

zapier.georgelambert.org · Markdown indexes

VERAE TIME × ZAPIER

04 register a hash

site/user-docs/04-register-a-hash.md

4. Register a SHA-256 (create timestamp)

Two Zapier **creates**:

Action	What happens	What you get back
Create Timestamp (async)	Submit hash, return immediately	<code>jobId</code> , <code>sha256</code> , <code>existing</code>
Create Timestamp and Wait	Submit hash, hold the Zap step until the certificate exists (or timeout)	<code>status</code> , <code>receipts[]</code> , <code>completedAt</code>

Inputs

- **Data** — raw text/bytes. Middleware hashes it with SHA-256. Prefer this when the previous Zap step has the document contents.
- **SHA256** — 64 hex characters if you already hashed the file outside Zapier (recommended for large files: hash locally, send only the digest).
- **Hash algorithm** — `SHA256` (default).

Do not send the whole PDF to the blockchain. Only the digest is sealed.

Already registered

If that SHA-256 was sealed before, you get the **original** `jobId` and certificate. A second chain write is not created. `existing: true`.

Example Zap

1. Trigger: “New file in Google Drive”.

2. Action (optional): compute SHA-256 in a Code step, or pass file contents as **Data**.
3. Action: **Create Timestamp and Wait**.
4. Action: store `jobId`, `sha256`, and certificate in your sheet/CRM.

Simulator: scenario **Register** (wait + `includeAttached`).