ature is in a pre-release state and might change or have limited support. For more information, see the product launc
s (/products/#product-launch-stages).

Cloud Functions can handle events in Cloud Firestore with no need to update client code. You can read and/or update Cloud Firestore in response to these events using the Firestore APIs and client libraries (https://cloud.google.com/firestore/docs/apis).

In a typical lifecycle, a Cloud Firestore function does the following:

1. Waits for changes to a particular document.
2. Triggers when an event occurs and performs its tasks.
3. Receives a data object with a snapshot of the affected document. For `write` or `update` events, the data object contains snapshots representing document state before and after the triggering event.

Cloud Firestore supports `create`, `update`, `delete`, and `write` events. The `write` event encompasses all modifications to a document.

Event type	Trigger
`providers/cloud.firestore/eventTypes/document.create`	Triggered when a document is written to for the first time.
`providers/cloud.firestore/eventTypes/document.update`	Triggered when a document already exists and has any value changed.
`providers/cloud.firestore/eventTypes/document.delete`	Triggered when a document with data is deleted.
`providers/cloud.firestore/eventTypes/document.write`	Triggered when a document is created, updated or deleted.

Triggers happen at the document-level. It is not possible to create a trigger for a specific field only.

Wildcards are written in triggers using curly braces, as follows: `/projects/YOUR_PROJECT_ID/databases/(default)/documents/collection/{document_wildcard}`

To trigger your function, specify a document path to listen to. Functions only respond to document changes, and cannot monitor specific fields or collections. Below are a few examples of valid document paths:

- `users/marie`: valid trigger. Monitors a single document, `/users/marie`.
- `users/{username}`: valid trigger. Monitors all user documents. Wildcards are used to monitor all documents in the collection.
- `users/{username}/addresses`: *invalid* trigger. Refers to the subcollection `addresses`, not a document.
- `users/{username}/addresses/home`: valid trigger. Monitors the home address document for all users.
- `users/{username}/addresses/{addressId}`: valid trigger. Monitors all address documents.

If you do not know the specific document you want to monitor, use a `{wildcard}` instead of the document ID:

- `users/{username}` listens for changes to all user documents.

In this example, when any field on any document in `users` is changed, it matches a wildcard called `{username}`.

If a document in `users` has subcollections, and a field in one of those subcollections' documents is changed, the `{username}` wildcard is *not* triggered.

Wildcard matches are extracted from document paths and stored in `event.params`. You can define as many wildcards as you like to substitute explicit collection or document IDs.

This trigger invokes your function with an event similar to the one shown below:

Each `Document` object contains one or more `Value` objects. See the `Value` documentation (/firestore/docs/reference/rest/v1/Value) for type references. This is especially useful if you're using a typed language (like Go) to write your functions.

The sample Cloud Function below prints the fields of a triggering Cloud Firestore event:

The example below retrieves the value added by the user, converts the string at that location to uppercase, and replaces the value with the uppercase string:

The following `gcloud` command deploys a function that is triggered by write events on the document `/messages/{pushId}`:

Argument	Description
`--runtime` *`RUNTIME`*	The name of the runtime you are using. For a complete list, see the **`gcloud`** reference (/sdk/gcloud/reference/functions/deploy#--runtime).
`--trigger-event` *`NAME`*	The event type that the function will monitor for (one of **`write`**, **`create`**, **`update`** or **`delete`**).
`--trigger-resource` *`NAME`*	The fully qualified database path to which the function will listen. This should conform to the following format: **`projects/`***`YOUR_PROJECT_ID`***`/databases/(default)/documents/`***`PATH`** The **`{pushId}`** text is a wildcard parameter described above in Specifying the document path (#specifying_the_document_path).

Firestore triggers for Cloud Functions is a beta feature with some known limitations:

- It can take up to 10 seconds for a function to respond to changes in Firestore.
- Ordering is not guaranteed. Rapid changes can trigger function invocations in an unexpected order.

- Events are delivered at least once, but a single event may result in multiple function invocations. Avoid depending on exactly-once mechanics, and write idempotent functions (https://cloud.google.com/blog/products/serverless/cloud-functions-pro-tips-building-idempotent-functions)
  .
- Firestore triggers for Cloud Functions is available only for Firestore in Native mode (https://cloud.google.com/firestore/docs/firestore-or-datastore). It is not available for Firestore in Datastore mode.