

TechDocsConnectorsService

September 4, 2026

TechDocsConnectorsService

Kind: Service**Source:** [atloria-monorepo/apps/api/src/technical-docs/techdocs-connectors.service.ts](https://github.com/atloria-monorepo/apps/api/src/technical-docs/techdocs-connectors.service.ts)

`TechDocsConnectorsService` manages integrations between the platform and external technical documentation providers. It handles connector lifecycle operations, enablement, removal, and both manual and scheduled synchronization of documentation content.

Methods

Method	Signature	Returns	Description
<code>create</code>	<code>create(projectId: string, userId: string, dto: CreateConnectorDto)</code>	unknown	
<code>list</code>	<code>list(projectId: string, userId: string)</code>	unknown	
<code>setEnabled</code>	<code>setEnabled(projectId: string, userId: string, id: string, enabled: boolean)</code>	unknown	
<code>remove</code>	<code>remove(projectId: string, userId: string, id: string)</code>	unknown	
<code>syncNow</code>	<code>syncNow(projectId: string, userId: string, id: string)</code>	unknown	Manual "sync now" from the UI.
<code>syncDue</code>	<code>syncDue()</code>	unknown	Cron: every 10 minutes, sync connectors whose interval has elapsed.
<code>syncOne</code>	<code>syncOne(id: string)</code>	unknown	Sync a single connector: pull the delta by cursor, ingest new items, advance the cursor.

Dependencies

- PrismaService
- TechDocsSourcesService

Where it refuses work

- TechDocsConnectorsService stops the work with NotFoundException when !c — “Connector not found”, in 3 places.
- TechDocsConnectorsService stops the work with NotFoundException when !project — “Project not found”.
- TechDocsConnectorsService stops the work with BadRequestException when !c.subdomain || !c.email || !c.apiToken — “Zendesk connector needs subdomain, email and apiToken.”.
- TechDocsConnectorsService stops the work with BadRequestException when !c.accessToken — “Intercom connector needs an accessToken.”.
- TechDocsConnectorsService stops the work with BadRequestException when type !== 'ZENDESK' && type !== 'INTERCOM' — “type must be zendesk or intercom.”.
- TechDocsConnectorsService stops the work with an early return when !due.length .

When something fails

- TechDocsConnectorsService handles failure in 1 place: it lets it reach the caller in all 1.

Diagram

```
sequenceDiagram
    participant Client
    participant Service as TechDocsConnectorsService
    participant Connector as Connector Repository
    participant Provider as External Docs Provider

    Client->>Service: create(connectorConfig)
    Service->>Connector: persist connector
    Connector-->>Service: connector
    Service-->>Client: created connector

    Client->>Service: syncNow(connectorId)
    Service->>Connector: load enabled connector
    Connector-->>Service: connector
    Service->>Provider: fetch documentation
    Provider-->>Service: documentation data
    Service->>Connector: store sync state/content
```

```

Service-->>Client: sync result

Service->>Service: syncDue()
Service->>Connector: find due enabled connectors
loop Each connector
  Service->>Service: syncOne(connector)
  Service->>Provider: fetch documentation
end

```

Usage

```

import { TechDocsConnectorsService } from './technical-docs/techdocs-connectors.service';

@Injectable()
export class DocumentationAdminService {
  constructor(
    private readonly techDocsConnectorsService: TechDocsConnectorsService,
  ) {}

  async addAndSyncConnector() {
    const connector = await this.techDocsConnectorsService.create({
      name: 'Engineering Docs',
      provider: 'github',
      enabled: true,
      // Include provider-specific repository, authentication, and sync settings.
    });

    await this.techDocsConnectorsService.syncNow(connector.id);

    return connector;
  }

  async disableConnector(connectorId: string) {
    return this.techDocsConnectorsService.setEnabled(connectorId, false);
  }
}

```

AI Coding Instructions

- Use `create()` for new connector configuration and validate provider-specific settings before attempting a sync.
- Prefer `syncNow()` for user-initiated synchronization; use `syncDue()` only from scheduled/background job integration points.
- Keep `syncOne()` focused on synchronizing a single connector and reuse it from both manual and scheduled sync flows.

- Check that a connector is enabled before fetching external documentation, and preserve sync status or errors for observability.
- When removing a connector with `remove()`, ensure related synced content, credentials, and scheduled-sync state are handled consistently.

Relationships

- `DEPENDS_ON` → `PrismaService`
- `DEPENDS_ON` → `TechDocsSourcesService`

Referenced By

- `TechnicalDocsController` (`DEPENDS_ON`)
- `TechnicalDocsModule` (`MODULE_PROVIDES`)