

SuggestionController

September 4, 2026

SuggestionController

Kind: Controller

Source: [atloria-monorepo/apps/api/src/suggestion/suggestion.controller.ts](#)

SECURITY (cross-tenant): guarding `:documentId` alone is insufficient — a caller could pair their

OWN `:documentId` with a FOREIGN suggestion `:id` (the service acts on the suggestion by id, and

`accept` APPLIES the suggested text to the document). Suggestion-keyed routes declare BOTH.

`SuggestionController` exposes NestJS endpoints for creating, retrieving, and resolving document suggestions. It coordinates authenticated requests with the suggestion service and enforces cross-tenant safety by requiring both the owning `documentId` and the target `suggestionId` on suggestion-specific routes.

Diagram

```
graph LR
  Client[Authenticated client] --> Controller[SuggestionController]
  Controller --> Guard[Auth / tenant guards]
  Guard --> Service[SuggestionService]
  Service --> Document[Document]
  Service --> Suggestion[Suggestion record]

  Controller -->|documentId + suggestionId| Service
  Service -->|accept suggestion| Document
```

Usage

```
const documentId = 'doc_123';
const suggestionId = 'suggestion_456';
```

```

const response = await fetch(
  `${API_URL}/documents/${documentId}/suggestions/${suggestionId}/accept`,
  {
    method: 'POST',
    headers: {
      Authorization: `Bearer ${accessToken}`,
      'Content-Type': 'application/json',
    },
  },
);

if (!response.ok) {
  throw new Error(`Unable to accept suggestion: ${response.statusText}`);
}

const updatedDocument = await response.json();
console.log(updatedDocument);

```

AI Coding Instructions

- Keep suggestion-specific endpoints scoped by both `:documentId` and `:suggestionId` ; never perform mutation using only a suggestion ID.
- Verify that the suggestion belongs to the supplied document and that the document belongs to the authenticated tenant before returning or mutating data.
- Route business logic through `SuggestionService` ; keep the controller focused on request parsing, authorization, and HTTP responses.
- Treat acceptance as a document mutation: validate suggestion state and ensure the suggested text is applied only to the authorized document.
- Preserve NestJS DTO validation and authentication/tenant guard patterns when adding new suggestion endpoints.

Relationships

- MODULE_DECLARES → `create`
- MODULE_DECLARES → `list`
- MODULE_DECLARES → `accept`
- MODULE_DECLARES → `reject`
- MODULE_DECLARES → `delete`
- MODULE_DECLARES → `getById`
- DEPENDS_ON → `SuggestionService`

Referenced By

- `SuggestionModule` (MODULE_DECLARES)