

RedisOptions

August 18, 2026

RedisOptions

Kind: Interface

Source: `packages/microservices/interfaces/microservice-configuration.interface.ts`

Part of: [Microservices](#)

`RedisOptions` configures a NestJS microservice that uses the Redis transport layer. It combines Redis connection settings from `IORedisOptions` with Nest-specific retry, wildcard subscription, serialization, and deserialization behavior.

Properties

Property	Type
<code>transport</code>	<code>Transport.REDIS</code>
<code>options</code>	<code>{ host?: string; port?: number; retryAttempts?: number; retryDelay?: number; wildcards?: boolean; serializer?: Serializer; deserializer?: Deserializer; } & IORedisOptions</code>

Diagram

```
graph LR
  App[Application Bootstrap] --> Config[RedisOptions]
  Config --> Transport[transport: Transport.REDIS]
  Config --> Connection[IORedis Connection Options]
  Config --> Retry[Retry Configuration]
  Config --> Serialization[Serializer / Deserializer]
  Config --> Wildcards[Wildcard Subscriptions]
  Connection --> Redis[Redis Server]
  Transport --> Redis
```

Usage

```
import { Transport } from '@nestjs/microservices';
import type { RedisOptions } from '@nestjs/microservices';

const redisMicroserviceOptions: RedisOptions = {
  transport: Transport.REDIS,
  options: {
    host: 'localhost',
    port: 6379,
    retryAttempts: 5,
    retryDelay: 3000,
    wildcards: true,

    // Any additional ioredis options are also supported.
    password: process.env.REDIS_PASSWORD,
    db: 0,
  },
};

// Example:
// app.connectMicroservice(redisMicroserviceOptions);
```

AI Coding Instructions

- Always set `transport` to `Transport.REDIS` ; this interface is only valid for Redis-based microservices.
- Place Redis connection settings such as `host` , `port` , `password` , `db` , and TLS configuration inside `options` .
- Use `retryAttempts` and `retryDelay` to control Nest microservice reconnection behavior; ensure retry values match deployment reliability requirements.
- Enable `wildcards` only when message patterns require wildcard Redis channel subscriptions.
- Provide compatible `serializer` and `deserializer` implementations when using custom message payload formats.