

KafkaJSDeleteGroupsErrorGroups

August 18, 2026

KafkaJSDeleteGroupsErrorGroups

Kind: Interface

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

Part of: [Microservices](#)

`KafkaJSDeleteGroupsErrorGroups` represents an error returned for a specific Kafka consumer group during a delete-groups operation. It identifies the affected group and provides both the Kafka protocol error code and the associated `KafkaJSError` details for handling or logging failures.

Properties

Property	Type
<code>groupId</code>	<code>string</code>
<code>errorCode</code>	<code>number</code>
<code>error</code>	<code>KafkaJSError</code>

Diagram

```
graph LR
  DeleteGroups[Kafka delete groups response] --> GroupError[KafkaJSDeleteGroupsErrorGroups]
  GroupError --> GroupId[groupId: string]
  GroupError --> ErrorCode[errorCode: number]
  GroupError --> Error[error: KafkaJSError]
```

Usage

```
import { KafkaJSError } from 'kafkajs';
import { KafkaJSDeleteGroupsErrorGroups } from './kafka.interface';
```

```
function handleDeleteGroupError(
  groupError: KafkaJSDeleteGroupsErrorGroups,
): void {
  console.error(
    `Failed to delete consumer group "${groupError.groupId}" ` +
    `(Kafka error code: ${groupError.errorCode})`,
    groupError.error,
  );

  if (groupError.error instanceof KafkaJSError) {
    // Retry, alert, or map the Kafka error to an application-level response.
  }
}
```

AI Coding Instructions

- Treat each interface instance as the failure result for one consumer group, even when a delete operation targets multiple groups.
- Use `groupId` when logging or reporting failures so operators can identify the affected consumer group.
- Preserve both `errorCode` and `error`; the numeric code supports protocol-level handling, while `KafkaJSError` contains diagnostic context.
- Handle delete-group failures individually rather than assuming the entire batch operation succeeded or failed uniformly.