

DocumentationService

September 4, 2026

DocumentationService

Kind: Service

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

`DocumentationService` coordinates the generation of user-facing documentation for the API. Its `generateUserDocumentation()` method runs the documentation generation workflow and returns either generated documentation content or a synchronization batch response, depending on the configured generation path.

Methods

Method	Signature	Returns	Description
<code>generateUserDocumentation</code>	<code>generateUserDocumentation(dto: GenerateUserDocsDto, user: JwpPayload)</code>	<code>Promise<GeneratedDocumentation</code>	<code>SyncBatchResponseDto></code>

Dependencies

- `WorkflowDetectorService`
- `UserDocsGeneratorService`
- `AzureClaudeProvider`

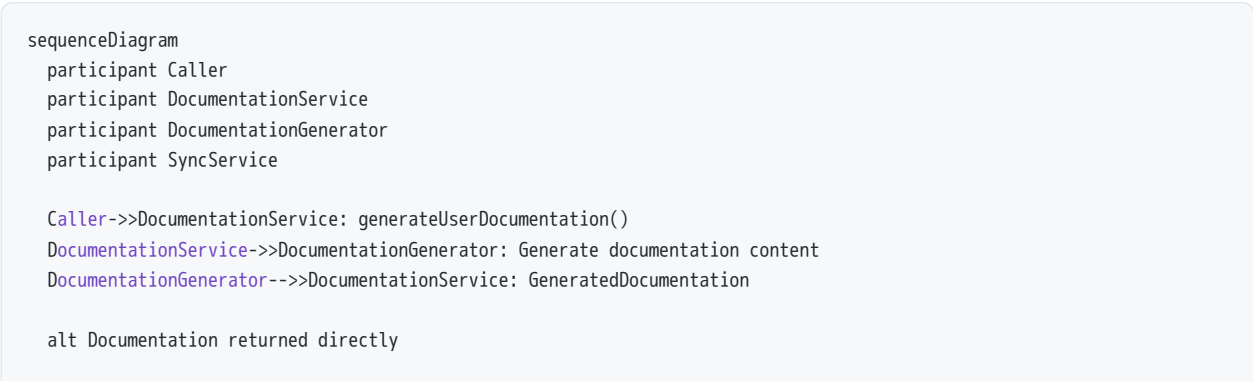
Where it refuses work

- `DocumentationService` stops the work with an early return when `dto.saveIncremental !== false && dto.projectId && user`.

When something fails

- `DocumentationService` handles failure in 2 places: it logs it and continues in all 2.

Diagram



```

DocumentationService-->>Caller: GeneratedDocumentation
else Documentation requires synchronization
DocumentationService-->>SyncService: Sync generated documentation
SyncService-->>DocumentationService: SyncBatchResponseDto
DocumentationService-->>Caller: SyncBatchResponseDto
end

```

Usage

```

import { Injectable } from '@nestjs/common';
import { DocumentationService } from '../documentation/documentation.service';

@Injectable()
export class DocumentationController {
  constructor(
    private readonly documentationService: DocumentationService,
  ) {}

  async generate() {
    const result = await this.documentationService.generateUserDocumentation();

    return result;
  }
}

```

AI Coding Instructions

- Use NestJS dependency injection to access `DocumentationService` ; do not instantiate it directly with `new` .
- Treat the return value as a union of `GeneratedDocumentation | SyncBatchResponseDto` and narrow it before consuming type-specific fields.
- Keep documentation generation orchestration inside this service rather than duplicating generation or synchronization logic in controllers.
- Preserve asynchronous error handling when calling `generateUserDocumentation()` , especially when exposing it through HTTP endpoints or background jobs.
- Update downstream consumers if the generated documentation schema or synchronization response contract changes.

Relationships

- `DEPENDS_ON` → `WorkflowDetectorService`
- `DEPENDS_ON` → `UserDocsGeneratorService`
- `DEPENDS_ON` → `AzureClaudeProvider`

Referenced By

- `DocumentationController` (`DEPENDS_ON`)
- `DocumentationModule` (`MODULE_PROVIDES`)
- `DocumentationModule` (`MODULE_EXPORTS`)