

ClientKafkaProxy

August 17, 2026

ClientKafkaProxy

Kind: Interface

Source: `packages/microservices/interfaces/client-kafka-proxy.interface.ts`

Part of: [Microservices](#)

`ClientKafkaProxy` defines the Kafka client resources managed by a microservice client proxy. It exposes nullable `consumer` and `producer` references so the proxy can create, reuse, and clean up Kafka connections during its lifecycle.

Properties

Property	Type
<code>consumer</code>	<code>`Consumer</code>
<code>producer</code>	<code>`Producer</code>

Diagram

```
graph LR
  Proxy[Kafka Client Proxy] --> Interface[ClientKafkaProxy]
  Interface --> Consumer[Consumer | null]
  Interface --> Producer[Producer | null]
  Consumer --> KafkaBroker[Kafka Broker]
  Producer --> Kafka
```

Usage

```
import type { Consumer, Producer } from 'kafajs';
import type { ClientKafkaProxy } from './client-kafka-proxy.interface';

class KafkaClient implements ClientKafkaProxy {
  consumer: Consumer | null = null;
```

```

producer: Producer | null = null;

async publish(topic: string, message: unknown) {
  if (!this.producer) {
    throw new Error('Kafka producer has not been initialized.');
```

```

  }

  await this.producer.send({
    topic,
    messages: [{ value: JSON.stringify(message) }],
  });
}

async close() {
  await this.consumer?.disconnect();
  await this.producer?.disconnect();

  this.consumer = null;
  this.producer = null;
}
}

```

AI Coding Instructions

- Treat `consumer` and `producer` as optional lifecycle-managed resources; always handle the `null` case before calling Kafka methods.
- Initialize Kafka clients before subscribing, consuming, or publishing messages, and reset references to `null` after disconnecting.
- Use the KafkaJS `Consumer` and `Producer` types when implementing this interface.
- Ensure shutdown logic disconnects both resources safely, using optional chaining or explicit null checks.