PublicSdkController

September 4, 2026

PublicSdkController

Kind: Controller

Source: `atloria-monorepo/apps/api/src/sdk/public-sdk.controller.ts`

Published-site plane for C2 SDK downloads (`/api/v1/public/p/:slugWithId/sdk/...`).

Same guard stack as PublicProjectController: OptionalJwtAuthGuard populates req.user (org-member bypass), ReaderAuthGuard decodes the reader token, and DocsAccessGuard enforces the A3 gate on every :slugWithId surface — so a gated project's SDKs are exactly as protected as its docs.

typescript → the hand-rolled flagship client, emitted per-request straight from the source-derived spec (never stored). The six generated languages → 302 to the content-addressed zip in blob storage.

`PublicSdkController` serves SDK downloads for published projects at `/api/v1/public/p/:slugWithId/sdk/...`. It applies the same optional organization-member, reader-token, and documentation access checks as the public documentation surface, ensuring gated SDKs remain protected. The TypeScript SDK is generated on demand from the source-derived spec, while generated SDKs for other languages redirect to content-addressed ZIP artifacts in blob storage.

Diagram

```
graph LR
  Client[SDK consumer] --> Route["/api/v1/public/p/:slugWithId/sdk/:language"]

  Route --> OptionalJwtAuthGuard[OptionalJwtAuthGuard]
  OptionalJwtAuthGuard --> ReaderAuthGuard[ReaderAuthGuard]
  ReaderAuthGuard --> DocsAccessGuard[DocsAccessGuard]

  DocsAccessGuard --> Controller[PublicSdkController]

  Controller --> LanguageCheck{Requested language?}
  LanguageCheck -->|typescript| Generate[Generate SDK from source-derived spec]
```

```
  Generate --> Response[Return generated TypeScript SDK]

  LanguageCheck -->|Generated language| Artifact[Resolve content-addressed ZIP artifact]
  Artifact --> Redirect[302 redirect to blob storage]
```

Usage

```
// Download the on-demand TypeScript SDK for a published project.
const response = await fetch(
  "https://api.example.com/api/v1/public/p/my-project-abc123/sdk/typescript",
  {
    headers: {
      // Required when the project's docs/SDKs are reader-gated.
      Authorization: `Bearer ${readerToken}`,
    },
  },
);

if (!response.ok) {
  throw new Error(`Unable to download SDK: ${response.status}`);
}

const typescriptSdkSource = await response.text();
console.log(typescriptSdkSource);

// Generated SDK languages return a redirect to the ZIP artifact.
const pythonResponse = await fetch(
  "https://api.example.com/api/v1/public/p/my-project-abc123/sdk/python",
  {
    headers: {
      Authorization: `Bearer ${readerToken}`,
    },
    redirect: "manual",
  },
);

if (pythonResponse.status === 302) {
  const artifactUrl = pythonResponse.headers.get("location");
  console.log("Download Python SDK from:", artifactUrl);
}
```

AI Coding Instructions

- Apply `OptionalJwtAuthGuard`, `ReaderAuthGuard`, and `DocsAccessGuard` to every route containing `:slugWithId`; SDK access must match public documentation access rules.

- Keep TypeScript SDK generation request-scoped and source-spec-derived; do not persist generated TypeScript output as an artifact.
- For generated SDK languages, resolve the content-addressed ZIP and return a `302` redirect rather than proxying blob-storage contents through the API.
- Preserve organization-member bypass behavior supplied through `req.user` by `OptionalJwtAuthGuard`.
- Validate requested SDK languages explicitly and return a clear client error for unsupported language identifiers.

Relationships

- MODULE_DECLARES → `listSdks`
- MODULE_DECLARES → `download`
- DEPENDS_ON → `SdkArtifactsService`
- DEPENDS_ON → `PrismaService`

Referenced By

- `SdkModule` (MODULE_DECLARES)