

Firestore is back at Google I/O on May 14! [Register now.](https://io.google/2024/?utm_source=firebase&utm_medium=embedded_marketing&utm_campaign=&utm_content=)
(https://io.google/2024/?utm_source=firebase&utm_medium=embedded_marketing&utm_campaign=&utm_content=)

Cloud Firestore



Use our flexible, scalable NoSQL cloud database, built on Google Cloud infrastructure, to store and sync data for client- and server-side development.

Cloud Firestore is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud. Like Firebase Realtime Database, it keeps your data in sync across client apps through realtime listeners and offers offline support for mobile and web so you can build responsive apps that work regardless of network latency or Internet connectivity. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including Cloud Functions.

Introducing Cloud Firestore



[Get started](/docs/firestore/quickstart) (/docs/firestore/quickstart)

Key capabilities

Flexibility

The Cloud Firestore data model supports flexible, hierarchical data structures. Store your data in documents, organized into collections. Documents can contain complex nested objects in addition to subcollections.

Expressive querying

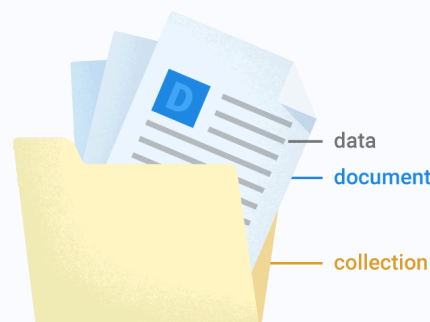
In Cloud Firestore, you can use queries to retrieve individual, specific documents or to retrieve all the documents in a collection that match your query parameters. Your queries can include multiple, chained filters and combine filtering and sorting. They're also indexed by default, so query performance is proportional to the size of your result set, not your data set.

Realtime updates	Like Realtime Database, Cloud Firestore uses data synchronization to update data on any connected device. However, it's also designed to make simple, one-time fetch queries efficiently.
Offline support	Cloud Firestore caches data that your app is actively using, so the app can write, read, listen to, and query data even if the device is offline. When the device comes back online, Cloud Firestore synchronizes any local changes back to Cloud Firestore.
Designed to scale	Cloud Firestore brings you the best of Google Cloud's powerful infrastructure: automatic multi-region data replication, strong consistency guarantees, atomic batch operations, and real transaction support. We've designed Cloud Firestore to handle the toughest database workloads from the world's biggest apps.

How does it work?

Cloud Firestore is a cloud-hosted, NoSQL database that your Apple, Android, and web apps can access directly via native SDKs. Cloud Firestore is also available in native Node.js, Java, Python, Unity, C++ and Go SDKs, in addition to REST and RPC APIs.

Following Cloud Firestore's NoSQL data model, you store data in documents that contain fields mapping to values. These documents are stored in collections, which are containers for your documents that you can use to organize your data and build queries. Documents support many different [data types](/docs/firestore/manage-data/data-types) (</docs/firestore/manage-data/data-types>), from simple strings and numbers, to complex, nested objects. You can also create subcollections within documents and build hierarchical data structures that scale as your database grows. The Cloud Firestore [data model](/docs/firestore/data-model) (</docs/firestore/data-model>) supports whatever data structure works best for your app.



Additionally, querying in Cloud Firestore is expressive, efficient, and flexible. Create shallow queries to retrieve data at the document level without needing to retrieve the entire collection, or any nested subcollections. Add sorting, filtering, and limits to your queries or cursors to paginate

your results. To keep data in your apps current, without retrieving your entire database each time an update happens, add realtime listeners. Adding realtime listeners to your app notifies you with a data snapshot whenever the data your client apps are listening to changes, retrieving only the new changes.

Protect access to your data in Cloud Firestore with Firebase Authentication and Cloud Firestore Security Rules for Android, Apple platforms, and JavaScript, or Identity and Access Management (IAM) for server-side languages.

Implementation path

1

Integrate the Cloud Firestore SDKs

Quickly include clients via Gradle, CocoaPods, or a script include.

2

Secure your data

Use Cloud Firestore Security Rules or Identity and Access Management (IAM) to secure your data for mobile/web and server development, respectively.

3

Add Data

Create documents and collections in your database.

4

Get Data

Create queries or use realtime listeners to retrieve data from the database.

Next steps

- [Get started](/docs/firestore/quickstart) (/docs/firestore/quickstart) with Cloud Firestore — set up your database, then add data and start reading it.
- Learn more about the Cloud Firestore [data model](/docs/firestore/data-model) (/docs/firestore/data-model).
- Explore the [differences between Realtime Database and Cloud Firestore](/docs/firestore/rtdb-vs-firestore) (/docs/firestore/rtdb-vs-firestore).

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 4.0 License](https://creativecommons.org/licenses/by/4.0/) (<https://creativecommons.org/licenses/by/4.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see the [Google Developers Site Policies](https://developers.google.com/site-policies) (<https://developers.google.com/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2024-04-02 UTC.