

GitSyncIngestService

September 4, 2026

GitSyncIngestService

Kind: Service

Source: `atloria-monorepo/apps/api/src/git-sync/git-sync-ingest.service.ts`

GitSyncIngestService — B1's INBOUND lane: a customer `git push` IS a change request.

Flow (deliberately PVC-free — webhook payload + GitHub API only, so it runs on any api

pod): App webhook push → triple echo guard → compare-API diff (payload-commits fallback)

→ blob fetch at the pushed head → path → document mapping via the MIRRORED identity

manifests (`<prefix>/atloria/pages.json` ; renames resolve through the previousSlugs ledger in `slug-registry.json` — never delete+add) → open a `kind:'git'` CR that lands through the existing S2 three-way-merge + CAS writeback.

`isManualOverride /source=USER`

marking happens automatically via the CR merge path.

Mapping rules per changed file under `<prefix>/` (only `*.md` ; `<prefix>/atloria/` is machine-owned and ignored):

- path in `pages.json` → content update against that docUuid
- renamed (compare `previous_filename`) → update against the ORIGINAL docUuid
- old slug in previousSlugs ledger → update against the ledger's docUuid
- unmapped, valid docs///.md → CR `createPath` (new page authored in git)
- removed → CR delete entry → merge UNPUBLISHES (never hard-deletes)

ECHO LOOP = the #1 failure mode; the triple guard:

- (a) head commit message carries the `Atloria-Source-Commit:` trailer,
- (b) the push sender/author is the App's bot login,
- (c) the pushed head equals lastMirroredSha (our own mirror commit).

All three skip WITHOUT opening a CR and advance lastIngestedSha so the next real push diffs from the right base.

`GitSyncIngestService` handles Atloria's inbound Git synchronization lane: a customer `git push` becomes a `kind: 'git'` change request rather than writing directly to documents. It reads webhook and GitHub compare/blob data, maps changed Markdown files through mirrored identity manifests, and relies on the existing S2 merge/CAS pipeline to apply updates, creates, and unpublish operations safely.

The service is deliberately PVC-free and can run on any API pod. It prevents mirror feedback loops with a triple echo guard and advances ingestion state even when an incoming push is identified as Atloria's own mirror commit.

Methods

Method	Signature	Returns	Description
<code>onModuleInit</code>	<code>onModuleInit()</code>	<code>void</code>	
<code>handlePush</code>	<code>handlePush(payload: any)</code>	<code>Promise<IngestOutcome[]></code>	Entry point for one webhook push delivery.

Dependencies

- `ConfigService`
- `PrismaService`
- `GithubSyncAdapter`
- `ChangeRequestService`
- `GitSyncTrigger`

Where it refuses work

- `GitSyncIngestService` stops the work with an early return when `!repoFullName || !branch || !after || ZERO_SHA.test(after) || payload?.deleted === true`.
- `GitSyncIngestService` stops the work with an early return when `rows.length === 0`.
- `GitSyncIngestService` stops the work with an early return when `!sub`.
- `GitSyncIngestService` stops the work with an early return when `direct`.
- `GitSyncIngestService` stops the work with an early return when `!parts`.

- `GitSyncIngestService` stops the work with an early return when `entry.slug === parts.slug || entry.previousSlugs.includes(parts.slug)` .

When something fails

- `GitSyncIngestService` handles failure in 2 places: it logs it and continues in 1, and turns it into a return value in 1.

Diagram

```
sequenceDiagram
    participant GH as GitHub App Webhook
    participant Ingest as GitSyncIngestService
    participant API as GitHub Compare/Blob API
    participant Manifest as Mirrored .atloria manifests
    participant CR as Change Request Pipeline
    participant Merge as S2 Merge + CAS Writeback

    GH->>Ingest: push webhook payload
    Ingest->>Ingest: Check triple echo guard

    alt Atloria mirror echo
        Note over Ingest: Trailer + bot sender + lastMirroredSha all match
        Ingest->>Ingest: Advance lastIngestedSha
        Ingest-->>GH: Skip; no CR opened
    else Customer-authored push
        Ingest->>API: Compare lastIngestedSha...pushedHead
        alt Compare API unavailable/incomplete
            Ingest->>Ingest: Fall back to payload commits
        end
    end

    Ingest->>API: Fetch changed Markdown blobs at pushed head
    Ingest->>Manifest: Read pages.json and slug-registry.json
    Ingest->>Ingest: Map paths, renames, previousSlugs, creates, deletes

    loop Each mapped change
        Ingest->>CR: Open kind: "git" change request
    end

    CR->>Merge: Three-way merge and CAS writeback
    Merge->>Merge: Mark source=USER / isManualOverride
    Ingest->>Ingest: Advance lastIngestedSha
end
```

Usage

```
import { GitSyncIngestService } from './git-sync-ingest.service';
import type { PushEvent } from '@octokit/webhooks-types';

@Injectable()
export class GitHubWebhookController {
  constructor(
    private readonly gitSyncIngestService: GitSyncIngestService,
  ) {}

  async handlePushWebhook(payload: PushEvent) {
    // The service uses the webhook payload plus GitHub APIs only;
    // no shared filesystem or PVC is required.
    const outcomes = await this.gitSyncIngestService.handlePush(payload);

    return {
      accepted: true,
      outcomes,
    };
  }
}
```

AI Coding Instructions

- Preserve the PVC-free design: use webhook payloads and GitHub APIs for diffs and blobs; do not add local checkout or shared-volume dependencies.
- Keep the triple echo guard atomic in behavior: skip only when the source trailer, App bot identity, and `lastMirroredSha` all match; skipped pushes must still advance `lastIngestedSha`.
- Treat `.atloria/` as machine-owned metadata and ignore it as document content; only ingest Markdown files beneath the configured content prefix.
- Resolve renames through `previous_filename` and `previousSlugs` before treating a file as a new document—renames must remain updates to the original `docUuid`, never delete-plus-add.
- Do not bypass the `kind: 'git'` change-request flow: deletes must unpublish through the S2 merge pipeline, and user-source/manual-override metadata is applied by that existing merge path.

Relationships

- `DEPENDS_ON` → `configservice`
- `DEPENDS_ON` → `PrismaService`
- `DEPENDS_ON` → `GithubSyncAdapter`

- `DEPENDS_ON` → `ChangeRequestService`
- `DEPENDS_ON` → `GitSyncTrigger`

Referenced By

- `GitSyncModule` (`MODULE_PROVIDES`)