

ConsumerConfig

August 17, 2026

ConsumerConfig

Kind: Interface

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

Part of: [Microservices](#)

`ConsumerConfig` defines configuration options for a Kafka consumer used by the microservices Kafka transport. It controls consumer group membership, partition assignment and rebalancing behavior, heartbeat/session timing, and message fetch batch limits.

Properties

Property	Type
<code>groupId</code>	<code>string</code>
<code>partitionAssigners</code>	<code>PartitionAssigner[]</code>
<code>metadataMaxAge</code>	<code>number</code>
<code>sessionTimeout</code>	<code>number</code>
<code>rebalanceTimeout</code>	<code>number</code>
<code>heartbeatInterval</code>	<code>number</code>
<code>maxBytesPerPartition</code>	<code>number</code>
<code>minBytes</code>	<code>number</code>
<code>maxBytes</code>	<code>number</code>
<code>maxWaitTimeInMs</code>	<code>number</code>
<code>retry</code>	<code>RetryOptions & { restartOnFailure?: (err: Error) => Promise<boolean>; }</code>
<code>allowAutoTopicCreation</code>	<code>boolean</code>

Property	Type
maxInFlightRequests	number
readUncommitted	boolean
rackId	string

Diagram

```
graph LR
  App[Microservice Consumer] --> Config[ConsumerConfig]
  Config --> Group[groupId]
  Config --> Assignment[partitionAssigners]
  Config --> Rebalancing[Session, rebalance, and heartbeat settings]
  Config --> Fetching[Fetch size and wait-time settings]
  Group --> Kafka[Kafka Consumer Group]
  Assignment --> Kafka
  Rebalancing --> Kafka
  Fetching --> Kafka
```

Usage

```
import type { ConsumerConfig } from './kafka.interface';

const consumerConfig: ConsumerConfig = {
  groupId: 'billing-service',
  partitionAssigners: [],
  metadataMaxAge: 300_000,
  sessionTimeout: 30_000,
  rebalanceTimeout: 60_000,
  heartbeatInterval: 3_000,
  maxBytesPerPartition: 1_048_576,
  minBytes: 1,
  maxBytes: 10_485_760,
  maxWaitTimeInMs: 5_000,
};

// Pass consumerConfig to the Kafka transport/client initialization.
```

AI Coding Instructions

- Use a stable, service-specific `groupId` ; consumers sharing a group ID load-balance partitions.

- Keep `heartbeatInterval` meaningfully lower than `sessionTimeout` to prevent unintended consumer removal.
- Ensure `maxBytes` is large enough for the expected aggregate batch size and is compatible with broker fetch limits.
- Configure `partitionAssigners` consistently across consumers in the same group to avoid rebalance compatibility issues.
- Treat timeout and fetch-size changes as operational tuning values; validate them against message volume and processing duration.

How it works

`ConsumerConfig` is an exported TypeScript interface representing the KafkaJS consumer-creation options in this repository's Kafka type surface; the file explicitly states that it contains KafkaJS types only, not NestJS logic.

`packages/microservices/external/kafka.interface.ts:1-8` A `Kafka` instance accepts this type as the required argument to `consumer()` and returns a `Consumer`.

`packages/microservices/external/kafka.interface.ts:19-24`