

ITopicConfig

August 17, 2026

ITopicConfig

Kind: Interface

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

Part of: [Microservices](#)

`ITopicConfig` defines the configuration required to create or manage a Kafka topic in the microservices integration layer. It specifies the topic name, partitioning and replication settings, optional replica placement, and broker-level topic configuration entries.

Properties

Property	Type
<code>topic</code>	<code>string</code>
<code>numPartitions</code>	<code>number</code>
<code>replicationFactor</code>	<code>number</code>
<code>replicaAssignment</code>	<code>object[]</code>
<code>configEntries</code>	<code>IResourceConfigEntry[]</code>

Diagram

```
graph LR
  A[ITopicConfig] --> B["topic: string"]
  A --> C["numPartitions: number"]
  A --> D["replicationFactor: number"]
  A --> E["replicaAssignment: object[]"]
  A --> F["configEntries: IResourceConfigEntry[]"]

  C --> G["Kafka partitions"]
  D --> H["Replica count"]
```

```
E --> I[Explicit broker assignment]
F --> J[Topic-level broker settings]
```

Usage

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

const topicConfig: ITopicConfig = {
  topic: 'orders.created',
  numPartitions: 3,
  replicationFactor: 2,
  replicaAssignment: [],
  configEntries: [
    {
      name: 'retention.ms',
      value: '604800000', // Keep messages for 7 days
    },
    {
      name: 'cleanup.policy',
      value: 'delete',
    },
  ],
};

// Pass to the Kafka topic administration or provisioning service.
await kafkaAdmin.createTopic(topicConfig);
```

AI Coding Instructions

- Provide a unique, descriptive `topic` name that follows the application's Kafka naming conventions.
- Set `numPartitions` based on expected consumer parallelism and throughput; changing partition counts later can affect message ordering assumptions.
- Ensure `replicationFactor` does not exceed the number of available Kafka brokers.
- Use `replicaAssignment` only when explicit partition-to-broker placement is required; otherwise provide an empty array or use the project's standard default.
- Add topic-specific retention, cleanup, and compaction settings through `configEntries` using valid Kafka broker configuration keys.