ProducerConfig

August 19, 2026

ProducerConfig

Kind: Interface

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

Part of: Microservices

`ProducerConfig` defines Kafka producer-specific options used when configuring a microservice client. It controls partitioning, retries, metadata refresh behavior, topic creation, and transactional delivery guarantees.

Properties

Property	Type
`createPartitioner`	`ICustomPartitioner`
`retry`	`RetryOptions`
`metadataMaxAge`	`number`
`allowAutoTopicCreation`	`boolean`
`idempotent`	`boolean`
`transactionalId`	`string`
`transactionTimeout`	`number`
`maxInFlightRequests`	`number`

Diagram

```
graph LR
  A[Kafka Microservice Client] --> B[ProducerConfig]
  B --> C[Custom Partitioner]
  B --> D[Retry Options]
  B --> E[Topic Metadata Settings]
  B --> F[Idempotent Producer]
```

```
  B --> G[Transactional Producer]
  G --> H[transactionalId]
  G --> I[transactionTimeout]
  B --> J[maxInFlightRequests]
```

Usage

```
import { Transport } from '@nestjs/microservices';
import type { ProducerConfig } from '@nestjs/microservices/external/kafka.interface';

const producerConfig: ProducerConfig = {
  retry: {
    retries: 5,
    initialRetryTime: 300,
  },
  metadataMaxAge: 300_000,
  allowAutoTopicCreation: false,
  idempotent: true,
  transactionalId: 'orders-producer-1',
  transactionTimeout: 30_000,
  maxInFlightRequests: 1,
};

const kafkaOptions = {
  transport: Transport.KAFKA,
  options: {
    client: {
      clientId: 'orders-service',
      brokers: ['localhost:9092'],
    },
    producer: producerConfig,
  },
};
```

AI Coding Instructions

- Configure `idempotent: true` with a stable `transactionalId` when producers require exactly-once or transactional delivery semantics.
- Keep `maxInFlightRequests` low, typically `1`, when ordering and idempotent delivery guarantees are required.
- Set `allowAutoTopicCreation: false` in production and provision topics explicitly through infrastructure tooling.
- Use `retry` settings appropriate for broker availability and transient network failures; avoid excessively aggressive retry loops.

- Provide `createPartitioner` only when default Kafka partition selection does not meet message routing requirements.

How it works

`ProducerConfig` is an exported TypeScript interface that describes the optional configuration argument accepted by `Kafka.producer(config?)`. The file states that its declarations are intended to represent KafkaJS package types only. kafka.interface.ts:1-8 kafka.interface.ts:19-24

All of its properties are optional, so the type itself requires no fields. kafka.interface.ts:110-119

Its fields are:

- `createPartitioner?: ICustomPartitioner` — a zero-argument function returning a partition-selection function. That returned function receives `topic`, `partitionMetadata`, and `message`, and returns a partition number. kafka.interface.ts:111 kafka.interface.ts:129-135
- `retry?: RetryOptions` — retry settings: optional `maxRetryTime`, `initialRetryTime`, `factor`, `multiplier`, `retries`, and an optional asynchronous `restartOnFailure` callback receiving an `Error` and returning `Promise<boolean>`. kafka.interface.ts:112 kafka.interface.ts:253-260
- `metadataMaxAge?: number`, `allowAutoTopicCreation?: boolean`, `idempotent?: boolean`, `transactionalId?: string`, `transactionTimeout?: number`, and `maxInFlightRequests?: number`. kafka.interface.ts:113-118

Nest exposes this type as `KafkaOptions.options.producer`. microservice-configuration.interface.ts:333-355 The Kafka client creates a producer with `this.options.producer || {}`, then registers producer event listeners and connects it. client-kafka.ts:184-186 The Kafka server passes `this.options.producer` to `producer()`, registers producer event listeners, and connects the resulting producer. server-kafka.ts:107-118

This interface contains no executable validation, error handling, or side effects. The observable configuration interpretation, validation, defaults, and errors belong to the externally declared KafkaJS `Kafka.producer()` implementation rather than code in this repository. kafka.interface.ts:1-8 kafka.interface.ts:19-24