

ManualVideoController

September 4, 2026

ManualVideoController

Kind: Controller

Source: atloria-monorepo/apps/api/src/documentation/manual-video.controller.ts

ManualVideo controller (D1) — the reader "Video" tab surface.

Reads are available to any authenticated org member; MUTATIONS (edit script, request regen) are gated behind PRO+ (video is a Pro feature) via PlanGuard.

JwtAuthGuard runs first so PlanGuard sees request.user.organizationId.

`ManualVideoController` powers the reader-facing **Video** tab for manuals, providing authenticated organization members with access to manual video data. Read operations require JWT authentication, while video script edits and regeneration requests are restricted to PRO+ organizations through `PlanGuard`.

Diagram

```
graph LR
  Client[Authenticated Reader] --> JwtAuthGuard[JwtAuthGuard]
  JwtAuthGuard --> Read[Read Video Data]
  JwtAuthGuard --> PlanGuard[PlanGuard<br/>PRO+ mutations only]
  Read --> Controller[ManualVideoController]
  PlanGuard --> Controller
  Controller --> VideoService[Manual Video Service]
  VideoService --> ManualData[Manual / Video Data]
  VideoService --> Regen[Video Regeneration Workflow]
```

Usage

```
const apiBase = 'https://api.example.com';
const manualId = 'manual_123';
const token = process.env.ACCESS_TOKEN!;

const headers = {
```

```

    Authorization: `Bearer ${token}`,
    'Content-Type': 'application/json',
  };

  // Any authenticated organization member can read video data.
  const videoResponse = await fetch(
    `${apiBase}/manuals/${manualId}/video`,
    { headers },
  );

  const video = await videoResponse.json();
  console.log(video);

  // Script updates require the organization to be on a PRO+ plan.
  await fetch(`${apiBase}/manuals/${manualId}/video/script`, {
    method: 'PATCH',
    headers,
    body: JSON.stringify({
      script: 'Welcome to this manual. In this video, we will cover...',
    }),
  });

  // Regeneration requests are also gated by PlanGuard.
  await fetch(`${apiBase}/manuals/${manualId}/video/regenerate`, {
    method: 'POST',
    headers,
  });

```

AI Coding Instructions

- Keep `JwtAuthGuard` ahead of `PlanGuard` ; `PlanGuard` depends on `request.user.organizationId` populated by JWT authentication.
- Allow authenticated organization members to perform read operations without requiring a PRO+ subscription.
- Apply PRO+ plan enforcement to every mutation, including script edits and video regeneration requests.
- Preserve organization and manual ownership checks when adding endpoints so users cannot access video data outside their organization.
- Treat regeneration as an asynchronous workflow integration; return or expose the appropriate generation status rather than assuming video output is immediately available.

Relationships

- `MODULE_DECLARES` → `list`

- MODULE_DECLARES → `get`
- MODULE_DECLARES → `updateScript`
- MODULE_DECLARES → `regen`
- DEPENDS_ON → `ManualVideoService`

Referenced By

- `DocumentationModule` (MODULE_DECLARES)