Google sign-in' configuration page in the Firebase console. At the top left is the Google logo. On the right, there is an 'Enable' toggle switch which is turned on. Below this, a paragraph explains that Google sign-in is automatically configured for iOS and web apps, and that for Android apps, the user needs to add the SHA1 fingerprint for each app on their Project Settings. There are two expandable sections: 'Whitelist client IDs from external projects (optional)' and 'Web SDK configuration', both with a downward arrow icon. At the bottom right, there are 'CANCEL' and 'SAVE' buttons.

Google

Enable ☒

Google sign-in is automatically configured on your connected iOS and web apps. To set up Google sign-in for your Android apps, you need to add the [SHA1 fingerprint](#) for each app on your [Project Settings](#).

Whitelist client IDs from external projects (optional) ?

Web SDK configuration ?

CANCEL SAVE

B. Authentification avec Google Account

2. Auth Service

```
doGoogleLogin() {  
  
  return new Promise<any>((resolve, reject) => {  
    let provider = new firebase.auth.GoogleAuthProvider();  
    provider.addScope('profile');  
    provider.addScope('email');  
    this.afAuth.auth  
      .signInWithPopup(provider)  
      .then(res => {  
        resolve(res);  
        console.log(res.additionalUserInfo.profile);  
        console.log(res.additionalUserInfo.profile);  
      })  
  })  
}
```