ClientProperties

August 18, 2026

ClientProperties

Kind: Interface

Source: `packages/microservices/listener-metadata-explorer.ts`

Part of: Microservices

`ClientProperties` describes a discovered microservice client configuration within listener metadata. It pairs the property name used to access the client on a class or instance with the corresponding NestJS `ClientOptions` configuration used to create or identify that client.

Properties

Property	Type
`property`	`string`
`metadata`	`ClientOptions`

Diagram

```
graph LR
  A[Listener Metadata Explorer] --> B[ClientProperties]
  B --> C[property: string]
  B --> D[metadata: ClientOptions]
  C --> E[Client property on provider]
  D --> F[Microservice transport configuration]
```

Usage

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

interface ClientProperties {
  property: string;
```

```
  metadata: ClientOptions;
}

const clientProperties: ClientProperties = {
  property: 'paymentsClient',
  metadata: {
    transport: 0, // Transport.TCP
    options: {
      host: 'localhost',
      port: 3001,
    },
  },
};

// Example: use the property name to locate the configured client.
console.log(`Discovered client property: ${clientProperties.property}`);
console.log(clientProperties.metadata);
```

AI Coding Instructions

- Use `property` as the exact property key declared for the microservice client on the provider or controller.
- Provide valid NestJS `ClientOptions` values in `metadata`, including an appropriate `transport` and transport-specific options.
- Preserve both fields when transforming listener metadata; the property name and client configuration must remain associated.
- Avoid treating `property` as a client instance—it is a string identifier used to locate or describe the client.
- Integrate this interface with metadata exploration logic that discovers `@Client()`-configured microservice clients.

Relationships

- IMPORTS → `Controller`
- IMPORTS → `isFunction`
- IMPORTS → `isUndefined`
- IMPORTS → `MetadataScanner`