

IResourceConfig

August 18, 2026

IResourceConfig

Kind: Interface

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

Part of: [Microservices](#)

`IResourceConfig` defines the configuration payload for a Kafka resource, such as a topic, broker, or other configurable Kafka entity. It combines the resource type and name with a collection of individual configuration entries that should be applied or managed by the external Kafka integration.

Properties

Property	Type
<code>type</code>	<code>ConfigResourceTypes</code>
<code>name</code>	<code>string</code>
<code>configEntries</code>	<code>IResourceConfigEntry[]</code>

Diagram

```
graph LR
  A[IResourceConfig] --> B[type: ConfigResourceTypes]
  A --> C[name: string]
  A --> D[configEntries: IResourceConfigEntry[]]
  D --> E[Kafka resource configuration values]
```

Usage

```
import type {
  IResourceConfig,
  ConfigResourceTypes,
```

```

} from './kafka.interface';

const topicConfig: IResourceConfig = {
  type: ConfigResourceTypes.TOPIC,
  name: 'orders.events',
  configEntries: [
    {
      name: 'retention.ms',
      value: '604800000',
    },
    {
      name: 'cleanup.policy',
      value: 'delete',
    },
  ],
};

// Pass the resource configuration to the Kafka administration layer.
await kafkaAdmin.updateResourceConfig(topicConfig);

```

AI Coding Instructions

- Set `type` to the Kafka resource category represented by `ConfigResourceTypes` ; ensure it matches the target resource being configured.
- Use the exact Kafka resource identifier in `name` , such as a topic name, because configuration updates are applied to that named resource.
- Populate `configEntries` with valid `IResourceConfigEntry` objects and Kafka-supported configuration keys.
- Keep configuration values in the expected serialized format, typically strings, to match Kafka Admin API conventions.
- Route `IResourceConfig` objects through the existing Kafka administration or external microservice integration rather than applying configuration changes directly.