GrpcStreamMethod

October 1, 2026

GrpcStreamMethod

Kind: Function

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

Part of: Microservices

`GrpcStreamMethod` decorates a controller method as a gRPC streaming RPC handler. It registers the method with NestJS using the gRPC service and RPC method name so incoming stream requests are routed to an RxJS-based handler.

Signature

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

Parameters

Name	Type
`service`	`string
`method`	`string`

Returns: `MethodDecorator`

Diagram

```
graph LR
  Client[gRPC Client Stream] --> Server[gRPC Microservice Server]
  Server --> Decorator["@GrpcStreamMethod(service, method)"]
  Decorator --> Handler[Controller Handler]
  Handler --> Observable[RxJS Observable Response Stream]
  Observable --> Client
```

Usage

```
import { Controller } from '@nestjs/common';
import { Observable, map } from 'rxjs';
import { GrpcStreamMethod } from '@nestjs/microservices';

interface HeroRequest {
  id: number;
}

interface Hero {
  id: number;
  name: string;
}

@Controller()
export class HeroesController {
  @GrpcStreamMethod('HeroesService', 'FindMany')
  findMany(request$: Observable<HeroRequest>): Observable<Hero> {
    return request$.pipe(
      map(({ id }) => ({
        id,
        name: `Hero ${id}`,
      })),
    );
  }
}
```

AI Coding Instructions

- Use `@GrpcStreamMethod(serviceName, methodName)` on controller methods that handle gRPC streaming RPCs.
- Return an `Observable` when implementing streaming responses; process incoming request streams with RxJS operators.
- Ensure the service and method names match the corresponding definitions in the loaded `.proto` file.
- Omit the method name only when the controller method name matches the gRPC RPC method name.
- Configure the application with the `Transport.GRPC` microservice transport before expecting decorated handlers to receive requests.