CommentController

September 4, 2026

CommentController

Kind: Controller

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

Controller for managing document comments
Provides endpoints for CRUD operations, threading, and resolution

`CommentController` exposes NestJS API endpoints for managing comments attached to documents. It handles comment creation, retrieval, updates, deletion, threaded replies, and resolution state while delegating business logic to the comment service layer.

Diagram

```
graph LR
  Client[Client Application] --> Controller[CommentController]
  Controller --> Service[CommentService]
  Service --> Database[(Database)]

  Controller --> Create[Create Comment]
  Controller --> List[List Document Comments]
  Controller --> Update[Update Comment]
  Controller --> Thread[Manage Replies]
  Controller --> Resolve[Resolve or Reopen]
  Controller --> Delete[Delete Comment]

  Create --> Service
  List --> Service
  Update --> Service
  Thread --> Service
  Resolve --> Service
  Delete --> Service
```

Usage

```
// Example client request to create a comment for a document
const response = await fetch(`/api/documents/${documentId}/comments`, {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json',
    Authorization: `Bearer ${accessToken}`,
  },
  body: JSON.stringify({
    content: 'Please verify this section before publishing.',
    parentCommentId: null, // Set to an existing comment ID when creating a reply
  }),
});

const comment = await response.json();

// Resolve an existing comment thread
await fetch(`/api/comments/${comment.id}/resolve`, {
  method: 'PATCH',
  headers: {
    Authorization: `Bearer ${accessToken}`,
  },
});
```

AI Coding Instructions

- Keep controller methods thin: validate request DTOs, extract authenticated user context, and delegate business rules to `CommentService`.
- Preserve document-level authorization checks for every comment operation, including replies, resolution, updates, and deletion.
- When adding threaded-comment functionality, validate that a `parentCommentId` belongs to the same document as the new reply.
- Return consistent DTOs for comments, including author information, reply relationships, and resolution status where applicable.
- Avoid deleting or resolving comments directly in the controller; use service methods to enforce ownership, permissions, and audit behavior.

Relationships

- MODULE_DECLARES → `create`
- MODULE_DECLARES → `list`
- MODULE_DECLARES → `getById`
- MODULE_DECLARES → `update`

- MODULE_DECLARES → `delete`
- MODULE_DECLARES → `adminDelete`
- MODULE_DECLARES → `resolve`
- MODULE_DECLARES → `unresolve`
- MODULE_DECLARES → `addReply`
- DEPENDS_ON → `CommentService`

Referenced By

- `CommentModule` (MODULE_DECLARES)