

L10nController

September 4, 2026

L10nController

Kind: Controller

Source: [atloria-monorepo/apps/api/src/l10n/l10n.controller.ts](https://github.com/atloria-monorepo/apps/api/src/l10n/l10n.controller.ts)

B6 author-side localization API. All routes are org-ownership-guarded (fail-closed): project-scoped routes verify the caller's org owns :projectId, document-scoped routes verify it owns :documentId's project.

`L10nController` exposes the author-side localization API for managing localized project and document content. It enforces fail-closed organization ownership checks before processing requests, ensuring callers can access only projects and documents owned by their organization.

Diagram

```
graph LR
  Client[Authenticated author client] --> Controller[L10nController]
  Controller --> Auth[Authentication context]
  Controller --> Guard[Organization ownership guard]
  Guard -->|Project route| ProjectCheck[Verify org owns projectId]
  Guard -->|Document route| DocumentCheck[Verify org owns documentId project]
  ProjectCheck --> Service[Service layer]
  DocumentCheck --> Service
  Service --> Storage[(Project/document localization data)]
```

Usage

```
// Register the localization module in the API application.
// Clients should call the controller through its authenticated HTTP routes;
// do not instantiate L10nController directly.

import { Module } from '@nestjs/common';
import { L10nModule } from './l10n/l10n.module';
```

```

@Module({
  imports: [LocalizationModule],
})
export class AppModule {}

// Example authenticated request pattern from an authoring client.
const response = await fetch(
  `${API_BASE_URL}/api/projects/${projectId}/...`,
  {
    headers: {
      Authorization: `Bearer ${accessToken}`,
      'Content-Type': 'application/json',
    },
  },
);

if (!response.ok) {
  throw new Error(`Localization request failed: ${response.status}`);
}

const localizationData = await response.json();

```

AI Coding Instructions

- Keep all new localization routes protected by the same organization-ownership guard pattern; authorization must fail closed when ownership cannot be verified.
- For project-scoped endpoints, validate ownership using `projectId` ; for document-scoped endpoints, resolve the document's project and validate ownership against that project.
- Keep controllers thin: validate request input, enforce guards, and delegate localization business logic to the service layer.
- Do not trust organization, project, or document identifiers supplied by the client without performing the relevant ownership check.
- Preserve authenticated request context when adding integrations so downstream localization operations remain organization-scoped.

Relationships

- MODULE_DECLARES → `overview`
- MODULE_DECLARES → `pages`
- MODULE_DECLARES → `bulkTranslate`
- MODULE_DECLARES → `getSettings`
- MODULE_DECLARES → `updateSettings`

- MODULE_DECLARES → `getPair`
- MODULE_DECLARES → `upsert`
- MODULE_DECLARES → `publish`
- MODULE_DECLARES → `unpublish`
- MODULE_DECLARES → `remove`
- MODULE_DECLARES → `machineTranslate`
- DEPENDS_ON → `TranslationsService`
- DEPENDS_ON → `TranslateQueueService`
- DEPENDS_ON → `L10nAutomationService`

Referenced By

- `L10nModule` (MODULE_DECLARES)