

DescribeConfigResponse

August 18, 2026

DescribeConfigResponse

Kind: Interface

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

Part of: [Microservices](#)

`DescribeConfigResponse` represents the response returned after requesting Kafka configuration details for one or more resources. It groups configuration entries, resource-level status information, and broker throttling metadata so callers can inspect successful and failed configuration lookups.

Properties

Property	Type
<code>resources</code>	<code>{ configEntries: ConfigEntries[]; errorCode: number; errorMessage: string; resourceName: string; resourceType: ConfigResourceTypes; }[]</code>
<code>throttleTime</code>	<code>number</code>

Diagram

```
graph LR
  Client[Kafka client] --> Request[Describe config request]
  Request --> Broker[Kafka broker]
  Broker --> Response[DescribeConfigResponse]

  Response --> Throttle[throttleTime]
  Response --> Resources[resources[]]
  Resources --> ResourceName[resourceName]
  Resources --> ResourceType[resourceType]
  Resources --> Entries[configEntries: ConfigEntries[]]
  Resources --> ErrorCode[errorCode]
  Resources --> ErrorMessage[errorMessage]
```

Usage

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

function logConfigResponse(response: DescribeConfigResponse): void {
  console.log(`Broker throttle time: ${response.throttleTime}ms`);

  for (const resource of response.resources) {
    if (resource.errorCode !== 0) {
      console.error(
        `Unable to describe ${resource.resourceName}: ${resource.errorMessage}`,
      );
      continue;
    }

    console.log(
      `Configuration for ${resource.resourceType}:${resource.resourceName}`,
    );

    for (const entry of resource.configEntries) {
      console.log(`${entry.configName}=${entry.configValue}`);
    }
  }
}
```

AI Coding Instructions

- Treat `resources` as a multi-resource response; validate each resource's `errorCode` instead of assuming the entire request succeeded.
- Check `errorCode` before reading or acting on `configEntries`, and surface `errorMessage` when a resource lookup fails.
- Preserve `resourceType` and `resourceName` together when mapping results to Kafka topics, brokers, or other configurable resources.
- Use `throttleTime` for observability and retry/backoff decisions when Kafka brokers are under load.

How it works

`DescribeConfigResponse` is an exported TypeScript interface representing the result type of `Admin.describeConfigs(...)`. The declaration file is intended only to mirror KafkaJS package types, rather than contain NestJS logic.

[packages/microservices/external/kafka.interface.ts:1-8](#)

[packages/microservices/external/kafka.interface.ts:548-551](#)

- `resources` is an array with one object per described resource. Each object contains:
 - `resourceName: string` and `resourceType: ConfigResourceTypes`, identifying the resource. `ConfigResourceTypes` permits `UNKNOWN ( 0 )`, `TOPIC ( 2 )`, `BROKER ( 4 )`, and `BROKER_LOGGER ( 8 )`.
[packages/microservices/external/kafka.interface.ts:365-372](#)
[packages/microservices/external/kafka.interface.ts:295-300](#)
 - `errorCode: number` and `errorMessage: string`.
[packages/microservices/external/kafka.interface.ts:366-372](#)
 - `configEntries: ConfigEntries[]`. Each entry has a configuration name and value, default, sensitivity, and read-only flags, a `ConfigSource`, and an array of synonym entries. [packages/microservices/external/kafka.interface.ts:349-357](#)
- Each synonym records `configName`, `configValue`, and `configSource`.
[packages/microservices/external/kafka.interface.ts:359-363](#) `ConfigSource` has declared values for unknown, topic, dynamic broker, dynamic default broker, static broker, default, and dynamic broker-logger sources.
[packages/microservices/external/kafka.interface.ts:302-310](#)
- `throttleTime: number` is a top-level field of the response.
[packages/microservices/external/kafka.interface.ts:365-374](#)

`Admin.describeConfigs` requires an object containing `resources: ResourceConfigQuery[]` and `includeSynonyms: boolean`, and its declared return type is `Promise<DescribeConfigResponse>`. A resource query requires a resource `type` and `name`, with optional `configNames`. [packages/microservices/external/kafka.interface.ts:343-347](#)
[packages/microservices/external/kafka.interface.ts:548-551](#)

This interface declares data shape only; it contains no executable validation, thrown errors, or side effects. [packages/microservices/external/kafka.interface.ts:365-374](#)