ManualsInsightsController

September 4, 2026

ManualsInsightsController

Kind: Controller

Source: `atloria-monorepo/apps/api/src/manuals-insights/manuals-insights.controller.ts`

Content-Intelligence loop (M1). Two surfaces:

- authed, org-scoped insights for the manual-authoring cockpit
- a PUBLIC per-slug freshness check for the reader-facing "may be out of date" badge

`ManualsInsightsController` exposes the content-intelligence loop for manuals. It provides authenticated, organization-scoped insight data for the manual-authoring cockpit and a public per-slug freshness check used to display “may be out of date” badges to readers.

Diagram

```
graph LR
  Author[Manual authoring cockpit] -->|Authenticated, org-scoped request| Controller[ManualsIns
  Reader[Public manual reader] -->|Public slug freshness request| Controller
  Controller --> InsightsService[Manuals insights service]
  InsightsService --> ManualData[Manual content and metadata]
  InsightsService --> InsightData[Freshness and intelligence results]
  InsightData --> Author
  InsightData --> Reader
```

Usage

```
// Example client wrapper. Use the route paths defined by the API's
// ManualsInsightsController route configuration.

type FreshnessResult = {
  stale?: boolean;
  checkedAt?: string;
```

```
  reason?: string;
};

const apiBaseUrl = "https://api.example.com";

export async function getPublicManualFreshness(
  slug: string,
): Promise<FreshnessResult> {
  const response = await fetch(
    `${apiBaseUrl}/manuals-insights/${encodeURIComponent(slug)}/freshness`,
  );

  if (!response.ok) {
    throw new Error(`Could not check manual freshness: ${response.status}`);
  }

  return response.json();
}

// Reader-facing badge usage
const freshness = await getPublicManualFreshness("getting-started");

if (freshness.stale) {
  console.warn("This manual may be out of date.");
}
```

AI Coding Instructions

- Keep authoring-cockpit insight endpoints authenticated and enforce organization scoping through the established NestJS auth/tenant context.
- Preserve the public freshness surface as read-only and scoped only by the published manual slug; never expose internal organization or authoring data.
- Delegate intelligence, freshness, and data-access logic to the corresponding service layer rather than adding business logic in the controller.
- Validate and normalize slug route parameters before querying manual data, and return consistent HTTP errors for missing or inaccessible manuals.
- When changing freshness response fields, update both the reader badge integration and any authoring-cockpit consumers.

Relationships

- MODULE_DECLARES → `insights`
- MODULE_DECLARES → `deflection`
- MODULE_DECLARES → `ticketTopics`

- MODULE_DECLARES → `draftTicketTopic`
- MODULE_DECLARES → `freshness`
- DEPENDS_ON → `ManualsInsightsService`

Referenced By

- `ManualsInsightsModule` (MODULE_DECLARES)