

zapier.georgelambert.org · Markdown indexes

VERAE TIME × ZAPIER

Hop-trace simulator

[site/packages/verae-zapier-simulator/README.md](https://zapier.georgelambert.org/site/packages/verae-zapier-simulator/README.md)

Verae Zapier interface simulator

In-process stand-in for the Zapier editor. Run a step, then read a **trace console** of every hop from Zapier input through zappier-edge, middleware, splitter, mock chain, WORM archives, and tree nodes, back to the Zapier output.

Use it to validate messaging and to catch errors, delays, failures, and recoveries **before** the Zapier app is pushed live.

```
cd packages/verae-zapier-simulator
npm test
npm start # http://127.0.0.1:3847/
```

What it simulates

Zapier action	Path
Sign up	zappier portal → API key
Connect	Zapier test auth (<code>x-api-key</code>)
Create Timestamp (async)	202 <code>jobId</code> + <code>jobs.watch</code>
Create Timestamp and Wait	hold until <code>jobs.events</code>
Find Timestamp by SHA256	central Verae chain only
Find Hash (tree nodes)	chain miss → broadcast <code>verae.archive.query</code> <code>kinds= tree</code>
Create Batch	Merkle root sealed on chain; leaves on tree nodes
Verify	mock certificate

Faults you can inject: edge 401/402, chain timeout, NATS watch/events drop, archive put fail, all archives silent, one tree node down, hop delays, then **recover** (retry).

The monitor panel flags broken assumptions (Zapier talking to NATS, file bytes on chain, bloom miss sending a reply) and suggests changes (tree-node search, wait+hook pairing, 402 upgrade URL, archive quorum).

Catalog: <https://zapier.georgelambert.org/packages/verae-zapier-simulator/README.pdf>

Live sample UI (loopback): <http://127.0.0.1:3847/> — also inside the operator console Trace tab.

Static copy: <https://zapier.georgelambert.org/simulator/>

Clone: <ssh://git@git.georgelambert.org:2223/marchon/verae-zapier-simulator.git>