

SnippetsController

September 4, 2026

SnippetsController

Kind: Controller

Source: `atlوريا-monorepo/apps/api/src/snippets/snippets.controller.ts`

B4 — project-scoped snippet CRUD. Every route rides `:projectId` and is org-verified fail-closed via `ResourceOrgGuard(kind: 'project')`; the service additionally filters every query by `projectId`.

`SnippetsController` exposes project-scoped CRUD endpoints for managing snippets within the API. Every route is nested under `:projectId`, protected by `ResourceOrgGuard(kind: 'project')`, and delegates to the snippets service, which additionally filters all data access by `projectId` for defense in depth.

Diagram

```
graph LR
  Client[API Client] --> Route[/projects/:projectId/snippets]
  Route --> Guard[ResourceOrgGuard<br/>kind: project]
  Guard -->|Organization access verified| Controller[SnippetsController]
  Guard -->|Access denied| Forbidden[403 Forbidden]
  Controller --> Service[SnippetsService]
  Service --> Filter[Filter queries by projectId]
  Filter --> Database[(Database)]
```

Usage

```
// Example HTTP requests for project-scoped snippet CRUD.

const projectId = 'project_123';
const baseUrl = `https://api.example.com/projects/${projectId}/snippets`;

const headers = {
  Authorization: `Bearer ${accessToken}`,
  'Content-Type': 'application/json',
```

```

};

// Create a snippet.
const created = await fetch(baseUrl, {
  method: 'POST',
  headers,
  body: JSON.stringify({
    title: 'Deploy command',
    content: 'pnpm --filter api deploy',
    language: 'bash',
  }),
}).then((response) => response.json());

// List snippets belonging only to this project.
const snippets = await fetch(baseUrl, { headers }).then((response) =>
  response.json(),
);

// Update a project-owned snippet.
await fetch(`${baseUrl}/${created.id}`, {
  method: 'PATCH',
  headers,
  body: JSON.stringify({
    content: 'pnpm --filter @atlolia/api deploy',
  }),
});

// Delete a project-owned snippet.
await fetch(`${baseUrl}/${created.id}`, {
  method: 'DELETE',
  headers,
});

```

AI Coding Instructions

- Keep all snippet endpoints project-scoped; include `:projectId` in controller routes and pass it through to service methods.
- Preserve `ResourceOrgGuard(kind: 'project')` on routes so organization authorization fails closed before controller logic runs.
- Do not query snippets by snippet ID alone; service-layer reads, updates, and deletes must also constrain results by `projectId`.
- Use the existing DTOs and validation conventions when adding snippet fields or endpoints.
- Return authorization-safe not-found behavior for snippets that do not belong to the requested project.

Relationships

- MODULE_DECLARES → list
- MODULE_DECLARES → get
- MODULE_DECLARES → references
- MODULE_DECLARES → create
- MODULE_DECLARES → update
- MODULE_DECLARES → remove
- DEPENDS_ON → SnippetsService

Referenced By

- SnippetsModule (MODULE_DECLARES)