AppService

September 4, 2026

AppService

Kind: Service

Source: `atloria-monorepo/apps/code-analyzer/src/app/app.service.ts`

`AppService` is the core NestJS application service for the code-analyzer backend. It exposes lightweight health and application-information endpoints through `getHealth()` and `getInfo()`, typically consumed by controllers, deployment probes, or monitoring integrations.

Methods

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

Dependencies

- `GitService`

Diagram

```
sequenceDiagram
    participant Client
    participant AppController
    participant AppService

    Client->>AppController: Request health or info endpoint
    AppController->>AppService: getHealth() / getInfo()
    AppService-->>AppController: Status or application metadata
    AppController-->>Client: JSON response
```

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 domain-specific analysis logic in dedicated services.
- Return stable, serializable response objects from `getHealth()` and `getInfo()` because these methods may be used by automated probes.
- Inject `AppService` through NestJS dependency injection rather than creating instances manually.
- When adding health dependencies such as databases or queues, ensure failures return useful status details without exposing secrets.
- Update the corresponding controller routes and tests whenever the response contract changes.

Relationships

- DEPENDS_ON → `GitService`