

DocsConfigService

September 4, 2026

DocsConfigService

Kind: Service

Source: `atloria-monorepo/apps/api/src/change-request/docs-config.service.ts`

DocsConfigService — exposes the per-project docs review gate (ProjectDocsRepo.requireReview, the S2 author-time protection toggle). A project with an EXPLICIT config row uses its stored value; a project with NO row falls back to the platform default, which is governed by the `DOCS_REVIEW_DEFAULT` env flag. That makes widening the S2 rollout a reversible flag flip (`DOCS_REVIEW_DEFAULT=true` → every un-configured project protects published public docs via propose → CR; the per-project toggle remains the opt-out). Reads never write.

DocsConfigService resolves and updates the per-project documentation review gate (`requireReview`). Projects with an explicit configuration use their stored value; projects without a row fall back to the platform-wide `DOCS_REVIEW_DEFAULT` environment flag. Read operations are side-effect free, while updates persist the project-specific opt-in or opt-out setting.

Methods

Method	Signature	Returns	Description
<code>get</code>	<code>get(projectId: string)</code>	<code>Promise<DocsConfig></code>	Read the review gate.
<code>set</code>	<code>set(projectId: string, requireReview: boolean)</code>	<code>Promise<DocsConfig></code>	Set the review gate, creating the singleton config row when absent.

Dependencies

- PrismaService
- ConfigService

Diagram

```
sequenceDiagram
    participant Caller
    participant DocsConfigService
    participant ProjectDocsRepo
    participant Env as DOCS_REVIEW_DEFAULT

    Caller->>DocsConfigService: get(projectId)
    DocsConfigService->>ProjectDocsRepo: find config row(projectId)

    alt Explicit config exists
        ProjectDocsRepo-->>DocsConfigService: requireReview value
        DocsConfigService-->>Caller: stored DocsConfig
    else No config row
        DocsConfigService->>Env: read default flag
        Env-->>DocsConfigService: default requireReview value
        DocsConfigService-->>Caller: fallback DocsConfig
    end

    Caller->>DocsConfigService: set(projectId, requireReview)
    DocsConfigService->>ProjectDocsRepo: create or update config row
    ProjectDocsRepo-->>DocsConfigService: persisted DocsConfig
    DocsConfigService-->>Caller: persisted DocsConfig
```

Usage

```
import { Injectable } from '@nestjs/common';
import { DocsConfigService } from './docs-config.service';

@Injectable()
export class ProjectSettingsService {
    constructor(private readonly docsConfigService: DocsConfigService) {}

    async getDocsReviewSettings(projectId: string) {
        // Returns the explicit project setting, or DOCS_REVIEW_DEFAULT when absent.
        return this.docsConfigService.get(projectId);
    }

    async disableDocsReviewForProject(projectId: string) {
        // Persists an explicit per-project opt-out.
        return this.docsConfigService.set(projectId, {
            requireReview: false,
        });
    }
}
```

```
}  
}
```

AI Coding Instructions

- Treat `get()` as read-only: resolving a fallback from `DOCS_REVIEW_DEFAULT` must not create or update a configuration row.
- Preserve the distinction between an explicit `requireReview` value and a missing config row; missing rows must use the platform default.
- Use `set()` for intentional per-project overrides, including explicit opt-outs when the global default enables review protection.
- Keep `DOCS_REVIEW_DEFAULT` parsing and fallback behavior centralized in this service rather than duplicating it in callers.
- When integrating document publishing flows, use the resolved `requireReview` value to decide whether public-document changes must go through propose → change request.

Relationships

- `DEPENDS_ON` → `PrismaService`
- `DEPENDS_ON` → `configservice`

Referenced By

- `ChangeRequestModule` (`MODULE_PROVIDES`)
- `DocsConfigController` (`DEPENDS_ON`)