

Kafka

August 18, 2026

Kafka

Kind: Class

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

Part of: [Microservices](#)

`Kafka` provides access to the Kafka client components used by the microservices layer: producers, consumers, administrative operations, and logging. It acts as a central integration point for publishing messages, subscribing to topics, managing Kafka resources, and reporting Kafka-related activity.

Methods

Method	Signature	Returns
<code>producer</code>	<code>producer(config: ProducerConfig)</code>	<code>Producer</code>
<code>consumer</code>	<code>consumer(config: ConsumerConfig)</code>	<code>Consumer</code>
<code>admin</code>	<code>admin(config: AdminConfig)</code>	<code>Admin</code>
<code>logger</code>	<code>logger()</code>	<code>Logger</code>

Diagram

```
graph LR
  App[Microservice] --> Kafka[Kafka]
  Kafka --> Producer[Producer]
  Kafka --> Consumer[Consumer]
  Kafka --> Admin[Admin]
  Kafka --> Logger[Logger]

  Producer --> Broker[Broker]
  Consumer --> Broker
  Admin --> Broker
  Logger --> Logs[Application Logs]
```

Usage

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

async function publishUserCreated(kafka: Kafka) {
  const producer = kafka.producer();

  await producer.connect();

  await producer.send({
    topic: 'users.created',
    messages: [
      {
        key: 'user-123',
        value: JSON.stringify({
          id: 'user-123',
          email: 'user@example.com',
        }),
      },
    ],
  });

  kafka.logger().info('Published users.created event', {
    userId: 'user-123',
  });

  await producer.disconnect();
}
```

AI Coding Instructions

- Access Kafka clients through `producer()`, `consumer()`, and `admin()` instead of creating KafkaJS clients directly.
- Connect and disconnect producers or consumers according to the surrounding application lifecycle; avoid creating a new connection for every message when a shared client is available.
- Serialize message payloads consistently, typically with `JSON.stringify`, and use stable message keys when partition ordering matters.
- Use `logger()` for Kafka-related errors, retries, and lifecycle events rather than writing directly to console output.
- Use `admin()` only for broker or topic management operations, not for normal message publishing or consumption.