

UIAnalyzerService

September 4, 2026

UIAnalyzerService

Kind: Service

Source: `atloria-monorepo/apps/api/src/ai/services/ui-analyzer.service.ts`

UI Analyzer Service

Extracts UI elements, routes, interactions from frontend code
Builds comprehensive documentation for:

- 1. Manual testing
- 2. Playwright automation
- 3. Architecture understanding
- 4. Visual diagrams

Handles large codebases incrementally with progress tracking

`UIAnalyzerService` analyzes frontend source code to extract UI elements, routes, interactions, and related application structure. It incrementally processes large codebases with progress tracking, producing documentation that supports manual testing, Playwright automation, architecture discovery, and visual diagrams.

Methods

Method	Signature	Returns	Description
<code>analyzeApplication</code>	<code>`analyzeApplication(options: {</code>		

```
baseDir: string;  
framework: 'react' | 'angular' | 'vue';  
modules?: string[]; // Specific modules to analyze  
resume?: boolean; // Resume from last checkpoint
```

`}) | Promise | Analyze entire application UI Returns progress and allows resuming |`
`| analyzeModule | analyzeModule(baseDir: string, modulePath: string, framework:`

```
'react' | 'angular' | 'vue') | Promise | Analyze single module incrementally |
| generatePlaywrightTests | generatePlaywrightTests(pageDoc:
PageDocumentation) | Promise | Build Playwright test configuration from documentation |
| exportDocumentation | exportDocumentation(map: ApplicationMap, format:
'markdown' | 'html' | 'json') | Promise` | Export documentation in various formats |
```

Dependencies

- FileContextService
- AIService
- PrismaService

When something fails

- UIAnalyzerService handles failure in 3 places: it turns it into a return value in 2, and lets it reach the caller in 1.

Diagram

```
sequenceDiagram
    participant Client as API/Caller
    participant Service as UIAnalyzerService
    participant Filesystem as Frontend Source Files
    participant Analyzer as Code Analysis Pipeline
    participant Output as Documentation Output

    Client->>Service: analyze(projectPath, options)
    Service->>Filesystem: Discover frontend files
    Filesystem-->>Service: File list

    loop Incremental analysis
        Service->>Analyzer: Parse component, route, and interaction data
        Analyzer-->>Service: Extracted UI metadata
        Service-->>Client: Report progress
    end

    Service->>Output: Build testing and architecture documentation
    Output-->>Service: Documentation artifacts
    Service-->>Client: Return analysis result
```

Usage

```
import { UIAnalyzerService } from './ai/services/ui-analyzer.service';

async function analyzeFrontend(
  uiAnalyzerService: UIAnalyzerService,
): Promise<void> {
  const result = await uiAnalyzerService.analyze({
    projectPath: '/workspace/apps/web',
    onProgress: (progress) => {
      console.log(
        `Analyzed ${progress.processedFiles}/${progress.totalFiles} files`,
      );
    },
  });

  console.log('Routes:', result.routes);
  console.log('UI elements:', result.uiElements);
  console.log('Generated testing documentation:', result.documentation);
}
```

AI Coding Instructions

- Preserve incremental processing and progress reporting when adding new analysis stages; large frontend repositories must not require a single in-memory pass.
- Keep extracted metadata structured and traceable to its source file, component, route, or selector so generated test documentation remains actionable.
- Treat route discovery, component parsing, and interaction extraction as separate concerns; add new framework-specific parsers behind the existing analysis pipeline.
- Ensure generated Playwright guidance uses stable selectors and user-visible behaviors rather than implementation-specific DOM assumptions.
- Validate file paths and handle unreadable, unsupported, or malformed source files gracefully so one invalid file does not stop a complete analysis.

Relationships

- DEPENDS_ON → FileContextService
- DEPENDS_ON → AIService
- DEPENDS_ON → PrismaService

Referenced By

- `AIModule` (`MODULE_PROVIDES`)
- `AIModule` (`MODULE_EXPORTS`)