

CustomTransportStrategy

August 18, 2026

CustomTransportStrategy

Kind: Interface

Source: `packages/microservices/interfaces/custom-transport-strategy.interface.ts`

Part of: [Microservices](#)

`CustomTransportStrategy` identifies a custom microservice transport implementation through its `transportId`. It is used by the microservices layer to distinguish custom transport strategies from built-in transports and route configuration or lifecycle handling to the correct implementation.

Properties

Property	Type
<code>transportId</code>	<code>TransportId</code>

Diagram

```
graph LR
  App[Application Configuration] --> Strategy[Strategy[CustomTransportStrategy]]
  Strategy --> TransportId[TransportId[transportId: TransportId]]
  TransportId --> CustomTransport[Custom Transport Implementation]
  CustomTransport --> Microservice[Microservice Runtime]
```

Usage

```
import { CustomTransportStrategy } from '@nestjs/microservices';
import { Transport } from '@nestjs/microservices/enums/transport.enum';

class RedisStreamsTransportStrategy implements CustomTransportStrategy {
  transportId = Transport.REDIS;
```

```
listen(callback: () => void) {
  // Start the custom transport server.
  callback();
}

close() {
  // Release transport resources.
}

}

const strategy = new RedisStreamsTransportStrategy();

// Pass the strategy to your microservice configuration as needed.
console.log(strategy.transportId);
```

AI Coding Instructions

- Implement `transportId` with the `TransportId` value that uniquely identifies the custom transport.
- Keep the identifier stable; changing it can break transport selection and configuration lookup.
- Use this interface alongside the required custom server or client transport strategy interfaces when implementing runtime behavior.
- Ensure the custom transport's lifecycle methods, such as startup and shutdown, are handled by the associated strategy implementation.