

SocketsContainer

August 18, 2026

SocketsContainer

Kind: Class

Source: `packages/websockets/sockets-container.ts`

Part of: [Websockets](#)

`SocketsContainer` manages the collection of `ServerAndEventStreamsHost` instances used by the WebSocket layer. It provides lookup, registration, enumeration, and cleanup operations so socket server hosts can be reused and managed consistently during application lifecycle events.

Methods

Method	Signature	Returns
<code>getAll</code>	<code>getAll()</code>	<code>`Map<string</code>
<code>getOneByConfig</code>	<code>getOneByConfig(options: T)</code>	<code>ServerAndEventStreamsHost</code>
<code>addOne</code>	<code>addOne(options: T, host: ServerAndEventStreamsHost)</code>	<code>void</code>
<code>clear</code>	<code>clear()</code>	<code>void</code>

Diagram

```
graph LR
  A[WebSocket Configuration] --> B[SocketsContainer]
  B -->|addOne| C[ServerAndEventStreamsHost]
  B -->|getOneByConfig| C
  B -->|getAll| D[Map of Socket Hosts]
  B -->|clear| E[Release Registered Hosts]
```

Usage

```
import { SocketsContainer } from './sockets-container';
import { ServerAndEventStreamsHost } from './server-and-event-streams-host';

const socketsContainer = new SocketsContainer();

// Create or obtain a configured host from the WebSocket bootstrap flow.
const host = new ServerAndEventStreamsHost(/* websocket adapter */);

// Register the host using the socket server configuration.
socketsContainer.addOne(
  { port: 3000, path: '/socket.io' },
  host,
);

// Reuse the registered host when processing the same configuration.
const registeredHost = socketsContainer.getOneByConfig({
  port: 3000,
  path: '/socket.io',
});

// Inspect all active socket hosts.
console.log(socketsContainer.getAll());

// Clear registrations during application shutdown or test cleanup.
socketsContainer.clear();
```

AI Coding Instructions

- Register hosts through `addOne()` rather than mutating the map returned by `getAll()`.
- Use the same socket configuration values when calling `addOne()` and `getOneByConfig()` so host reuse works correctly.
- Treat `SocketsContainer` as application infrastructure; create and populate it during WebSocket server bootstrap.
- Call `clear()` during teardown to prevent socket host state from leaking between application instances or tests.

How it works

SocketsContainer

`SocketsContainer` is an in-memory collection of `ServerAndEventStreamsHost` objects, indexed by a hash derived from gateway configuration objects. Its backing collection is a `Map<string | RegExp, ServerAndEventStreamsHost>`, although its internal key-generation

method returns a `string` . [packages/websockets/sockets-container.ts:4-8]
[packages/websockets/sockets-container.ts:33-37]

A stored host has a `server` plus `init` , `connection` , and `disconnect` RxJS subjects.
[packages/websockets/interfaces/server-and-event-streams-host.interface.ts:6-11]