

DocsTrueupService

September 4, 2026

DocsTrueupService

Kind: Service

Source: `atloria-monorepo/apps/api/src/technical-docs/docs-trueup.service.ts`

Docs true-up GC (docs-custody). The delta pipelines (S3.1 screenshots, S4 techdocs) RETAIN a

page whose source screen/entity disappeared, badging it (`metadata.screenRemoved` / `metadata.techEntityRemoved` + a `removedAt` stamp) instead of deleting it — so a transient capture/parse miss can never destroy a good page or its deep links. Those badges were

write-only: nothing ever cleaned them, so dead pages accumulated forever. This service is the

missing GC: it HARD-DELETES a badged page only once it has been continuously badged past a

grace period (a reappeared source clears the badge at upsert time, so a page still badged after

the grace is genuinely gone), and it self-corrects — a techdocs page whose entity is back in the

live snapshot has its badge cleared instead of being deleted.

Deleting a Document cascades its OWNED children (comments, suggestions, revisions, versions,

collaboration sessions, code symbols — all `onDelete: Cascade`); no FK RESTRICTs it (verified

against the schema). Deletes run one-at-a-time with try/catch so a single unexpected FK edge

can't abort the whole sweep.

Gated OFF by default (`TRUEUP_GC_ENABLED=true` to run). Grace via `TRUEUP_GC_GRACE_DAYS` (default 14).

`DocsTrueupService` is the garbage collector for retained technical-documentation pages whose source screen or tech entity has disappeared. It keeps pages during transient pipeline misses, then permanently deletes pages that remain continuously badged beyond the configured grace period; if a tech entity is present again in the live snapshot, it clears the stale removal badge instead.

The service is disabled by default and runs only when `TRUEUP_GC_ENABLED=true`. Deletions are isolated per document so one unexpected failure does not stop the overall sweep.

Methods

Method	Signature	Returns	Description
<code>healActiveVersions</code>	<code>healActiveVersions()</code>	<code>Promise<void></code>	Self-heal: a (project, audienceScope) that has PUBLISHED versions but NO ACTIVE one was left activeless by a stray demotion (observed once on Healthy: v52 de...
<code>scheduledSweep</code>	<code>scheduledSweep()</code>	<code>Promise<void></code>	Daily sweep across every project that currently has badged-removed pages.
<code>garbageCollect</code>	<code>garbageCollect(projectId: string)</code>	<code>Promise<{ deleted: number; cleared: number; skipped: number<br code="" }><=""/></code>	GC one project.

Dependencies

- `PrismaService`
- `DocVersionService`

Where it refuses work

- DocsTrueupService stops the work with an early return when `process.env.VERSION_HEAL_DISABLED === 'true' .`
- DocsTrueupService stops the work with an early return when `process.env.TRUEUP_GC_ENABLED !== 'true' .`
- DocsTrueupService stops the work with an early return when `badged.length === 0 .`

When something fails

- DocsTrueupService handles failure in 2 places: it logs it and continues in all 2.

Diagram

```
sequenceDiagram
    participant Scheduler
    participant Trueup as DocsTrueupService
    participant DB as Database
    participant Snapshot as Live Tech Snapshot

    Scheduler->>Trueup: scheduledSweep()
    Trueup->>Trueup: Check TRUEUP_GC_ENABLED

    alt GC disabled
        Trueup-->>Scheduler: Skip sweep
    else GC enabled
        Trueup->>Trueup: garbageCollect()
        Trueup->>DB: Find documents with removal badges

        loop Each badged document
            alt Techdocs entity exists in live snapshot
                Trueup->>Snapshot: Verify entity is active
                Snapshot-->>Trueup: Entity found
                Trueup->>DB: Clear techEntityRemoved and removedAt
                Trueup->>Trueup: Increment cleared count
            else Badge is younger than grace period
                Trueup->>Trueup: Increment skipped count
            else Badge exceeds grace period
                Trueup->>DB: Delete document
                Note over DB: Owned comments, revisions, versions,<br/>sessions, and symbols cascade
                Trueup->>Trueup: Increment deleted count
            end
        end

        Trueup-->>Scheduler: { deleted, cleared, skipped }
    end
```

Usage

```
import { DocsTrueupService } from './technical-docs/docs-trueup.service';

@Injectable()
export class MaintenanceService {
  constructor(private readonly docsTrueupService: DocsTrueupService) {}

  async runDocumentationCleanup() {
    // Typically invoked by the scheduled job. GC must be enabled through:
    // TRUEUP_GC_ENABLED=true
    const result = await this.docsTrueupService.garbageCollect();

    console.log(
      `Docs true-up complete: ${result.deleted} deleted, ` +
      `${result.cleared} restored, ${result.skipped} retained.`
    );

    return result;
  }

  async repairActiveVersionState() {
    await this.docsTrueupService.healActiveVersions();
  }
}
```

AI Coding Instructions

- Keep removal badges non-destructive in ingestion pipelines: set `metadata.screenRemoved` or `metadata.techEntityRemoved` with `removedAt` ; clear the badge when the source reappears.
- Respect the `TRUEUP_GC_ENABLED` gate and `TRUEUP_GC_GRACE_DAYS` configuration; do not make cleanup destructive by default.
- Process documents independently with per-item error handling so a failed delete cannot abort the remaining sweep.
- Before deleting techdocs pages, verify whether the entity exists in the current live snapshot; clear stale badges instead of deleting restored content.
- Preserve document-level deletion behavior so ORM/database cascade rules remove owned child records consistently.

Relationships

- `DEPENDS_ON` → `PrismaService`
- `DEPENDS_ON` → `DocVersionService`

Referenced By

- `TechnicalDocsController` (DEPENDS_ON)
- `TechnicalDocsModule` (MODULE_PROVIDES)