CommitLockService

September 4, 2026

CommitLockService

Kind: Class

Source: `atloria-monorepo/apps/api/src/docs-repo/commit-lock.ts`

Per-project commit mutex for the docs substrate.

FAIL-CLOSED, deliberately the OPPOSITE of the technical-docs org mutex (which fails OPEN
so a Redis blip can't dead-lock the parse lane). Here, committing without the lock could
corrupt a project's git tree with a concurrent write, so if Redis is unreachable we DENY the commit (throw LockUnavailableError) and let the job requeue. A dedicated ioredis connection (lazyConnect, maxRetriesPerRequest:1, offline queue off) keeps a Redis outage
from stalling everything else.

Implements: `OnModuleDestroy`

Methods

Method	Signature	Returns
`acquire`	`acquire(projectId: string, ttlMs: undefined)`	`Promise<string`
`release`	`release(projectId: string, token: string)`	`Promise<void>`
`renew`	`renew(projectId: string, token: string, ttlMs: undefined)`	`Promise<boolean>`
`withLock`	`withLock(projectId: string, fn: () => Promise<T>, ttlMs: undefined)`	`Promise<T>`
`onModuleDestroy`	`onModuleDestroy()`	`void`

Where it refuses work

- `CommitLockService` stops the work with `LockHeldError` when `!token`.

When something fails

- `CommitLockService` handles failure in 3 places: it logs it and continues in 1, turns it into a return value in 1, and lets it reach the caller in 1.

Referenced By

- `DocsRepoModule` (MODULE_PROVIDES)
- `DocsRepoModule` (MODULE_EXPORTS)