Firebase

What is Firebase?

Firebase is Google's comprehensive app development platform that provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users, store data, and more.

Why is Firebase interesting?

- **Rapid Development:** Build apps faster with pre-built backend services
- **Real-time Capabilities:** Synchronize data across clients instantly
- **Scalability:** Automatic scaling without infrastructure management
- **Integrated Ecosystem:** All services work seamlessly together
- **Cross-platform:** Web, iOS, Android, and more
- **Built-in Authentication:** Easy and secure user login methods
- **Hosting & Functions:** Deploy static and dynamic content with ease
- **Analytics & Monitoring:** Track usage and fix issues with built-in tools
- **Security Rules:** Fine-grained access control for data and storage
- **Generous Free Tier:** Ideal for small projects and MVPs

Limitations to consider

- **Query limitations:** Limited complex queries compared to SQL
- **Vendor lock-in:** Difficult to migrate away from Firebase
- **Cost scaling:** Can become expensive with high usage
- **Regional restrictions:** Limited control over data location
- **Real-time limits:** Connection limits for concurrent users

Best practices

1. Use Firestore Subcollections for hierarchical data
2. Implement proper security rules before going to production
3. Use Cloud Functions for server-side logic and data validation
4. Optimize queries by creating appropriate indexes
5. Handle offline scenarios in your client code
6. Use Firebase Analytics to understand user behavior
7. Implement proper error handling for all Firebase operations

Firebase pricing model

Firebase follows a pay-as-you-go model with a generous free tier, making it accessible for small and medium businesses to start without upfront costs.

Cost components

- **Firestore:** A NoSQL document database. Include elements like the number of reads, writes, deletes, ...
- **Authentication:** Manages use identity and login. Include elements like phone authentication
- **Hosting:** Hosting static files and dynamic content via Cloud Functions ou CLoud run. Include elements like storage and data transfer
- **Cloud Functions:** Serverless backend code triggered by events. Include elements like compute time, number of invocations, ...
- **Storage:** File storage. Include elements like storage, downloads and operations
- **Real-time Database:** A different NoSQL database with real-time synching. Include lements like data storage and download data

Free Tier (Spark Plan)

Service	Allocation
Firestore	50K reads, 20K writes, 20K deletes per day
Authentication	Unlimited users
Hosting	10GB storage, 125 operations per day
Cloud Functions	125K invocations, 40K GB/s
Storage	5GB total storage
Real-Time Database	1GB storage, 10GB/month transfer

Cost for a medium platform

- 2,000 active users
- 100K database reads/day (3M/month)
- 10K database writes/day (300K/month)
- 50GB file storage
- 100GB hosting bandwidth/month
- 10K Cloud Function invocations/day

Monthly Cost Breakdown:
- Firestore: $18 (reads) + $9 (writes) = $27
- Storage: $1.25
- Hosting: $8.50 (bandwidth overage)
- Cloud Functions: $2.40
- **Total: ~$40/month**

Getting started

Prerequisites

Install Node.js (version 14 or later)
Install Firebase CLI

```
npm install -g firebase-tools
```

Project Setup

Login to Firebase

```
firebase login
```

Initialize new project

```
firebase init
```

Select services you want to use:
- Firestore (database)
- Functions (serverless)
- Hosting (web hosting)
- Storage (file storage)

Configuration

firebase-config.js

```
import { initializeApp } from 'firebase/app';

const firebaseConfig = {
  apiKey: "your-api-key",
  authDomain: "your-project.firebaseapp.com",
  projectId: "your-project-id",
  storageBucket: "your-project.appspot.com",
  messagingSenderId: "123456789",
  appId: "your-app-id"
};

const app = initializeApp(firebaseConfig);
export default app;
```

Core Operations (hello-world exemple)

Authentication

```
import { getAuth, createUserWithEmailAndPassword,
signInWithEmailAndPassword } from 'firebase/auth';

const auth = getAuth();

// Create user
createUserWithEmailAndPassword(auth, email,
password)
  .then((userCredential) => {
    const user = userCredential.user;
  });

// Sign in user
signInWithEmailAndPassword(auth, email, password)
  .then((userCredential) => {
    const user = userCredential.user;
  });

// Sign out
auth.signOut();
```

Hosting

```
# Build your app
npm run build
# Deploy to Firebase Hosting
firebase deploy --only hosting
# Deploy specific functions
firebase deploy --only functions
# Deploy everything
firebase deploy
```

Firestore Database

```
import { getFirestore, collection, addDoc, getDocs,
onSnapshot } from 'firebase/firestore';

const db = getFirestore();

// Add document
addDoc(collection(db, "tasks"), {
  title: "Learn Firebase",
  completed: false,
  createdAt: new Date()
});

// Read documents
const querySnapshot = await getDocs(collection(db,
"tasks"));
querySnapshot.forEach((doc) => {
  console.log(doc.id, " => ", doc.data());
});

// Real-time listener
onSnapshot(collection(db, "tasks"), (snapshot) => {
  snapshot.forEach((doc) => {
    console.log(doc.data());
  });
});
```

Cloud Storage

```
import { getStorage, ref, uploadBytes,
getDownloadURL } from 'firebase/storage';

const storage = getStroage();

// Upload file
const fileRef = ref(storage, ‘uploads/’ +
file.name);
uploadBytes(fileRef, file).then((snapshot) => {
  console.log(’File uploaded successfully’);
});

// Get download URL
getDownloadURL(fileRef).then(url) => {
  console.log(’File available at:’, url);
});
```

Cloud Functions

```
// functions/index.js
const functions = require(’firebase-functions’);
const admin = require(’firebase-admin’);

admin.initializeApp();

// HTTP Cloud Function
exports.helloWorld =
functions.https.onRequest((request, response) => {
  response.send("Hello from Firebase!");
});

// Firestore Trigger
exports.createUserProfile = functions.firestore
  .document('users/{userId}')
  .onCreate((snap, context) => {
    const userData = snap.data();
    console.log('New user created:', userData);
  });
```

Security Rules

Firestore Rules

```
// firestore.rules
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    // Users can only read/write their own data
    match /users/{userId} {
      allow read, write: if request.auth ≠ null &&
request.auth.uid == userId;
    }

    // Public read, authenticated write
    match /tasks/{taskId} {
      allow read: if true;
      allow write: if request.auth ≠ null;
    }
  }
}
```

Storage Rules

```
// storage.rules
rules_version = '2';
service firebase.storage {
  match /b/{bucket}/o {
    match /uploads/{userId}/{allPaths=**} {
      allow read, write: if request.auth ≠ null &&
request.auth.uid == userId;
    }
  }
}
```

Developement commands

Commands

```
# Start local emulators
firebase emulators:start
# Deploy functions only
firebase deploy --only functions
# Deploy hosting only
firebase deploy --only hosting
# View logs
firebase functions:log
# Set environment config
firebase functions:config:set api.key="your-key"
# Get environment config
firebase functions:config:get
```