ResourceConfigQuery

August 17, 2026

ResourceConfigQuery

Kind: Interface

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

Part of: Microservices

`ResourceConfigQuery` defines a request for retrieving configuration values from a Kafka resource. It identifies the Kafka resource type and name, then specifies which configuration keys should be returned.

Properties

Property	Type
`type`	`ConfigResourceTypes`
`name`	`string`
`configNames`	`string[]`

Diagram

```
graph LR
  Query[ResourceConfigQuery] --> Type[type: ConfigResourceTypes]
  Query --> Name[name: string]
  Query --> ConfigNames[configNames: string[]]

  Type --> Resource[Kafka resource type]
  Name --> Target[Target resource name]
  ConfigNames --> Keys[Requested configuration keys]
```

Usage

```ts
import type { ResourceConfigQuery } from './kafka.interface';
import { ConfigResourceTypes } from './kafka.interface';

const topicConfigQuery: ResourceConfigQuery = {
  type: ConfigResourceTypes.TOPIC,
  name: 'orders.created',
  configNames: [
    'retention.ms',
    'cleanup.policy',
    'min.insync.replicas',
  ],
};

// Pass the query to the Kafka admin/configuration service.
const configs = await kafkaConfigService.describeResourceConfig(topicConfigQuery);
```

AI Coding Instructions

- Set `type` to the matching `ConfigResourceTypes` value for the target Kafka resource, such as a topic or broker.
- Provide the exact Kafka resource identifier in `name`; topic names and broker identifiers must match the cluster configuration.
- Include only the configuration keys needed in `configNames` to avoid unnecessary admin API requests.
- Keep this interface aligned with the Kafka admin client integration that resolves resource configurations.
- Validate or handle missing configuration keys when consuming the returned configuration response.

How it works

`ResourceConfigQuery` is an exported TypeScript interface in a file declared to represent KafkaJS package types only; it contains no executable implementation.
packages/microservices/external/kafka.interface.ts:1-8
packages/microservices/external/kafka.interface.ts:343-347

- It describes one resource entry for `Admin.describeConfigs`. That method requires a `resources` array of these entries and an `includeSynonyms` boolean, then returns `Promise<DescribeConfigResponse>`.
  packages/microservices/external/kafka.interface.ts:502-503
  packages/microservices/external/kafka.interface.ts:548-551

- Each query must contain:
  - `type`: a `ConfigResourceTypes` enum value. The declared values are `UNKNOWN` (`0`), `TOPIC` (`2`), `BROKER` (`4`), and `BROKER_LOGGER` (`8`). packages/microservices/external/kafka.interface.ts:295-300 packages/microservices/external/kafka.interface.ts:343-345
  - `name`: a string. packages/microservices/external/kafka.interface.ts:343-346
- It may contain `configNames`, an optional string array. packages/microservices/external/kafka.interface.ts:343-347
- The corresponding response contains resource results with `configEntries`, an `errorCode`, an `errorMessage`, a resource name and type, plus a `throttleTime`. Individual entries include their name, value, source, default/sensitive/read-only flags, and synonyms. packages/microservices/external/kafka.interface.ts:349-374
- No runtime validation, error throwing, network action, or other side effect is implemented by `ResourceConfigQuery` itself in this file; it is only a structural type declaration. packages/microservices/external/kafka.interface.ts:1-8 packages/microservices/external/kafka.interface.ts:343-347