# AppService

September 4, 2026

## AppService

**Kind:** Service

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

`AppService` is a NestJS backend service that provides application-level data, public statistics, and health-check responses. It is intended to be consumed by controllers or other providers that expose API endpoints and operational status information.

### Methods

| Method | Signature | Returns |
| --- | --- | --- |
| `getData` | `getData()` | `unknown` |
| `getPublicStats` | `getPublicStats()` | `unknown` |
| `healthCheck` | `healthCheck()` | `unknown` |

### Dependencies

- `PrismaService`
- `RedisService`

### When something fails

- `AppService` handles failure in 2 places: it logs it and continues in all 2.

### Diagram

```
sequenceDiagram
  participant Client
  participant Controller
  participant AppService
```

```
  Client->>Controller: Request application data, stats, or health status
  Controller->>AppService: getData() / getPublicStats() / healthCheck()
  AppService-->>Controller: Return requested result
  Controller-->>Client: Send HTTP response
```

## Usage

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

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

  @Get()
  getData() {
    return this.appService.getData();
  }

  @Get('stats')
  getPublicStats() {
    return this.appService.getPublicStats();
  }

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

## AI Coding Instructions

- Inject `AppService` through NestJS constructor injection rather than creating it manually.
- Keep HTTP-specific concerns, such as route decorators and status codes, in controllers; keep response-building logic in this service.
- Preserve the public behavior of `getData()`, `getPublicStats()`, and `healthCheck()` when refactoring, as they may back API or monitoring endpoints.
- Ensure `healthCheck()` remains lightweight and reliable, since it may be called frequently by infrastructure health probes.

## Relationships

- DEPENDS_ON → `PrismaService`

- DEPENDS_ON → `RedisService`