Cloud Firestore: Node.js Client

(https://github.com/googleapis/nodejs-firestore)

`release level` `stable`
(https://cloud.google.com/terms/launch-stages)
`npm` `v7.6.0`
(https://www.npmjs.org/package/@google-cloud/firestore)

This is the Node.js Server SDK for Google Cloud Firestore
(https://firebase.google.com/docs/firestore/). Google Cloud Firestore is a NoSQL document database built for automatic scaling, high performance, and ease of application development.

This Cloud Firestore Server SDK uses Google's Cloud Identity and Access Management for authentication and should only be used in trusted environments. Your Cloud Identity credentials allow you bypass all access restrictions and provide read and write access to all data in your Cloud Firestore project.

The Cloud Firestore Server SDKs are designed to manage the full set of data in your Cloud Firestore project and work best with reliable network connectivity. Data operations performed via these SDKs directly access the Cloud Firestore backend and all document reads and writes are optimized for high throughput.

Applications that use Google's Server SDKs should not be used in end-user environments, such as on phones or on publicly hosted websites. If you are developing a Web or Node.js application that accesses Cloud Firestore on behalf of end users, use the firebase Client SDK.

Note: This Cloud Firestore Server SDK does not support Firestore databases created in Datastore mode (https://cloud.google.com/datastore/docs/firestore-or-datastore#in_datastore_mode). To access these databases, use the Datastore SDK (https://www.npmjs.com/package/@google-cloud/datastore).

A comprehensive list of changes in each version may be found in the CHANGELOG (https://github.com/googleapis/nodejs-firestore/blob/main/CHANGELOG.md).

- Cloud Firestore Node.js Client API Reference
  (https://cloud.google.com/nodejs/docs/reference/firestore/latest)

- Cloud Firestore Documentation (https://cloud.google.com/firestore)

- github.com/googleapis/nodejs-firestore (https://github.com/googleapis/nodejs-firestore)

Read more about the client libraries for Cloud APIs, including the older Google APIs Client Libraries, in Client Libraries Explained
 (https://cloud.google.com/apis/docs/client-libraries-explained).

Table of contents:

Quickstart

Before you begin

1. Select or create a Cloud Platform project (https://console.cloud.google.com/project).

2. Enable the Cloud Firestore API
   (https://console.cloud.google.com/flows/enableapi?apiid=firestore.googleapis.com).

3. Set up authentication with a service account
   (https://cloud.google.com/docs/authentication/getting-started) so you can access the API
   from your local workstation.

Installing the client library

```
npm install @google-cloud/firestore
```

Using the client library

```javascript
const {Firestore} = require('@google-cloud/firestore');

// Create a new client
const firestore = new Firestore();

async function quickstart() {
  // Obtain a document reference.
  const document = firestore.doc('posts/intro-to-firestore');

  // Enter new data into the document.
  await document.set({
    title: 'Welcome to Firestore',
    body: 'Hello World',
  });
  console.log('Entered new data into the document');

  // Update an existing document.
  await document.update({
    body: 'My first Firestore app',
  });
  console.log('Updated an existing document');

  // Read the document.
  const doc = await document.get();
  console.log('Read the document');

  // Delete the document.
  await document.delete();
  console.log('Deleted the document');
}
quickstart();
```

Samples

Samples are in the __samples/__ (https://github.com/googleapis/nodejs-firestore/tree/main/samples)
directory. Each sample's `README.md` has instructions for running its sample.

Sample	Source Code	Try it
Limit-to-last-query	source code (https://github.com/googleapis/nodejs-firestore/blob/main/samples/limit-to-last-query.js)	OPEN IN GOOGLE CLOUD SHELL (https://console.cloud.google.com/cloudshell/oper git_repo=https://github.com/googleapis/nodejs-firestore&page=editor&open_in_editor=samples/lim query.js,samples/README.md)
Quickstart	source code (https://github.com/googleapis/nodejs-firestore/blob/main/samples/quickstart.js)	OPEN IN GOOGLE CLOUD SHELL (https://console.cloud.google.com/cloudshell/oper git_repo=https://github.com/googleapis/nodejs-firestore&page=editor&open_in_editor=samples/qui
Solution-counters	source code (https://github.com/googleapis/nodejs-firestore/blob/main/samples/solution-counters.js)	OPEN IN GOOGLE CLOUD SHELL (https://console.cloud.google.com/cloudshell/oper git_repo=https://github.com/googleapis/nodejs-firestore&page=editor&open_in_editor=samples/sol counters.js,samples/README.md)

The Cloud Firestore Node.js Client API Reference (https://cloud.google.com/nodejs/docs/reference/firestore/latest) documentation also contains samples.

Supported Node.js Versions

Our client libraries follow the Node.js release schedule (https://github.com/nodejs/release#release-schedule). Libraries are compatible with all current *active* and *maintenance* versions of Node.js. If you are using an end-of-life version of Node.js, we recommend that you update as soon as possible to an actively supported LTS version.

Google's client libraries support legacy versions of Node.js runtimes on a best-efforts basis with the following warnings:

- Legacy versions are not tested in continuous integration.

- Some security patches and features cannot be backported.

- Dependencies cannot be kept up-to-date.

Client libraries targeting some end-of-life versions of Node.js are available, and can be installed through npm dist-tags (https://docs.npmjs.com/cli/dist-tag). The dist-tags follow the naming convention `legacy-(version)`. For example, `npm install @google-cloud/firestore@legacy-8` installs client libraries for versions compatible with Node.js 8.

Versioning

This library follows Semantic Versioning (http://semver.org/).

This library is considered to be **stable**. The code surface will not change in backwards-incompatible ways unless absolutely necessary (e.g. because of critical security issues) or with an extensive deprecation period. Issues and requests against **stable** libraries are addressed with the highest priority.

More Information: Google Cloud Platform Launch Stages (https://cloud.google.com/terms/launch-stages)

Contributing

Contributions welcome! See the Contributing Guide (https://github.com/googleapis/nodejs-firestore/blob/main/CONTRIBUTING.md).

Please note that this `README.md`, the `samples/README.md`, and a variety of configuration files in this repository (including `.nycrc` and `tsconfig.json`) are generated from a central template. To edit one of these files, make an edit to its templates in directory (https://github.com/googleapis/synthtool).

License

Apache Version 2.0

See LICENSE (https://github.com/googleapis/nodejs-firestore/blob/main/LICENSE)