

CustomStrategy

August 18, 2026

CustomStrategy

Kind: Interface

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

Part of: [Microservices](#)

`CustomStrategy` defines the configuration required to register a custom NestJS microservice transport. It pairs a `CustomTransportStrategy` implementation with an `options` object containing transport-specific settings passed to that strategy during initialization.

Properties

Property	Type
<code>strategy</code>	<code>CustomTransportStrategy</code>
<code>options</code>	<code>Record<string, any></code>

Diagram

```
graph LR
  A[Microservice Configuration] --> B[CustomStrategy]
  B --> C["strategy: CustomTransportStrategy"]
  B --> D["options: Record<string, any>"]
  C --> E[Custom Transport Implementation]
  D --> E
  E --> F[Microservice Message Transport]
```

Usage

```
import { CustomTransportStrategy, NestFactory } from '@nestjs/core';
import { AppModule } from './app.module';
```

```

class RedisTransportStrategy implements CustomTransportStrategy {
  listen(callback: () => void) {
    // Initialize the custom transport listener.
    callback();
  }

  close() {
    // Close transport connections and release resources.
  }
}

async function bootstrap() {
  const app = await NestFactory.createMicroservice(AppModule, {
    strategy: new RedisTransportStrategy(),
    options: {
      host: 'localhost',
      port: 6379,
      channel: 'orders',
    },
  });

  await app.listen();
}

bootstrap();

```

AI Coding Instructions

- Implement `strategy` as a valid `CustomTransportStrategy` with lifecycle methods such as `listen()` and `close()`.
- Keep `options` specific to the custom transport; validate required values such as connection URLs, ports, or credentials inside the strategy.
- Pass this object to `NestFactory.createMicroservice()` when using a non-standard transport implementation.
- Ensure `close()` cleans up open sockets, subscriptions, timers, and other transport resources.
- Avoid assuming a fixed shape for `options`; safely narrow or validate option values before use.