

VersionDiffService

September 4, 2026

VersionDiffService

Kind: Service

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

`VersionDiffService` compares two document versions and produces a structured `VersionDiff` describing the detected changes. It is a NestJS backend service responsible for calculating document-level differences and summary statistics that can be consumed by version-history, review, or audit features.

Methods

Method	Signature	Returns	Description
<code>compareVersions</code>	<code>compareVersions(versionAId: string, versionBId: string)</code>	<code>Promise<VersionDiff></code>	Compare two doc versions at the doc-set level.
<code>diffDocument</code>	<code>diffDocument(docIdA: string, docIdB: string)</code>	<code>unknown</code>	Get raw content for per-document diff (frontend renders the diff).
<code>computeDiffStats</code>	<code>computeDiffStats(contentA: string, contentB: string)</code>	<code>DiffStats</code>	Compute diff statistics without full content.

Dependencies

- `PrismaService`

Diagram

```
sequenceDiagram
    participant Client as API Consumer
    participant Service as VersionDiffService
    participant Old as Previous Version
    participant New as Current Version

    Client->>Service: compareVersions(oldVersion, newVersion)
    Service->>Old: Read document content
    Service->>New: Read document content
    Service->>Service: diffDocument(oldContent, newContent)
    Service->>Service: computeDiffStats(diff)
    Service-->>Client: Promise<VersionDiff>
```

Usage

```
import { VersionDiffService } from './doc-version/version-diff.service';

async function compareDocumentVersions(
  versionDiffService: VersionDiffService,
  previousVersion: DocumentVersion,
  currentVersion: DocumentVersion,
) {
  const diff = await versionDiffService.compareVersions(
    previousVersion,
    currentVersion,
  );

  console.log('Changes:', diff);
  console.log('Diff statistics:', diff.stats);

  return diff;
}
```

AI Coding Instructions

- Use `compareVersions()` as the public entry point; keep document comparison and statistics logic encapsulated in `diffDocument()` and `computeDiffStats()`.
- Preserve the expected `VersionDiff` and `DiffStats` shapes when extending diff output, since API consumers may depend on their fields.
- Handle missing, empty, or structurally different document content before running the document diff to avoid invalid comparisons.

- Keep diff calculations deterministic: identical version content should always produce the same empty or zero-change result.
- Update downstream version-history, review, or audit integrations when adding new change categories or statistics.

Relationships

- `DEPENDS_ON` → `PrismaService`

Referenced By

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