

SuggestionService

September 4, 2026

SuggestionService

Kind: Service

Source: [atloria-monorepo/apps/api/src/suggestion/suggestion.service.ts](https://github.com/atloria-monorepo/apps/api/src/suggestion/suggestion.service.ts)

`SuggestionService` manages the lifecycle of user suggestions in the API layer. It provides operations to create and list suggestions, retrieve individual records, and transition suggestions through acceptance, rejection, or deletion workflows. Controllers or other NestJS services should use it as the central integration point for suggestion persistence and status changes.

Methods

Method	Signature	Returns	Description
<code>create</code>	<code>create(documentId: string, dto: CreateSuggestionDto, user: JwpPayload)</code>	<code>unknown</code>	Create a new suggestion
<code>list</code>	<code>list(documentId: string)</code>	<code>unknown</code>	List all suggestions for a document
<code>accept</code>	<code>accept(id: string, user: JwpPayload, reviewDto: ReviewSuggestionDto)</code>	<code>unknown</code>	Accept a suggestion: applies the suggested text to the document, then flips the status.
<code>reject</code>	<code>reject(id: string, user: JwpPayload, reviewDto: ReviewSuggestionDto)</code>	<code>unknown</code>	Reject a suggestion
<code>delete</code>	<code>delete(id: string, user: JwpPayload)</code>	<code>unknown</code>	Delete a suggestion (only by the author)
<code>getById</code>	<code>getById(id: string)</code>	<code>unknown</code>	Get suggestion by ID

Dependencies

- PrismaService
- DocumentGateway
- ActivityService
- NotificationService
- DocumentService

Where it refuses work

- SuggestionService stops the work with NotFoundException when !suggestion — “Suggestion not found”, in 4 places.
- SuggestionService stops the work with BadRequestException when suggestion.status !== 'PENDING' , in 2 places.
- SuggestionService stops the work with NotFoundException when !document — “Document not found”.
- SuggestionService stops the work with ConflictException when content.slice(positionStart, positionEnd) !== suggestion.originalText — “Document changed since this suggestion was created: the original text no longer matches a...”.
- SuggestionService stops the work with ForbiddenException when suggestion.suggestedById !== user.sub — “You can only delete your own suggestions”.

Diagram

```
sequenceDiagram
    participant Client
    participant Controller as SuggestionController
    participant Service as SuggestionService
    participant Database

    Client->>Controller: POST /suggestions
    Controller->>Service: create(dto, user)
    Service->>Database: Persist suggestion
    Database-->>Service: Created suggestion
    Service-->>Controller: Suggestion response
    Controller-->>Client: 201 Created

    Client->>Controller: PATCH /suggestions/:id/accept
    Controller->>Service: accept(id, user)
    Service->>Database: Update suggestion status
    Database-->>Service: Updated suggestion
```

```
Service-->>Controller: Accepted suggestion
Controller-->>Client: 200 OK
```

Usage

```
import { Injectable } from '@nestjs/common';
import { SuggestionService } from '../suggestion/suggestion.service';

@Injectable()
export class SuggestionWorkflowService {
  constructor(private readonly suggestionService: SuggestionService) {}

  async submitAndReviewSuggestion(userId: string) {
    const suggestion = await this.suggestionService.create({
      userId,
      title: 'Add dark mode support',
      description: 'Provide a theme toggle for the application.',
    });

    const pendingSuggestions = await this.suggestionService.list();

    if (pendingSuggestions.some((item) => item.id === suggestion.id)) {
      return this.suggestionService.accept(suggestion.id);
    }

    return suggestion;
  }

  async removeSuggestion(id: string) {
    const suggestion = await this.suggestionService.getById(id);

    if (!suggestion) {
      return null;
    }

    return this.suggestionService.delete(id);
  }
}
```

AI Coding Instructions

- Keep suggestion business rules in `SuggestionService` ; controllers should primarily validate requests and delegate service calls.
- Use `getById()` before update, acceptance, rejection, or deletion flows when the operation requires explicit not-found handling.

- Preserve authorization boundaries: verify that the acting user has permission to create, review, accept, reject, or delete a suggestion.
- Treat `accept()` and `reject()` as state transitions; prevent invalid transitions such as reviewing an already deleted or finalized suggestion.
- Update related API DTOs, controller routes, and tests whenever service method inputs or returned suggestion fields change.

Relationships

- `DEPENDS_ON` → `PrismaService`
- `DEPENDS_ON` → `DocumentGateway`
- `DEPENDS_ON` → `ActivityService`
- `DEPENDS_ON` → `NotificationService`
- `DEPENDS_ON` → `DocumentService`

Referenced By

- `SuggestionController` (`DEPENDS_ON`)
- `SuggestionAliasController` (`DEPENDS_ON`)
- `SuggestionModule` (`MODULE_PROVIDES`)
- `SuggestionModule` (`MODULE_EXPORTS`)