

Version: 4.x

Client Initialization

Once you have [installed](#) the Socket.IO client library, you can now init the client. The complete list of options can be found [here](#).



TIP

For TypeScript users, it is possible to provide type hints for the events. Please check [this](#).

In the examples below, the `io` object comes either from:

- the `<script>` import

```
<script src="/socket.io/socket.io.js"></script>
```

- an ESM import

```
<script type="module">
  import { io } from "https://cdn.socket.io/4.8.1/socket.io.esm.min.js";
</script>
```

- NPM

[CommonJS](#) [ES modules](#) [TypeScript](#)

```
const { io } = require("socket.io-client");
```

From the same domain

If your front is served on the same domain as your server, you can simply use:

```
const socket = io();
```

The server URL will be deduced from the `window.location` object.

From a different domain

In case your front is not served from the same domain as your server, you have to pass the URL of your server.

```
const socket = io("https://server-domain.com");
```

In that case, please make sure to enable **Cross-Origin Resource Sharing (CORS)** on the server.

INFO

You can use either `https` or `wss` (respectively, `http` or `ws`).

```
// the following forms are similar
const socket = io("https://server-domain.com");
const socket = io("wss://server-domain.com");
const socket = io("server-domain.com"); // only in the browser when the page
is served over https (will not work in Node.js)
```

Custom namespace

In the examples above, the client will connect to the main namespace. Using only the main namespace should be sufficient for most use cases, but you can specify the namespace with:

```
// same origin version
const socket = io("/admin");
// cross origin version
const socket = io("https://server-domain.com/admin");
```

You can find more details about namespaces [here](#).

Options

The complete list of available options can be found [here](#).

 [Edit this page](#)

Last updated on Aug 22, 2025