

DocumentRevisionService

September 4, 2026

DocumentRevisionService

Kind: Service

Source: [atloria-monorepo/apps/api/src/document/services/document-revision.service.ts](#)

`DocumentRevisionService` manages historical snapshots of documents in the backend. It creates, retrieves, restores, counts, and removes revisions, enabling version history and recovery workflows for document updates.

Methods

Method	Signature	Returns	Description
<code>createRevision</code>	<code>createRevision(documentId: string, snapshot: RevisionSnapshot, userId: string, changeType: ChangeType)</code>	<code>Promise<DocumentRevision></code>	Create a revision snapshot of a document
<code>getRevisions</code>	<code>getRevisions(documentId: string, limit: number, offset: number)</code>	<code>Promise<DocumentRevision[]></code>	Get all revisions for a document with pagination
<code>getRevision</code>	<code>getRevision(revisionId: string)</code>	<code>Promise<DocumentRevision></code>	Get a specific revision by ID
<code>restoreFromRevision</code>	<code>restoreFromRevision(documentId: string, revisionId: string, userId: string)</code>	<code>Promise<any></code>	Restore a document from a specific revision
<code>getRevisionCount</code>	<code>getRevisionCount(documentId: string)</code>	<code>Promise<number></code>	Get count of revisions for a document

Method	Signature	Returns	Description
<code>cleanupOldRevisions</code>	<code>cleanupOldRevisions(documentId: string, keepLast: number)</code>	<code>Promise<number></code>	Cleanup old revisions beyond retention limit
<code>deleteAllRevisions</code>	<code>deleteAllRevisions(documentId: string)</code>	<code>Promise<number></code>	Delete all revisions for a document (use with caution)

Dependencies

- `PrismaService`

Where it refuses work

- `DocumentRevisionService` stops the work with `NotFoundException` when `!revision` — “Revision not found”.
- `DocumentRevisionService` stops the work with `BadRequestException` when `revision.documentId !== documentId` — “Revision does not belong to this document”.
- `DocumentRevisionService` stops the work with `NotFoundException` when `!document` — “Document not found”.
- `DocumentRevisionService` stops the work with an early return when `allRevisions.length <= keepLast` .

Diagram



```
Client->>RevisionService: restoreFromRevision(revisionId)
RevisionService->>Database: Load revision and update document
Database-->>RevisionService: Restored document
RevisionService-->>Client: Restored result
```

Usage

```
import { Injectable } from '@nestjs/common';
import { DocumentRevisionService } from './document-revision.service';

@Injectable()
export class DocumentHistoryController {
  constructor(
    private readonly documentRevisionService: DocumentRevisionService,
  ) {}

  async getHistory(documentId: string) {
    return this.documentRevisionService.getRevisions(documentId);
  }

  async restore(documentId: string, revisionId: string) {
    return this.documentRevisionService.restoreFromRevision(
      documentId,
      revisionId,
    );
  }

  async pruneHistory(documentId: string) {
    const deletedCount =
      await this.documentRevisionService.cleanupOldRevisions(documentId);

    return { deletedCount };
  }
}
```

AI Coding Instructions

- Create a revision before applying destructive or meaningful document updates so the previous document state remains recoverable.
- Scope revision queries, counts, cleanup, and deletion operations to the correct document identifier; never delete revisions globally unless explicitly intended.
- Preserve document ownership and authorization checks in the calling controller or service before exposing revision history or restore operations.
- Use `restoreFromRevision()` as the canonical restoration path rather than manually copying revision fields into a document.

- Run `cleanupOldRevisions()` according to the configured retention policy and verify that the newest required revisions are retained.

Relationships

- `DEPENDS_ON` → `PrismaService`

Referenced By

- `DocumentController` (`DEPENDS_ON`)
- `DocumentModule` (`MODULE_PROVIDES`)
- `DocumentService` (`DEPENDS_ON`)