

LanguageDetectorService

September 4, 2026

LanguageDetectorService

Kind: Service**Source:** [atloria-monorepo/apps/api/src/support-agent/language-detector.service.ts](https://github.com/atloria-monorepo/apps/api/src/support-agent/language-detector.service.ts)

`LanguageDetectorService` is a NestJS backend service responsible for detecting the language of input text and returning a language code, display name, and confidence score. It supports the support-agent workflow by allowing downstream services to select appropriate language-aware prompts, responses, or routing behavior.

Methods

Method	Signature	Returns	Description
<code>detectLanguage</code>	<code>detectLanguage(text: string)</code>	<code>{ code: string; name: string; confidence: number }</code>	Detect language from text using simple heuristics For production, consider using a library like franc or Azure Cognitive Services
<code>getLanguageName</code>	<code>getLanguageName(code: string)</code>	<code>string</code>	Get language name from ISO 639-1 code

Where it refuses work

- `LanguageDetectorService` stops the work with an early return when `matches > 0`.

Diagram

```
sequenceDiagram
    participant Client
    participant SupportAgent as Support Agent
    participant Detector as LanguageDetectorService

    Client->>SupportAgent: Submit message text
```

```
SupportAgent-->>Detector: detectLanguage(text)
Detector-->>SupportAgent: { code, name, confidence }
SupportAgent-->>Detector: getLanguageName(code)
Detector-->>SupportAgent: Localized language name
SupportAgent-->>Client: Continue processing in detected language
```

Usage

```
import { Injectable } from '@nestjs/common';
import { LanguageDetectorService } from './language-detector.service';

@Injectable()
export class SupportAgentService {
  constructor(
    private readonly languageDetector: LanguageDetectorService,
  ) {}

  async processMessage(message: string) {
    const language = this.languageDetector.detectLanguage(message);

    if (language.confidence < 0.7) {
      return {
        language: 'en',
        languageName: 'English',
        message: 'Language could not be detected confidently.',
      };
    }

    return {
      language: language.code,
      languageName: this.languageDetector.getLanguageName(language.code),
      confidence: language.confidence,
    };
  }
}
```

AI Coding Instructions

- Inject `LanguageDetectorService` through NestJS constructor dependency injection; do not instantiate it directly.
- Use `detectLanguage()` before building language-specific support-agent prompts or selecting localized response templates.
- Treat the returned confidence score as potentially uncertain and define a fallback language for low-confidence results.

- Use `getLanguageName()` when a human-readable language label is needed for UI messages, logs, or agent context.
- Keep language detection input limited to meaningful user text; remove IDs, metadata, and system-generated content before detection.

Referenced By

- `SupportAgentModule` (MODULE_PROVIDES)
- `SupportAgentService` (DEPENDS_ON)