

DocAutomationSchedulerService

September 4, 2026

DocAutomationSchedulerService

Kind: Service

Source: `atloria-monorepo/apps/api/src/documentation/services/doc-automation-scheduler.service.ts`

Phase 5: Cron-based scheduled documentation regeneration.

Uses setInterval-based scheduling (no external cron dependency needed).

`DocAutomationSchedulerService` manages scheduled documentation regeneration for the API using in-process `setInterval` timers rather than an external cron dependency. It coordinates recurring automation runs, triggers the documentation generation workflow, and ensures scheduled jobs are started and cleaned up with the NestJS application lifecycle.

Methods

Method	Signature	Returns
<code>onModuleInit</code>	<code>onModuleInit()</code>	<code>unknown</code>
<code>onModuleDestroy</code>	<code>onModuleDestroy()</code>	<code>unknown</code>

Dependencies

- `PrismaService`

Where it refuses work

- `DocAutomationSchedulerService` stops the work with an early return when `parts.length !== 5`.
- `DocAutomationSchedulerService` stops the work with an early return when `field === '*'`.

- `DocAutomationSchedulerService` stops the work with an early return when `field.includes(',')`.

When something fails

- `DocAutomationSchedulerService` handles failure in 1 place: it logs it and continues in all 1.

Diagram

```
sequenceDiagram
    participant App as NestJS Application
    participant Scheduler as DocAutomationSchedulerService
    participant Timer as setInterval Timer
    participant Generator as Documentation Generator
    participant Storage as Documentation Storage

    App->>Scheduler: onModuleInit()
    Scheduler->>Timer: register scheduled interval

    loop At configured interval
        Timer->>Scheduler: execute scheduled job
        Scheduler->>Generator: regenerate documentation
        Generator->>Storage: persist generated docs
        Storage-->>Generator: saved
        Generator-->>Scheduler: generation result
    end

    App->>Scheduler: onModuleDestroy()
    Scheduler->>Timer: clearInterval()
```

Usage

```
import { Injectable, OnModuleInit } from '@nestjs/common';
import { DocAutomationSchedulerService } from '../documentation/services/doc-automation-schedule

@Injectable()
export class AppBootstrapService implements OnModuleInit {
  constructor(
    private readonly docScheduler: DocAutomationSchedulerService,
  ) {}

  onModuleInit(): void {
    // Start the configured recurring documentation regeneration job.
    this.docScheduler.start();
  }
}
```

```
}  
}
```

AI Coding Instructions

- Keep scheduling in-process with `setInterval` ; do not introduce an external cron dependency unless the deployment architecture requires distributed scheduling.
- Ensure every created interval is cleared during module/application shutdown to prevent dangling timers and test-process hangs.
- Guard against overlapping regeneration runs when an execution takes longer than the configured interval.
- Route generation through the existing documentation automation/generation service rather than duplicating generation or persistence logic in the scheduler.
- Make schedule intervals configurable through application configuration, and use shorter intervals or mocked timers in tests.

Relationships

- `DEPENDS_ON` → `PrismaService`

Referenced By

- `DocumentationModule` (`MODULE_PROVIDES`)