

zapier.georgelambert.org · Markdown indexes

VERAE TIME × ZAPIER

timestampService

site/docs/modules/verae-zapier-middlewre/services/timestampService.md

verae-zapier-middlewre/services/ timestampService

Package: verae-zapier-middlewre

Source: packages/verae-zapier-middlewre/src/services/timestampService.js

Lines: 171

What this module is

Implementation module in `verae-zapier-middlewre`. The tables below are extracted from the source (signatures + JSDoc).

Exports

`createTimestamp`, `createTimestampAndWait`, `createBatchTimestamp`, `getJobStatus`,
`getBatchJobStatus`, `getJobVerification`, `lookupHash`

Types / interfaces / classes

None extracted.

Functions

Name	Parameters	Param types (JSDoc)	Returns	Calls (same file / helpers)
enqueueWatchForJob	ctx, jobId	ctx: object, jobId: string	unknown	see Call graph
createTimestamp	ctx, body	ctx: object	Promise<{ jobId: string, sha256?: string, existing?: boolean —>}	see Call graph
createTimestampAndWait	ctx, body	ctx: object	Promise<object> — StatusResponse	see Call graph
createBatchTimestamp	ctx, body	ctx: object	unknown	see Call graph
getJobStatus	ctx, jobId	ctx: object, jobId: string	unknown	see Call graph
getBatchJobStatus	ctx, body	ctx: object	unknown	see Call graph
getJobVerification	ctx, jobId	ctx: object, jobId: string	unknown	see Call graph
lookupHash	ctx, sha256, opts = {}	ctx: object, sha256: string	unknown	see Call graph

Methods (class / object)

Name	Parameters
enqueueJob	{ tenantId: ctx.tenantId, jobId, veraeToken: ctx.veraeToken, traceId, }
checkEntitlement	ctx.tenantId, 'timestamp'
recordUsage	ctx.tenantId, 'timestamp'
checkEntitlement	ctx.tenantId, 'batch_timestamp', { amount: itemCount }
recordUsage	ctx.tenantId, 'batch_timestamp', { amount: itemCount }
recordUsage	ctx.tenantId, 'status'
recordUsage	ctx.tenantId, 'status', { amount: body.jobIds?.length ?? 1 }
recordUsage	ctx.tenantId, 'status'
recordUsage	ctx.tenantId, 'status'

What it imports / requires

- ../config.js
- ../clients/veraeClient.js
- ../store/jobWatchers.js
- ../store/tokenRefs.js
- ./entitlementService.js
- ../debug/logger.js
- ../debug/trace-context.js

Call graph (identifiers invoked)

createDebugger, enqueueWatchForJob, getTraceId, import, enqueueWatch, issueTokenRef, debug, enqueueJob, createTimestamp, checkEntitlement, recordUsage, createTimestampAndWait, waitForJobEvent, getStatus, waitForJob, createBatchTimestamp, getJobStatus, getBatchJobStatus, getBatchStatus, getJobVerification, lookupHash

Each identifier is a call site in this file. Follow the import list to see the defining module; open that module's MD for parameter and return types.

Return values (how to read this)

- HTTP route handlers return Express `res.json(...)` bodies (see route docs).
- Zapier `perform` functions return a **single object** (creates) or an **array** (triggers/searches).
- Pricing functions return integer **cents** on `Quote.totalCents`.