KafkaJSServerDoesNotSupportApiKeyMetadata

August 17, 2026

KafkaJSServerDoesNotSupportApiKeyMetadata

Kind: Interface

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

Part of: Microservices

`KafkaJSServerDoesNotSupportApiKeyMetadata` describes a Kafka API operation that is not supported by the connected KafkaJS server or broker. It pairs the numeric Kafka API key with its human-readable API name, enabling capability checks, diagnostics, and compatibility handling.

Properties

Property	Type
`apiKey`	`number`
`apiName`	`string`

Diagram

```
graph LR
  Client[KafkaJS Client] --> CapabilityCheck[Broker API Capability Check]
  CapabilityCheck --> Unsupported[KafkaJSServerDoesNotSupportApiKeyMetadata]
  Unsupported --> ApiKey[apiKey: number]
  Unsupported --> ApiName[apiName: string]
  Unsupported --> Handling[Compatibility Handling / Error Reporting]
```

Usage

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

function formatUnsupportedApi(
  metadata: KafkaJSServerDoesNotSupportApiKeyMetadata,
```

```
): string {
  return `Kafka broker does not support ${metadata.apiName} (API key ${metadata.apiKey}).`;
}

const unsupportedApi: KafkaJSServerDoesNotSupportApiKeyMetadata = {
  apiKey: 68,
  apiName: 'ConsumerGroupHeartbeat',
};

console.warn(formatUnsupportedApi(unsupportedApi));
```

AI Coding Instructions

- Use this interface only for Kafka APIs confirmed as unsupported by the connected broker or KafkaJS server.
- Preserve both `apiKey` and `apiName`; the numeric key supports protocol-level handling, while the name improves logs and error messages.
- Keep `apiKey` aligned with Kafka protocol API key values rather than application-specific identifiers.
- Use the metadata when building compatibility errors, fallback behavior, or broker capability diagnostics.