

Firestore

Firestore è una piattaforma di sviluppo di applicazioni mobile e web sviluppata da Google. Offre una vasta gamma di servizi e strumenti per aiutare gli sviluppatori a creare, migliorare e scalare le loro applicazioni più rapidamente ed efficacemente. Ecco una panoramica dei principali servizi e funzionalità offerti da Firestore:

1. Realtime Database:

- Firestore Realtime Database è un database NoSQL in tempo reale che consente agli sviluppatori di sincronizzare i dati tra gli utenti in tempo reale.
- I dati vengono memorizzati in formato JSON e sincronizzati automaticamente con ogni client connesso in tempo reale.
- È ideale per applicazioni che richiedono aggiornamenti istantanei dei dati, come chat in tempo reale, giochi multiplayer e app collaborative.

2. Authentication:

- Firestore Authentication offre un sistema di autenticazione completo e scalabile che consente agli sviluppatori di gestire l'autenticazione degli utenti in modo sicuro e semplice.
- Supporta l'autenticazione tramite email/password, provider di social media come Google, Facebook, Twitter, GitHub, e altro ancora, così come l'autenticazione tramite telefono.
- Fornisce funzionalità avanzate come la gestione delle sessioni, il ripristino delle password, la verifica dell'email e molto altro.

3. Cloud Firestore:

- Firestore è un database NoSQL documentale scalabile che offre una maggiore flessibilità e scalabilità rispetto al Realtime Database.
- Supporta query complesse, indicizzazione avanzata, transazioni e sincronizzazione in tempo reale dei dati.

- È ideale per applicazioni con grandi volumi di dati e requisiti di query complessi, come applicazioni di e-commerce, social network e strumenti di produttività.

4. Storage:

- Firebase Storage offre un servizio di hosting di file completamente gestito che consente agli sviluppatori di caricare e scaricare file come immagini, video, audio e altri contenuti multimediali.
- Supporta la memorizzazione sicura e scalabile dei file, così come l'accesso diretto ai file tramite URL generati automaticamente.

5. Cloud Functions:

- Firebase Cloud Functions consente agli sviluppatori di scrivere e distribuire codice serverless che viene eseguito in risposta agli eventi generati dai servizi Firebase e da altri servizi cloud.
- Le funzioni possono essere scritte in JavaScript, TypeScript o altri linguaggi supportati e possono essere utilizzate per eseguire compiti come l'elaborazione dei dati, l'invio di notifiche push e l'interazione con servizi esterni.

6. Hosting:

- Firebase Hosting è un servizio di hosting Web che consente agli sviluppatori di distribuire facilmente e velocemente le loro applicazioni web e i contenuti statici.
- Supporta l'hosting di singole pagine, applicazioni a pagina singola (SPA) e applicazioni web progressive (PWA), e offre un'integrazione diretta con altri servizi Firebase come Authentication e Firestore.

7. Analytics, Performance Monitoring, Crashlytics:

- Firebase offre anche una serie di strumenti per monitorare le prestazioni delle applicazioni, analizzare il comportamento degli utenti, tracciare gli errori e risolvere i problemi.

- Questi strumenti includono Firebase Analytics per analizzare il comportamento degli utenti, Firebase Performance Monitoring per monitorare le prestazioni delle applicazioni e Firebase Crashlytics per tracciare e risolvere i crash dell'applicazione.

Come utilizzare l'autenticazione con Firebase

Inizializzare un'applicazione Firebase:

- Per iniziare, devi creare un progetto Firebase sulla console Firebase (<https://console.firebase.google.com/>) e seguire le istruzioni per aggiungere Firebase al tuo progetto.
- Quindi, includi il SDK Firebase nel tuo progetto, il che ti consentirà di utilizzare le funzionalità di Firebase come l'autenticazione.
- Puoi inizializzare Firebase nel tuo progetto utilizzando il codice seguente:

```
import firebase from 'firebase/app';
import 'firebase/auth';

const firebaseConfig = {
  apiKey: "YOUR_API_KEY",
  authDomain: "YOUR_AUTH_DOMAIN",
  projectId: "YOUR_PROJECT_ID",
  storageBucket: "YOUR_STORAGE_BUCKET",
  messagingSenderId: "YOUR_MESSAGING_SENDER_ID",
  appId: "YOUR_APP_ID"
};

firebase.initializeApp(firebaseConfig);
```

Utilizzare i metodi di autenticazione:

- Firebase supporta diversi metodi di autenticazione, tra cui email/password, OAuth (Google, Facebook, Twitter, ecc.), telefono, e altro ancora.
- Puoi utilizzare questi metodi di autenticazione chiamando le relative funzioni nell'oggetto `firebase.auth()`.

Gestire lo stato di autenticazione degli utenti:

- Puoi monitorare lo stato di autenticazione degli utenti registrando un listener per gli eventi di autenticazione con `onAuthStateChanged()`.
- Questo listener verrà chiamato ogni volta che lo stato di autenticazione dell'utente cambia (ad esempio, quando un utente effettua il login o il logout).

Sign In con email e password:

- Per consentire agli utenti di effettuare il login utilizzando l'email e la password, puoi utilizzare la funzione `signInWithEmailAndPassword()`.

```
firebase.auth().signInWithEmailAndPassword(email, password)
  .then((userCredential) => {
    // Login success
    const user = userCredential.user;
  })
  .catch((error) => {
    // Handle login error
  });
```

Sign Out:

- Per disconnettere un utente, puoi utilizzare la funzione `signOut()`.

```
firebase.auth().signOut().then(() => {
  // Sign-out successful.
```

```
}).catch((error) => {  
  // An error happened.  
});
```

Sign Up:

- Per consentire agli utenti di registrarsi, puoi utilizzare la funzione

`createUserWithEmailAndPassword()`.

```
firebase.auth().createUserWithEmailAndPassword(email, password)  
  .then((userCredential) => {  
    // Sign up success  
    const user = userCredential.user;  
  })  
  .catch((error) => {  
    // Handle sign up error  
  });
```

Mandare una mail di verifica:

- Dopo la registrazione, puoi inviare un'email di verifica utilizzando la funzione

`sendEmailVerification()`.

```
const user = firebase.auth().currentUser;  
  
user.sendEmailVerification().then(() => {  
  // Email sent  
}).catch((error) => {  
  // An error happened  
});
```

Aggiornare la password:

- Gli utenti possono aggiornare la propria password utilizzando la funzione `updatePassword()`.

```
const user = firebase.auth().currentUser;

user.updatePassword(newPassword).then(() => {
  // Password updated
}).catch((error) => {
  // An error happened
});
```

Reset della password:

- Per consentire agli utenti di reimpostare la propria password dimenticata, puoi utilizzare la funzione `sendPasswordResetEmail()`.

```
firebase.auth().sendPasswordResetEmail(email)
  .then(() => {
    // Password reset email sent
  })
  .catch((error) => {
    // Handle password reset error
  });
```

Cancellare un utente:

- Per cancellare un utente, puoi utilizzare la funzione `delete()`.

```
const user = firebase.auth().currentUser;

user.delete().then(() => {
```

```
// User deleted
}).catch((error) => {
  // An error happened
});
```

OAuth2 Authentication:

- Firebase offre anche il supporto per l'autenticazione OAuth2 con provider come Google, Facebook, Twitter, ecc.
- Puoi abilitare l'autenticazione OAuth2 sul console Firebase e integrare facilmente l'autenticazione OAuth2 nel tuo frontend o backend utilizzando le API Firebase appropriate.