

PartitionOffset

August 17, 2026

PartitionOffset

Kind: Interface

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

Part of: [Microservices](#)

`PartitionOffset` represents a Kafka consumer position within a specific topic partition. It pairs a numeric partition identifier with a string offset, allowing Kafka-related integrations to track, commit, or seek to an exact message position.

Properties

Property	Type
<code>partition</code>	<code>number</code>
<code>offset</code>	<code>string</code>

Diagram

```
graph LR
  Consumer[Kafka Consumer] --> PartitionOffset
  PartitionOffset --> Partition[partition: number]
  PartitionOffset --> Offset[offset: string]
  PartitionOffset --> Commit[Commit or seek consumer position]
```

Usage

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

const checkpoint: PartitionOffset = {
  partition: 2,
  offset: '1542',
};
```

```
// Example: use the position when committing processed messages.
await consumer.commitOffsets([
  {
    topic: 'orders',
    partition: checkpoint.partition,
    offset: checkpoint.offset,
  },
]);
```

AI Coding Instructions

- Keep `partition` as a zero-based numeric Kafka partition index.
- Store `offset` as a string; Kafka client libraries commonly use string offsets to avoid integer precision issues.
- Use this interface when passing partition-specific checkpoints between consumer, retry, seek, or commit logic.
- Do not treat the offset as the currently processed message unless the surrounding Kafka client API explicitly defines it that way; commit semantics often expect the next offset to consume.

How it works

`PartitionOffset` is an exported TypeScript interface representing an offset for one Kafka partition. It has two required properties:

- `partition` : a `number` identifying the partition.
[packages/microservices/external/kafka.interface.ts:771-773]
- `offset` : a `string` representing that partition's offset.
[packages/microservices/external/kafka.interface.ts:771-774]

It is a type-only declaration in a file intended to represent KafkaJS package types, not NestJS logic. [packages/microservices/external/kafka.interface.ts:1-8] The interface contains no runtime implementation, validation, error handling, or side effects.
[packages/microservices/external/kafka.interface.ts:771-774]

The type is used in several offset-oriented shapes:

- `TopicOffsets.partitions` is an array of `PartitionOffset` values, grouped under a topic string. [packages/microservices/external/kafka.interface.ts:776-783]
- `SeekEntry` is an alias of `PartitionOffset`.
[packages/microservices/external/kafka.interface.ts:438]

- `FetchOffsetsPartition` extends it with required `metadata: string | null` .
[packages/microservices/external/kafka.interface.ts:440-442]
- `Admin.fetchTopicOffsets()` returns entries containing these fields plus `high` and `low` string fields; `fetchTopicOffsetsByTimestamp()` returns `SeekEntry` values.
[packages/microservices/external/kafka.interface.ts:526-532]
- `Admin.setOffsets()` and `Admin.deleteTopicRecords()` accept arrays of `SeekEntry` values in their `partitions` options.
[packages/microservices/external/kafka.interface.ts:538-542]
[packages/microservices/external/kafka.interface.ts:562-565]