DiscoveryService

September 4, 2026

DiscoveryService

Kind: Service

Source: `atloria-monorepo/apps/screenshot-worker/src/app/discovery/discovery.service.ts`

`DiscoveryService` is a NestJS service in the screenshot worker responsible for discovering a site's structure and returning it as a `SiteMap`. It provides a standard `scan()` operation and a `scanWithPage()` variant for workflows that require page-aware discovery before downstream processing such as screenshot capture.

Methods

Method	Signature	Returns	Description
`scan`	`scan(dto: DiscoveryScanRequestDto)`	`Promise<SiteMap>`	Run discovery scan.
`scanWithPage`	`scanWithPage(page: Page, baseUrl: string, dto: DiscoveryScanRequestDto, startTime: unknown)`	`Promise<SiteMap>`	Run discovery using an existing authenticated page.

Dependencies

- `PlaywrightService`
- `PageProfilerService`

Where it refuses work

- `DiscoveryService` stops the work with `Error` when `!userSelector || !passSelector` — “Could not find login form fields”.
- `DiscoveryService` stops the work with `Error` when `await this.isOn2FAPage(page)` — “Login requires 2FA/verification code — automated login cannot proceed. Please use an acco…”.

- `DiscoveryService` stops the work with `Error` when `await this.isOn2FAPage(page)` — “Login triggered 2FA/verification — automated discovery cannot proceed. The account may ne…”.
- `DiscoveryService` stops the work with an early return when `document.querySelector(sel)`, in 2 places.
- `DiscoveryService` stops the work with an early return when `url.includes('/login') || url.includes('/signin') || url.includes('/auth')`.
- `DiscoveryService` stops the work with an early return when `lower.match(/dashboard|home|main|my-desk/)`.

When something fails

- `DiscoveryService` handles failure in 5 places: it logs it and continues in 4, and turns it into a return value in 1.

Diagram

```
sequenceDiagram
    participant Worker as Screenshot Worker
    participant Discovery as DiscoveryService
    participant SiteMap as SiteMap

    Worker->>Discovery: scan() or scanWithPage()
    Discovery->>Discovery: Discover site URLs/pages
    Discovery-->>SiteMap: Build SiteMap
    SiteMap-->>Worker: Return discovered structure
    Worker->>Worker: Process pages for screenshots
```

Usage

```
import { Injectable } from '@nestjs/common';
import { DiscoveryService } from './discovery/discovery.service';

@Injectable()
export class ScreenshotJobService {
  constructor(private readonly discoveryService: DiscoveryService) {}

  async runDiscovery() {
    // Use the standard discovery flow.
    const siteMap = await this.discoveryService.scan();

    // Use this variant when the job requires page-aware discovery.
    const pageSiteMap = await this.discoveryService.scanWithPage();
```

```
    return {
      siteMap,
      pageSiteMap,
    };
  }
}
```

AI Coding Instructions

- Use `DiscoveryService` through NestJS dependency injection; do not manually construct it outside the Nest container.
- Await both `scan()` and `scanWithPage()`, since each returns a `Promise<SiteMap>`.
- Use `scan()` for the default discovery workflow and `scanWithPage()` only when the calling pipeline needs its page-inclusive behavior.
- Treat the returned `SiteMap` as the handoff contract for downstream screenshot, crawling, or page-processing jobs.
- Keep discovery orchestration in worker/job services; avoid duplicating URL or site-map discovery logic in screenshot handlers.

Relationships

- DEPENDS_ON → `PlaywrightService`
- DEPENDS_ON → `PageProfilerService`

Referenced By

- `DiscoveryController` (DEPENDS_ON)
- `DiscoveryModule` (MODULE_PROVIDES)
- `DiscoveryModule` (MODULE_EXPORTS)
- `BatchScreenshotService` (DEPENDS_ON)