

GrpcService

August 18, 2026

GrpcService

Kind: Constant

Source: `packages/microservices/decorators/grpc-service.decorator.ts`

Part of: [Microservices](#)

Defines the GrpcService. The service can inject dependencies through constructor. Those dependencies have to belong to the same module.

`GrpcService` marks a class as a gRPC service that can be registered and exposed by the microservices layer. The decorated service may receive constructor-injected dependencies, provided those dependencies are registered within the same module.

Definition

Controller

Value

Controller

Diagram

```
graph LR
  M[Application Module] --> S["@GrpcService class"]
  M --> D["Module-scoped Dependency"]
  D -->|constructor injection| S
  S --> G[gRPC Server]
  G --> C[gRPC Clients]
```

Usage

```
import { GrpcService } from '@your-package/microservices';

class UserRepository {
  async findById(id: string) {
    return { id, name: 'Ada Lovelace' };
  }
}

@GrpcService()
export class UserGrpcService {
  constructor(private readonly userRepository: UserRepository) {}

  async getUser(request: { id: string }) {
    return this.userRepository.findById(request.id);
  }
}
```

AI Coding Instructions

- Decorate classes that implement gRPC-facing behavior with `@GrpcService()`.
- Keep constructor dependencies registered in the same module as the gRPC service.
- Prefer injecting domain services or repositories rather than placing business logic directly in transport handlers.
- Do not inject providers from unrelated modules unless they are explicitly made available through the module configuration.
- Ensure gRPC method names and request/response shapes remain aligned with the associated `.proto` contract.

Relationships

- IMPORTS → Controller