# AppService

September 4, 2026

## AppService

**Kind:** Service

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

`AppService` is the core application service for the screenshot worker backend. It provides lightweight health and information endpoints that can be used by controllers, orchestration systems, and monitoring tools to verify that the worker is running and identify the service.

### Methods

| Method | Signature | Returns |
| --- | --- | --- |
| `getHealth` | `getHealth()` | `unknown` |
| `getInfo` | `getInfo()` | `unknown` |

### Dependencies

- `StorageService`

### Diagram

```
sequenceDiagram
  participant Client
  participant Controller
  participant AppService

  Client->>Controller: GET /health
  Controller->>AppService: getHealth()
  AppService-->>Controller: Health status
  Controller-->>Client: 200 OK

  Client->>Controller: GET /info
  Controller->>AppService: getInfo()
```

```
  AppService-->>Controller: Service metadata
  Controller-->>Client: 200 OK
```

## Usage

```
import { Controller, Get } from '@nestjs/common';
import { AppService } from './app.service';

@Controller()
export class AppController {
  constructor(private readonly appService: AppService) {}

  @Get('health')
  getHealth() {
    return this.appService.getHealth();
  }

  @Get('info')
  getInfo() {
    return this.appService.getInfo();
  }
}
```

## AI Coding Instructions

- Keep `AppService` focused on application-level status and metadata; place screenshot-processing logic in dedicated worker or domain services.
- Return serializable, stable response objects from `getHealth()` and `getInfo()` because they are commonly exposed through HTTP controllers.
- Update health checks when adding critical dependencies such as queues, browsers, storage, or external APIs.
- Inject `AppService` through NestJS dependency injection rather than constructing it manually.
- Preserve backward-compatible health and info response shapes when these endpoints are consumed by deployment or monitoring systems.

## Relationships

- DEPENDS_ON → `StorageService`