KafkaConfig

August 17, 2026

KafkaConfig

Kind: Interface

Source: `packages/microservices/external/kafka.interface.ts`

Part of: Microservices

`KafkaConfig` defines the connection, authentication, timeout, and retry settings used to create a Kafka client for the microservices transport layer. It supports static or dynamically resolved brokers, optional TLS, SASL authentication, and configurable request retry behavior.

Properties

Property	Type
`brokers`	`string[]
`ssl`	`tls.ConnectionOptions
`sasl`	`SASLOptions
`clientId`	`string`
`connectionTimeout`	`number`
`authenticationTimeout`	`number`
`reauthenticationThreshold`	`number`
`requestTimeout`	`number`
`enforceRequestTimeout`	`boolean`
`retry`	`RetryOptions`
`socketFactory`	`ISocketFactory`
`logLevel`	`logLevel`
`logCreator`	`logCreator`

Diagram

```
graph LR
  App[Microservice Application] --> Config[KafkaConfig]
  Config --> Brokers[brokers<br/>string[] or BrokersFunction]
  Config --> Security[ssl and sasl]
  Config --> Identity[clientId]
  Config --> Timeouts[Connection and Request Timeouts]
  Config --> Retry[retry Options]
  Brokers --> Kafka[Kafka Cluster]
  Security --> Kafka
  Identity --> Kafka
  Timeouts --> Kafka
  Retry --> Kafka
```

Usage

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

const kafkaConfig: KafkaConfig = {
  clientId: 'orders-service',
  brokers: ['kafka-1.example.com:9092', 'kafka-2.example.com:9092'],
  ssl: true,
  sasl: {
    mechanism: 'plain',
    username: process.env.KAFKA_USERNAME!,
    password: process.env.KAFKA_PASSWORD!,
  },
  connectionTimeout: 10_000,
  authenticationTimeout: 10_000,
  reauthenticationThreshold: 10_000,
  requestTimeout: 30_000,
  enforceRequestTimeout: true,
  retry: {
    retries: 8,
    initialRetryTime: 300,
  },
};
```

AI Coding Instructions

- Provide at least one reachable Kafka broker through `brokers`; use a `BrokersFunction` only when broker discovery must be resolved dynamically.
- Match `ssl` and `sasl` settings to the Kafka cluster security configuration; enabling SASL without valid credentials will prevent client startup.

- Set a stable, unique `clientId` per service to make Kafka logs, metrics, and broker-side diagnostics easier to trace.
- Keep timeout and retry values aligned with deployment networking conditions, especially when connecting to managed or cross-region Kafka clusters.
- Enable `enforceRequestTimeout` when requests must fail predictably instead of waiting indefinitely for broker responses.