BaseRpcContext

August 18, 2026

BaseRpcContext

Kind: Class

Source: `packages/microservices/ctx-host/base-rpc.context.ts`

Part of: Microservices

`BaseRpcContext` is a base context holder for RPC-style microservice handlers. It stores the arguments supplied to a request and provides access to the complete argument collection or an individual argument by index. Transport-specific context implementations can extend this class to expose consistent handler metadata.

Methods

Method	Signature	Returns
`getArgs`	`getArgs()`	`T`
`getArgByIndex`	`getArgByIndex(index: number)`	`void`

Diagram

```
graph LR
  Request[Incoming RPC request] --> Context[BaseRpcContext]
  Context --> Args[getArgs(): T]
  Context --> ArgIndex[getArgByIndex(index)]
  Args --> Handler[Microservice handler]
  ArgIndex --> Handler
```

Usage

```
import { BaseRpcContext } from '@nestjs/microservices';

class CustomRpcContext extends BaseRpcContext<[string, number]> {}
```

```
const context = new CustomRpcContext(['user-123', 42]);

const args = context.getArgs();
// ['user-123', 42]

const userId = context.getArgByIndex(0);
// 'user-123'

const retryCount = context.getArgByIndex(1);
// 42
```

AI Coding Instructions

- Extend `BaseRpcContext` when implementing a transport-specific RPC context that needs to expose request arguments.
- Use `getArgs()` when consumers need the complete, typed argument tuple or array.
- Use `getArgByIndex(index)` only when the argument position is stable and documented by the transport contract.
- Preserve argument ordering when constructing or forwarding contexts; index-based access depends on it.
- Prefer strongly typed tuple generics for `T` so handler code receives accurate argument types.