

ChangelogService

September 4, 2026

ChangelogService

Kind: Service**Source:** [atloria-monorepo/apps/api/src/doc-version/changelog.service.ts](https://github.com/atloria-monorepo/apps/api/src/doc-version/changelog.service.ts)

`ChangelogService` generates a changelog document for the current API or documentation version and can persist the generated output for later retrieval. It centralizes changelog creation and storage so versioning workflows can generate consistent release notes without duplicating logic across controllers or jobs.

Methods

Method	Signature	Returns	Description
<code>generateChangelog</code>	<code>generateChangelog(versionAId: string, versionBId: string)</code>	<code>Promise<string></code>	Generate markdown changelog between two versions.
<code>generateAndStore</code>	<code>generateAndStore(versionId: string)</code>	<code>Promise<string></code>	Generate and store changelog on a DocVersion.

Dependencies

- `VersionDiffService`
- `PrismaService`

Diagram

```
sequenceDiagram
    participant Caller
    participant ChangelogService
```

```

participant VersionData as Version/Documentation Data
participant Storage as Persistence Layer

Caller->>ChangelogService: generateChangelog()
ChangelogService->>VersionData: Collect version changes
VersionData-->>ChangelogService: Change metadata
ChangelogService-->>Caller: Promise<string> (generated changelog)

Caller->>ChangelogService: generateAndStore()
ChangelogService->>ChangelogService: generateChangelog()
ChangelogService->>Storage: Save generated changelog
Storage-->>ChangelogService: Stored result
ChangelogService-->>Caller: Promise<string> (generated changelog)

```

Usage

```

import { Injectable } from '@nestjs/common';
import { ChangelogService } from '../changelog.service';

@Injectable()
export class DocumentationReleaseService {
  constructor(
    private readonly changelogService: ChangelogService,
  ) {}

  async publishVersionChangelog(): Promise<string> {
    // Generate the changelog without persisting it.
    const preview = await this.changelogService.generateChangelog();
    console.log(preview);

    // Generate and persist the changelog for the version release.
    return this.changelogService.generateAndStore();
  }
}

```

AI Coding Instructions

- Prefer `generateChangelog()` when callers need a preview or custom handling before persistence.
- Use `generateAndStore()` for release workflows that require the generated changelog to be saved automatically.
- Keep changelog formatting and version-diff logic inside this service rather than duplicating it in controllers or jobs.
- Ensure persistence failures from `generateAndStore()` are surfaced or handled by the calling workflow.

- When changing changelog content, preserve deterministic output so repeated generation for the same version produces consistent results.

Relationships

- `DEPENDS_ON` → `VersionDiffService`
- `DEPENDS_ON` → `PrismaService`

Referenced By

- `DocVersionController` (`DEPENDS_ON`)
- `DocVersionModule` (`MODULE_PROVIDES`)
- `DocVersionModule` (`MODULE_EXPORTS`)