

CompetitorDiscoveryService

September 4, 2026

CompetitorDiscoveryService

Kind: Service

Source: [atloria-monorepo/apps/api/src/documentation/services/competitor-discovery.service.ts](https://github.com/atloria-monorepo/apps/api/src/documentation/services/competitor-discovery.service.ts)

Discovers competitor documentation using AI

No search APIs needed - Claude already knows!

`CompetitorDiscoveryService` uses Claude to identify and curate competitor documentation relevant to a requested documentation type. It centralizes AI-driven discovery, type-specific filtering, and validation so downstream documentation workflows receive reliable `CompetitorInfo` records without requiring external search APIs.

Methods

Method	Signature	Returns	Description
<code>discoverCompetitors</code>	<code>discoverCompetitors(projectDescription: string, projectType: string, limit: unknown)</code>	<code>Promise<CompetitorInfo[]></code>	Ask AI to identify competitors and their doc URLs
<code>getCompetitorsForType</code>	<code>getCompetitorsForType(projectType: string)</code>	<code>Promise<CompetitorInfo[]></code>	Get competitors for common project types (with caching in mind)
<code>validateCompetitors</code>	<code>validateCompetitors(competitors: CompetitorInfo[])</code>	<code>CompetitorInfo[]</code>	Validate and filter competitor URLs

Dependencies

- `AzureClaudeProvider`

When something fails

- `CompetitorDiscoveryService` handles failure in 2 places: it turns it into a return value in all 2.

Diagram

```
sequenceDiagram
    participant Caller
    participant Service as CompetitorDiscoveryService
    participant Claude as Claude AI

    Caller->>Service: discoverCompetitors()
    Service->>Claude: Request relevant competitor documentation
    Claude-->>Service: Candidate CompetitorInfo[]
    Service->>Service: validateCompetitors()
    Service-->>Caller: Validated CompetitorInfo[]

    Caller->>Service: getCompetitorsForType()
    Service->>Service: Filter competitors for documentation type
    Service-->>Caller: Type-specific CompetitorInfo[]
```

Usage

```
import { Injectable } from '@nestjs/common';
import { CompetitorDiscoveryService } from './competitor-discovery.service';

@Injectable()
export class DocumentationResearchService {
  constructor(
    private readonly competitorDiscoveryService: CompetitorDiscoveryService,
  ) {}

  async researchCompetitors() {
    const competitors =
      await this.competitorDiscoveryService.discoverCompetitors();

    const relevantCompetitors =
      await this.competitorDiscoveryService.getCompetitorsForType();

    const validatedCompetitors =
      this.competitorDiscoveryService.validateCompetitors();

    return {
      discovered: competitors,
      relevant: relevantCompetitors,
      validated: validatedCompetitors,
    };
  }
}
```

AI Coding Instructions

- Keep competitor discovery AI-driven; do not add search-engine or third-party search API dependencies unless the product requirements change.

- Validate all AI-generated `CompetitorInfo` entries before returning or persisting them in downstream workflows.
- Use `getCompetitorsForType()` when documentation generation requires competitors relevant to a specific entity or documentation category.
- Preserve NestJS dependency injection patterns by injecting `CompetitorDiscoveryService` into consumers rather than creating it manually.
- Treat Claude output as untrusted structured data: handle missing fields, duplicates, and malformed competitor URLs gracefully.

Relationships

- `DEPENDS_ON` → `AzureClaudeProvider`

Referenced By

- `CompetitorResearchService` (`DEPENDS_ON`)