GitSyncService

September 4, 2026

GitSyncService

Kind: Service

Source: `atloria-monorepo/apps/api/src/git-sync/git-sync.service.ts`

GitSyncService — settings CRUD for ProjectDocsSync plus the sync-now kick. DB + queue only
(api pods can't reach the PVC); the mirror itself runs as a 'docs-sync' worker job.

`GitSyncService` manages CRUD operations for `ProjectDocsSync` settings and provides a `mirrorNow()` command to trigger documentation synchronization. It persists configuration in the database and enqueues sync work for the `docs-sync` worker, since API pods do not have access to the PVC containing the mirror.

Methods

Method	Signature	Returns	Description
`get`	`get(projectId: string)`	`Promise<GitSyncView>`	
`update`	`update(projectId: string, dto: UpdateGitSyncDto, userId: string)`	`Promise<GitSyncView>`	Upsert the sync config.
`mirrorNow`	`mirrorNow(projectId: string)`	`Promise<{ queued: boolean }>`	Kick a mirror now (worker processes it).

Dependencies

- `PrismaService`
- `DocsSyncQueue`

Where it refuses work

- `GitSyncService` stops the work with `BadRequestException` when `prefix.split('/').some((s) => s === '.atloria' || s === '..' || s === '.')` — “pathPrefix may not contain ".atloria" or dot segments”.
- `GitSyncService` stops the work with `BadRequestException` when `!row || row.direction === 'off'` — “git sync is not enabled for this project”.
- `GitSyncService` stops the work with an early return when `!row`.

Diagram

```
sequenceDiagram
  participant Client
  participant API as API Controller
  participant Service as GitSyncService
  participant DB as Database
  participant Queue as Job Queue
  participant Worker as docs-sync Worker
  participant PVC as Git Mirror PVC

  Client->>API: Update sync settings
  API->>Service: update()
  Service->>DB: Save ProjectDocsSync settings
  DB-->>Service: Updated settings
  Service-->>API: GitSyncView
  API-->>Client: Sync settings

  Client->>API: Trigger sync now
  API->>Service: mirrorNow()
  Service->>Queue: Enqueue docs-sync job
  Queue-->>Service: Job accepted
  Service-->>API: { queued: true }
  API-->>Client: Sync queued

  Worker->>Queue: Consume docs-sync job
  Worker->>PVC: Update git mirror
```

Usage

```
import { Injectable } from '@nestjs/common';
import { GitSyncService } from './git-sync.service';

@Injectable()
export class ProjectDocsController {
  constructor(private readonly gitSyncService: GitSyncService) {}

  async getSyncSettings() {
    return this.gitSyncService.get();
```

```
  }

  async updateSyncSettings() {
    // The concrete update payload depends on the ProjectDocsSync DTO.
    return this.gitSyncService.update();
  }

  async syncNow() {
    const result = await this.gitSyncService.mirrorNow();

    return {
      message: result.queued ? 'Documentation sync queued' : 'Unable to queue sync',
      ...result,
    };
  }
}
```

AI Coding Instructions

- Keep this service limited to database-backed sync settings and job enqueueing; do not perform git mirror operations in API pods.
- Route all filesystem/PVC-dependent mirror work through the `docs-sync` worker job.
- Preserve the `GitSyncView` return shape for both `get()` and `update()` so API consumers receive consistent settings data.
- Ensure `mirrorNow()` reports queue acceptance with `{ queued: boolean }`; do not imply that the mirror completed successfully.
- When adding settings fields, update the persistence model, DTO/view mapping, and worker job payload handling together.

Relationships

- DEPENDS_ON → `PrismaService`
- DEPENDS_ON → `DocsSyncQueue`

Referenced By

- `GitSyncController` (DEPENDS_ON)
- `GitSyncModule` (MODULE_PROVIDES)
- `GitSyncModule` (MODULE_EXPORTS)