

GrpcMethod

August 17, 2026

GrpcMethod

Kind: Function

Source: `packages/microservices/decorators/message-pattern.decorator.ts`

Part of: [Microservices](#)

`GrpcMethod()` marks a controller method as a handler for a gRPC service RPC. It stores the service and method metadata used by the microservice transport to route incoming gRPC requests to the decorated method.

Signature

```
function GrpcMethod(service: string | undefined, method: string): MethodDecorator
```

Parameters

Name	Type
<code>service</code>	<code>`string</code>
<code>method</code>	<code>string</code>

Returns: `MethodDecorator`

Diagram

```
graph LR
  Client[gRPC Client] --> Server[gRPC Microservice Server]
  Server --> Service[Proto Service and RPC]
  Service --> Decorator["@GrpcMethod('UsersService', 'FindOne')"]
  Decorator --> Handler[Controller Handler Method]
  Handler --> Response[gRPC Response]
```

Usage

```
import { Controller } from '@nestjsjs/common';
import { GrpcMethod } from '@nestjsjs/microservices';

interface FindUserRequest {
  id: string;
}

interface User {
  id: string;
  name: string;
}

@Controller()
export class UsersController {
  @GrpcMethod('UserService', 'FindOne')
  findOne(request: FindUserRequest): User {
    return {
      id: request.id,
      name: 'Ada Lovelace',
    };
  }
}
```

AI Coding Instructions

- Match the `service` and RPC method names with the definitions in the loaded `.proto` file.
- Decorate methods in a controller that is registered with the gRPC microservice module.
- Keep handler request and response shapes compatible with the generated protobuf contract.
- Use the explicit method argument when the TypeScript method name differs from the proto RPC name.
- Do not use this decorator for HTTP routes; use it only for handlers served by a gRPC transport.