

SdkController

September 4, 2026

SdkController

Kind: Controller

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

Owner surface for C2 multi-language SDKs: per-project language enablement, generation status for the current spec hash, and a manual regenerate.

`SdkController` exposes the owner-facing API surface for managing C2 SDK generation across supported programming languages. It handles per-project language enablement, reports generation status for the current API specification hash, and provides a manual regeneration endpoint when SDK artifacts need to be refreshed.

Diagram

```
graph LR
  Owner[Project Owner] --> Controller[SdkController]
  Controller --> Auth[Owner Authorization]
  Controller --> Config[SDK Language Configuration]
  Controller --> Spec[Current API Spec Hash]
  Controller --> Generator[SDK Generation Service]
  Generator --> Artifacts[Generated Multi-language SDK Artifacts]
  Controller --> Status[Generation Status Response]
```

Usage

```
import { Controller, Get, Post, Param, Body } from '@nestjs/common';

@Controller('projects/:projectId/sdk')
export class SdkController {
  constructor(private readonly sdkService: SdkService) {}

  @Get('status')
  getStatus(@Param('projectId') projectId: string) {
    return this.sdkService.getGenerationStatus(projectId);
  }
}
```

```

}

@Post('languages')
enableLanguage(
  @Param('projectId') projectId: string,
  @Body() body: { language: 'typescript' | 'python' | 'go' },
) {
  return this.sdkService.enableLanguage(projectId, body.language);
}

@Post('regenerate')
regenerate(@Param('projectId') projectId: string) {
  return this.sdkService.regenerateForCurrentSpec(projectId);
}
}

// Example client calls:
// GET /projects/proj_123/sdk/status
// POST /projects/proj_123/sdk/languages { "language": "typescript" }
// POST /projects/proj_123/sdk/regenerate

```

AI Coding Instructions

- Keep all endpoints scoped to a project and enforce owner-level authorization before reading or mutating SDK configuration.
- Use the current specification hash when reporting status or triggering generation; do not treat artifacts generated from older specs as current.
- Delegate language validation, generation orchestration, and artifact persistence to SDK services rather than adding generation logic directly in the controller.
- Return generation state consistently, including enabled languages, current spec hash, artifact hash, and whether regeneration is required.
- Treat manual regeneration as an idempotent operation where possible to avoid duplicate generation jobs for the same project, language, and spec hash.

Relationships

- MODULE_DECLARES → `getStatus`
- MODULE_DECLARES → `setConfig`
- MODULE_DECLARES → `regenerate`
- DEPENDS_ON → `SdkArtifactsService`

Referenced By

- `SdkModule` (`MODULE_DECLARES`)