

api

Tool API

Server name: `verae-timestamping`.

stamp

Argument	Default	Meaning
<code>sha256</code>	required	64 hex characters
<code>scope</code>	<code>local</code>	<code>local</code> , <code>shared</code> , or <code>global</code>
<code>cube_id</code>	empty	Stored on the local receipt. Sent only as a local field on the global receipt object; the VeraeTime POST body is the digest
<code>org_id</code>	empty	Local idempotency key. Ignored by the global client

Returns { "receipts": [Receipt] }.

Receipt fields: `receipt_id`, `hash`, `scope`, `cube_id`, `org_id`, `created_at`, `merkle`, `restamp_of`.

verify

Argument	Default	Meaning
<code>receipt_id</code>	required	Local id or VeraeTime certificate / job id

Argument	Default	Meaning
<code>want_scope</code>	empty	<code>global</code> forces VeracTime. A weaker scope fails closed when the stored scope ranks lower

Returns `{ "valid": bool, "error": string or null, "receipt": Receipt or null }`.

Scope rank: `local < shared < global`.

hash_report

Argument	Default	Meaning
<code>sha256</code>	required	Digest to list
<code>org_id</code>	empty	Filter when set
<code>want_scope</code>	empty	Drop receipts below this rank

Returns `{ "receipts": [ ... ] }` from the local file. Global-only certificates are not listed.

login

Argument	Default	Meaning
<code>username</code>	required	VeracTime account
<code>password</code>	required	VeracTime password

POST `/auth/login`. Stores `access_token`, `id_token`, `auth_token`, `expires_at`, `username`, and `password` in the auth properties file (mode 0600) and replaces the in-memory JWT. Returns `username`, `role`, and `expires_at`. The JWT is not returned.

session

No arguments. GET `/auth/validate` with the stored JWT.

Returns `valid`, `username`, `role`, and `expires_at`.

signup

Argument	Default	Meaning
username	required	New account
password	required	New password

POST /auth/signup with no bearer token. The timestamping service creates the account with role **user**. Returns **id**, **username**, and **role**. Does not replace the JWT stored for **stamp**. A new account calls **login** to become the stored session. **create_user** remains the admin path and can set **admin**.

create_user

Argument	Default	Meaning
username	required	New account
password	required	New password
role	required	user or admin

POST /auth/users with the stored JWT. Refuses unless **session** reports role **admin**. Returns **id**, **username**, and **role**. There is no public signup; the first admin already exists on the timestamping service.

A stored login is refreshed when **stamp** or **verify** runs and **expires_at** is missing or within 5 minutes, if **username** and **password** are in the auth file.

Environment

Name	Default
VERAE_GLOBAL_URL	https://api.veraetime.net
VERAE_GLOBAL_TOKEN	empty (then auth.properties)
VERAE_AUTH_PROPERTIES	~/.verae/auth.properties
VERAE_LEDGER_PATH	~/.verae/mcp-timestamp-local.json
VERAE_MCP_TRANSPORT	stdio
VERAE_MCP_HOST	127.0.0.1
VERAE_MCP_PORT	8000
VERAE_MCP_CORS	unset. 1 adds CORS for the sample page

Client config (stdio)

```
{  
  "mcpServers": {
```

```
    "verae-timestamping": {  
      "command": "uv",  
      "args": ["run", "--directory", "/Users/marchon/research/modelcontextprotocol/verae-mcp"]  
    }  
  }  
}
```