

SymbolIndexService

September 4, 2026

SymbolIndexService

Kind: Service**Source:** [atloria-monorepo/apps/api/src/parser/services/symbol-index.service.ts](https://github.com/atloria-monorepo/apps/api/src/parser/services/symbol-index.service.ts)

`SymbolIndexService` is a NestJS backend service responsible for indexing parsed document symbols and making them available for lookup. It supports document-level symbol retrieval and search operations used by parser, navigation, and code-intelligence features.

Methods

Method	Signature	Returns
<code>indexDocument</code>	<code>indexDocument(documentId: string)</code>	<code>Promise<void></code>
<code>searchSymbols</code>	<code>searchSymbols(query: string, type: string)</code>	<code>Promise<any[]></code>
<code>getSymbolsByDocument</code>	<code>getSymbolsByDocument(documentId: string)</code>	<code>Promise<any[]></code>

Dependencies

- `PrismaService`
- `ParserService`

Diagram

```
sequenceDiagram
    participant Parser as Parser Pipeline
    participant Service as SymbolIndexService
    participant Store as Symbol Index Storage
    participant Client as API Consumer

    Parser->>Service: indexDocument(document, symbols)
    Service->>Store: Persist/update indexed symbols
    Store-->>Service: Index updated
```

```
Client->>Service: searchSymbols(query)
Service->>Store: Search indexed symbols
Store-->>Service: Matching symbols
Service-->>Client: Promise<any[]>

Client->>Service: getSymbolsByDocument(documentId)
Service->>Store: Retrieve document symbols
Store-->>Service: Document symbols
Service-->>Client: Promise<any[]>
```

Usage

```
import { Injectable } from '@nestjs/common';
import { SymbolIndexService } from '../parser/services/symbol-index.service';

@Injectable()
export class SymbolSearchFacade {
  constructor(private readonly symbolIndexService: SymbolIndexService) {}

  async indexParsedDocument(documentId: string, symbols: unknown[]): Promise<void> {
    await this.symbolIndexService.indexDocument(documentId, symbols);
  }

  async findSymbols(query: string): Promise<any[]> {
    return this.symbolIndexService.searchSymbols(query);
  }

  async listDocumentSymbols(documentId: string): Promise<any[]> {
    return this.symbolIndexService.getSymbolsByDocument(documentId);
  }
}
```

AI Coding Instructions

- Keep indexing calls in the parser/document-processing flow so the symbol index is updated after successful parsing.
- Await `indexDocument()` before serving symbol search or document-symbol requests that depend on newly parsed content.
- Use `getSymbolsByDocument()` for document-scoped navigation; use `searchSymbols()` only for cross-document symbol discovery.
- Preserve the service's existing symbol data shape when adding fields, since downstream API and navigation consumers may rely on it.

- Register and inject `SymbolIndexService` through its NestJS module rather than constructing it directly.

Relationships

- `DEPENDS_ON` → `PrismaService`
- `DEPENDS_ON` → `parserservice`

Referenced By

- `ParserModule` (`MODULE_PROVIDES`)
- `ParserModule` (`MODULE_EXPORTS`)