# CrWorker

September 4, 2026

## CrWorker

**Kind:** Class

**Source:** `atloria-monorepo/apps/api/src/change-request/cr.worker.ts`

CrWorker — the single, gated consumer of the 'docs-cr' queue (S2.1 branch lifecycle
+
S2.3 merge writeback).

Only spins up where DOCS_REPO_WORKER==='true' (the deployment that mounts the docs-repos
PVC); api pods enqueue-only. Jobs:

- 'cr-open' stages the CR's proposed content on a cr/ branch cut from main.
- 'cr-merge' THE writeback: conflict-gates each file against CURRENT POSTGRES (the serving
  source of truth — NOT lagging git main), then writes the effective content
  back to Postgres through DocumentService under a per-document CAS (revisions
  created; the outbox → reconciler then projects main — that projection commit
  is the merge commit, authored by the MERGER, GitHub-style). The CR branch is
  never git-merged and is KEPT for audit.
- 'cr-refresh' recomputes diffCache/conflictState against current main; status untouched.
- 'cr-restage' commits a pendingResolution onto the CR branch as the RESOLVER's author.

Git-branch work runs under the same FAIL-CLOSED per-project commit lock the committer uses;
merge writeback is DB-only and deliberately does NOT hold the lock (DocumentService calls
must never sit inside a git-lock window).

INVARIANT: the repo is checked back out to `main` in a finally, even when staging throws
mid-way — the reconciler commits to whatever branch is checked out, so a repo left on a CR
branch would silently route the next DB-wins rebuild onto the proposal branch.

**Implements:** `OnModuleInit`, `OnModuleDestroy`

## Methods

| Method | Signature | Returns |
|---|---|---|
| `onModuleInit` | `onModuleInit()` | `void` |
| `dispatch` | `dispatch(data: DocsCrJob)` | `Promise<void>` |
| `handle` | `handle(crId: string)` | `Promise<void>` |
| `handleMerge` | `handleMerge(crId: string)` | `Promise<void>` |
| `handleRestage` | `handleRestage(crId: string)` | `Promise<void>` |
| `openChangeRequest` | `openChangeRequest(crId: string)` | `Promise<void>` |
| `mergeChangeRequest` | `mergeChangeRequest(crId: string)` | `Promise<void>` |
| `refreshChangeRequest` | `refreshChangeRequest(crId: string)` | `Promise<void>` |
| `restageResolution` | `restageResolution(crId: string)` | `Promise<void>` |
| `onModuleDestroy` | `onModuleDestroy()` | `Promise<void>` |

## Where it refuses work

- `CrWorker` stops the work with `CrPermanentError` when `files.length === 0` — “change request has no files”.
- `CrWorker` stops the work with `CrPermanentError` when `files.length === 0 || files.some((f) => !f.path && !f.delete)` — “change request has no staged files to merge”.
- `CrWorker` stops the work with `CrPermanentError` when `!parseDocsPath(f.createPath)`.
- `CrWorker` stops the work with `CrPermanentError` when `!repoPath` — “document not in served set”.
- `CrWorker` stops the work with `CrPermanentError` when `!parts`.
- `CrWorker` stops the work with `CrPermanentError` when `!served`.

## When something fails

- `CrWorker` handles failure in 12 places: it lets it reach the caller in 8, logs it and continues in 2, turns it into a return value in 1, and discards it silently in 1. A failure discarded silently leaves no trace for whoever debugs this later.

## Referenced By

- `ChangeRequestModule` (MODULE_PROVIDES)