

KafkaJSErrorMetadata

August 18, 2026

KafkaJSErrorMetadata

Kind: Interface

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

Part of: [Microservices](#)

`KafkaJSErrorMetadata` describes additional context attached to KafkaJS-related errors in the microservices Kafka transport. It identifies whether an operation can be retried and provides the affected topic, partition, and Kafka partition metadata needed for diagnostics or recovery logic.

Properties

Property	Type
<code>retriable</code>	<code>boolean</code>
<code>topic</code>	<code>string</code>
<code>partitionId</code>	<code>number</code>
<code>metadata</code>	<code>PartitionMetadata</code>

Diagram

```
graph LR
    Error[KafkaJS Error] --> Metadata[KafkaJSErrorMetadata]
    Metadata --> Retriable[Retriable[retriable: boolean]]
    Metadata --> Topic[Topic[topic: string]]
    Metadata --> Partition[Partition[partitionId: number]]
    Metadata --> PartitionMetadata[PartitionMetadata[metadata: PartitionMetadata]]
    Retriable --> RetryLogic[RetryLogic[Retry / Failure Handling]]
    Topic --> Diagnostics[Logging and Diagnostics]
    Partition --> Diagnostics
```

Usage

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

function handleKafkaError(error: Error & Partial<KafkaJSErrorMessageMetadata>) {
  if (!error.topic || error.partitionId === undefined) {
    throw error;
  }

  const context: KafkaJSErrorMessageMetadata = {
    retrieable: error.retrieable ?? false,
    topic: error.topic,
    partitionId: error.partitionId,
    metadata: error.metadata!,
  };

  console.error(
    `Kafka error on ${context.topic}[${context.partitionId}]`,
    context.metadata,
  );

  if (context.retrieable) {
    // Retry the Kafka operation using the topic and partition context.
    return;
  }

  throw error;
}
```

AI Coding Instructions

- Preserve the KafkaJS naming and metadata types when mapping Kafka errors into application-level error handling.
- Check `retrieable` before scheduling retries; do not retry non-retrieable errors automatically.
- Use `topic` and `partitionId` in logs, metrics, and tracing to make partition-specific failures diagnosable.
- Treat `metadata` as Kafka partition context and avoid replacing it with partial or unrelated metadata objects.