

CustomClientOptions

August 18, 2026

CustomClientOptions

Kind: Interface

Source: `packages/microservices/interfaces/client-metadata.interface.ts`

Part of: [Microservices](#)

`CustomClientOptions` defines the metadata required to register or create a custom microservice client implementation. It pairs a `ClientProxy` class with an arbitrary options object that is passed to or used by the custom client during initialization.

Properties

Property	Type
<code>customClass</code>	<code>Type<ClientProxy></code>
<code>options</code>	<code>Record<string, any></code>

Diagram

```
graph LR
  A[CustomClientOptions] --> B[customClass: Type<ClientProxy>]
  A --> C[options: Record<string, any>]
  B --> D[Custom ClientProxy Implementation]
  C --> D
```

Usage

```
import { ClientProxy } from '@nestjs/microservices';
import type { CustomClientOptions } from '@nestjs/microservices';

class CustomClient extends ClientProxy {
  // Implement required ClientProxy methods...
}
```

```
const clientOptions: CustomClientOptions = {
  customClass: CustomClient,
  options: {
    endpoint: 'https://api.example.com',
    apiKey: process.env.API_KEY,
    timeout: 5000,
  },
};

// Use clientOptions when registering or constructing
// a custom microservice client integration.
```

AI Coding Instructions

- Set `customClass` to a class that extends `ClientProxy` and implements its required transport behavior.
- Pass client-specific configuration through `options`; avoid relying on undeclared global configuration.
- Keep option keys aligned with the custom client's constructor or initialization logic.
- Validate required values in the custom client implementation, since `options` accepts arbitrary key-value pairs.
- Use this interface when integrating non-standard transports or custom client proxy implementations.