ChangelogPublicService

September 4, 2026

ChangelogPublicService

Kind: Service

Source: `atloria-monorepo/apps/api/src/changelog/changelog-public.service.ts`

Published-plane reads for the B5 changelog: entry list, RSS 2.0 and JSON Feed 1.1. All of it sits behind the same OptionalJwtAuthGuard → ReaderAuthGuard → DocsAccessGuard stack as every other :slugWithId surface, so a gated project's changelog/feeds 401 exactly like llms.txt does.

`ChangelogPublicService` provides published changelog data for public project surfaces, including structured entry lists, RSS 2.0 feeds, and JSON Feed 1.1 documents. It resolves the published project context and returns only public-facing entries after the route-level `OptionalJwtAuthGuard → ReaderAuthGuard → DocsAccessGuard` access stack has authorized the request.

Methods

Method	Signature	Returns	Description
`resolvePublishedProject`	`resolvePublishedProject(slugWithId: string)`	`Promise<{ projectId: string; organizationId: string; name: string }>`	Same base62-suffix resolution as the other public planes (sdk precedent).
`publishedEntries`	`publishedEntries(projectId: string, limit: unknown)`	`Promise<PublicChangelogEntry[]>`	
`getChangelog`	`getChangelog(slugWithId: string, limit: number)`	`unknown`	
`rssFeed`	`rssFeed(slugWithId: string)`	`Promise<string>`	RSS 2.0 — served with Content-Type application/rss+xml.
`jsonFeed`	`jsonFeed(slugWithId: string)`	`unknown`	JSON Feed 1.1 — served with Content-Type application/feed+json.

Dependencies

- `PrismaService`

Where it refuses work

- `ChangelogPublicService` stops the work with `BadRequestException` when `!match` — “Invalid URL format”.

- `ChangelogPublicService` stops the work with `NotFoundException` when `!project || !project.isPublic` — “Project not found”.

Diagram

```
sequenceDiagram
  participant Client
  participant Controller
  participant Guards as Auth Guards
  participant Service as ChangelogPublicService
  participant Data as Changelog Data Store

  Client->>Controller: GET /:slugWithId/changelog or feed endpoint
  Controller->>Guards: OptionalJwtAuthGuard
  Guards->>Guards: ReaderAuthGuard
  Guards->>Guards: DocsAccessGuard

  alt Access denied
    Guards-->>Client: 401 Unauthorized
  else Access granted
    Guards->>Service: getChangelog() / rssFeed() / jsonFeed()
    Service->>Service: resolvePublishedProject()
    Service->>Data: Load published changelog entries
    Data-->>Service: Published entries
    Service-->>Controller: Changelog data or serialized feed
    Controller-->>Client: JSON, RSS XML, or changelog response
  end
```

Usage

```
import { Controller, Get, Param } from '@nestjs/common';
import { ChangelogPublicService } from './changelog-public.service';

@Controller(':slugWithId/changelog')
export class ChangelogPublicController {
  constructor(
    private readonly changelogPublicService: ChangelogPublicService,
  ) {}

  @Get()
  getChangelog() {
    return this.changelogPublicService.getChangelog();
  }

  @Get('rss.xml')
  async getRssFeed(): Promise<string> {
    return this.changelogPublicService.rssFeed();
  }

  @Get('feed.json')
  getJsonFeed() {
    return this.changelogPublicService.jsonFeed();
  }

  @Get('entries')
  async getEntries() {
    return this.changelogPublicService.publishedEntries();
  }
}
```

AI Coding Instructions

- Keep all public changelog queries scoped through `resolvePublishedProject()` so entries cannot leak across projects or organizations.
- Return only published/public entries from `publishedEntries()`; do not reuse admin or draft-oriented changelog queries for public feeds.
- Preserve the route guard chain— `OptionalJwtAuthGuard`, `ReaderAuthGuard`, and `DocsAccessGuard`—for every `:slugWithId` changelog and feed endpoint.
- Use `rssFeed()` for serialized RSS XML responses and `jsonFeed()` for JSON Feed 1.1 payloads; set the corresponding controller content type when needed.
- Ensure gated projects receive the same unauthorized behavior as other public documentation surfaces such as `llms.txt`.

Relationships

- DEPENDS_ON → `PrismaService`

Referenced By

- `ChangelogModule` (MODULE_PROVIDES)
- `PublicChangelogController` (DEPENDS_ON)