VersionCarryForwardService

September 4, 2026

VersionCarryForwardService

Kind: Service

Source: `atloria-monorepo/apps/api/src/doc-version/carry-forward.service.ts`

`VersionCarryForwardService` carries reusable content from one document version into another during versioning workflows. It resolves carry-forward flags, identifies placeholder references, and transfers associated screenshots while reporting what was successfully carried or missing.

Methods

Method	Signature	Returns	Description
`carryForward`	`carryForward(newVersionId: string)`	`Promise<CarryForwardResult>`	Carry forward documents from previousVersion → newVersion.
`resolveFlag`	`resolveFlag(previousDocId: string, newVersionId: string, action: 'keep_previous' | 'use_new' | 'discard')`		
`extractPlaceholders`	`extractPlaceholders(docs: Array<{ id: string; content: string }>)`	`Array<{ text: string; routePath: string; documentId: string }>`	Phase 4: Extract screenshot placeholders from document content.
`carryForwardScreenshots`	`carryForwardScreenshots(previousVersionId: string, newVersionId: string)`	`Promise<{ carried: number; missing: number }>`	Phase 4: Carry forward screenshots from previous version to new version.

Dependencies

- `PrismaService`
- `DocVersionService`

Where it refuses work

- `VersionCarryForwardService` stops the work with an early return when `!newVersion.previousVersionId`.
- `VersionCarryForwardService` stops the work with an early return when `!oldCategoryId`.
- `VersionCarryForwardService` stops the work with an early return when `!oldCategory`.

- `VersionCarryForwardService` stops the work with an early return when `newCategory`.
- `VersionCarryForwardService` stops the work with an early return when `!previousDoc`.
- `VersionCarryForwardService` stops the work with an early return when `placeholders.length === 0`.

When something fails

- `VersionCarryForwardService` handles failure in 2 places: it logs it and continues in 1, and discards it silently in 1. A failure discarded silently leaves no trace for whoever debugs this later.

Diagram

```
sequenceDiagram
  participant Caller as Version Workflow
  participant Service as VersionCarryForwardService
  participant Docs as Document Version Store
  participant Assets as Screenshot Storage

  Caller->>Service: carryForward()
  Service->>Service: resolveFlag()
  Service->>Docs: Load source and target versions
  Service->>Service: extractPlaceholders()
  Service->>Docs: Copy eligible placeholder content
  Service->>Service: carryForwardScreenshots()
  Service->>Assets: Copy referenced screenshots
  Assets-->>Service: carried / missing counts
  Service-->>Caller: CarryForwardResult
```

Usage

```
import { VersionCarryForwardService } from './carry-forward.service';

// Typically injected by NestJS into a workflow service or controller.
@Injectable()
export class DocumentVersionWorkflowService {
  constructor(
    private readonly carryForwardService: VersionCarryForwardService,
  ) {}

  async createVersion(): Promise<void> {
    const result = await this.carryForwardService.carryForward();

    console.log('Carry-forward complete:', result);
  }
}
```

AI Coding Instructions

- Treat `carryForward()` as the orchestration entry point; keep new carry-forward steps coordinated through it rather than calling internal helpers independently.
- Run `resolveFlag()` before copying content so document-version state is normalized before placeholder and asset processing.
- Preserve the `{ text, routePath, documentId }` shape returned by `extractPlaceholders()` when extending placeholder detection or routing logic.

- Handle missing screenshots as an expected outcome from `carryForwardScreenshots()`; report them through the result instead of failing the entire carry-forward operation.
- Keep document persistence and screenshot-storage integration concerns separated so carry-forward behavior remains testable.

Relationships

- DEPENDS_ON → `PrismaService`
- DEPENDS_ON → `DocVersionService`

Referenced By

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