

ConfigEntries

August 17, 2026

ConfigEntries

Kind: Interface

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

Part of: [Microservices](#)

`ConfigEntries` represents a single Kafka configuration entry returned or managed through the microservices Kafka integration. It captures the configuration value, its origin, mutability, sensitivity, default status, and any equivalent configuration synonyms.

Properties

Property	Type
<code>configName</code>	<code>string</code>
<code>configValue</code>	<code>string</code>
<code>isDefault</code>	<code>boolean</code>
<code>configSource</code>	<code>ConfigSource</code>
<code>isSensitive</code>	<code>boolean</code>
<code>readOnly</code>	<code>boolean</code>
<code>configSynonyms</code>	<code>ConfigSynonyms[]</code>

Diagram

```
graph LR
  ConfigEntries[ConfigEntries]
  ConfigEntries --> Name[configName: string]
  ConfigEntries --> Value[configValue: string]
  ConfigEntries --> Source[configSource: ConfigSource]
  ConfigEntries --> Default[isDefault: boolean]
```

```
ConfigEntries --> Sensitive[isSensitive: boolean]
ConfigEntries --> ReadOnly[readOnly: boolean]
ConfigEntries --> Synonyms[configSynonyms: ConfigSynonyms[]]
```

Usage

```
import type {
  ConfigEntries,
  ConfigSource,
  ConfigSynonyms,
} from './kafka.interface';

const retentionConfig: ConfigEntries = {
  configName: 'retention.ms',
  configValue: '604800000',
  isDefault: false,
  configSource: ConfigSource.DYNAMIC_TOPIC_CONFIG,
  isSensitive: false,
  readOnly: false,
  configSynonyms: [
    {
      configName: 'retention.ms',
      configValue: '604800000',
      configSource: ConfigSource.DYNAMIC_TOPIC_CONFIG,
    } as ConfigSynonyms,
  ],
};

if (!retentionConfig.readOnly && !retentionConfig.isSensitive) {
  console.log(
    `${retentionConfig.configName} is set to ${retentionConfig.configValue}`,
  );
}
```

AI Coding Instructions

- Preserve Kafka-provided metadata such as `configSource`, `isDefault`, and `readOnly` when mapping admin-client configuration responses.
- Do not log or expose `configValue` when `isSensitive` is `true`.
- Check `readOnly` before attempting to update a configuration entry through Kafka Admin APIs.
- Treat `configSynonyms` as alternate or inherited values; use the primary `configValue` as the effective configuration value.

- Use the corresponding `ConfigSource` and `ConfigSynonyms` types from `kafka.interface.ts` rather than replacing them with loose strings or objects.