ISocketFactoryArgs

August 17, 2026

ISocketFactoryArgs

Kind: Interface

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

Part of: Microservices

`ISocketFactoryArgs` defines the connection parameters provided when creating a Kafka socket. It groups the broker address, TLS configuration, and connection lifecycle callback required by the microservices Kafka transport.

Properties

Property	Type
`host`	`string`
`port`	`number`
`ssl`	`tls.ConnectionOptions`
`onConnect`	`() => void`

Diagram

```
graph LR
  SocketFactory[Kafka Socket Factory] --> Args[ISocketFactoryArgs]
  Args --> Host[host: string]
  Args --> Port[port: number]
  Args --> SSL[ssl: tls.ConnectionOptions]
  Args --> OnConnect[onConnect: () => void]
  SSL --> TLS[TLS-secured broker connection]
  OnConnect --> Ready[Connection established]
```

Usage

```
import type { ConnectionOptions } from 'node:tls';

interface ISocketFactoryArgs {
  host: string;
  port: number;
  ssl: ConnectionOptions;
  onConnect: () => void;
}

const socketArgs: ISocketFactoryArgs = {
  host: 'kafka.example.com',
  port: 9093,
  ssl: {
    rejectUnauthorized: true,
    // ca, cert, and key can be provided when required by the broker.
  },
  onConnect: () => {
    console.log('Connected to Kafka broker');
  },
};

// Pass socketArgs to the Kafka transport socket factory.
```

AI Coding Instructions

- Provide a valid Kafka broker hostname and port; use the TLS listener port when `ssl` is configured.
- Pass `tls.ConnectionOptions` through unchanged so certificate, key, CA, and verification settings remain available to the underlying socket.
- Use `onConnect` only for post-connection work, such as logging or readiness updates; avoid blocking operations.
- Do not place connection error handling in `onConnect`; handle socket and transport errors through the relevant Kafka client hooks.