[**socket.io**](http://socket.io/)

[#](http://socket.io/docs/server-api/#server-api)Server API

[#](http://socket.io/docs/server-api/#server)Server

Exposed by require('socket.io').

[#](http://socket.io/docs/server-api/#server())Server()

Creates a new Server. Works with and without new:

var io = require('socket.io')();

// or

var Server = require('socket.io');

var io = new Server();

[#](http://socket.io/docs/server-api/#server(opts:object))Server(opts:Object)

Optionally, the first or second argument (see below) of the Server  
constructor can be an options object.

The following options are supported:

* serveClient sets the value for Server#serveClient()
* path sets the value for Server#path()

The same options passed to socket.io are always passed to  
the engine.io Server that gets created. See engine.io  
[options](https://github.com/learnboost/engine.io#methods-1)  
as reference.

[#](http://socket.io/docs/server-api/#server(srv:http#server,-opts:object))Server(srv:http#Server, opts:Object)

**Creates a new Server and attaches it to the given srv**. Optionally  
opts can be passed.

[#](http://socket.io/docs/server-api/#server(port:number,-opts:object))Server(port:Number, opts:Object)

Binds socket.io to a new http.Server that listens on port.

[#](http://socket.io/docs/server-api/#server#serveclient(v:boolean):server)Server#serveClient(v:Boolean):Server

If v is true the attached server (see Server#attach) will serve  
the client files. Defaults to true.

**This method has no effect after attach is called.**

// pass a server and the `serveClient` option

var io = require('socket.io')(http, { serveClient: false });

// or pass no server and then you can call the method

var io = require('socket.io')();

io.serveClient(false);

io.attach(http);

If no arguments are supplied this method returns the current value.

[#](http://socket.io/docs/server-api/#server#path(v:string):server)Server#path(v:String):Server

Sets the path v under which engine.io and the static files will be  
served. Defaults to /socket.io.

If no arguments are supplied this method returns the current value.

[#](http://socket.io/docs/server-api/#server#adapter(v:adapter):server)Server#adapter(v:Adapter):Server

Sets the adapter v. Defaults to an instance of the Adapter that  
ships with socket.io which is memory based. See  
[socket.io-adapter](https://github.com/learnboost/socket.io-adapter).

If no arguments are supplied this method returns the current value.

[#](http://socket.io/docs/server-api/#server#origins(v:string):server)Server#origins(v:String):Server

Sets the allowed origins v. Defaults to any origins being allowed.

If no arguments are supplied this method returns the current value.

[#](http://socket.io/docs/server-api/#server#sockets:namespace)Server#sockets:Namespace

The default (/) namespace.

[#](http://socket.io/docs/server-api/#server#attach(srv:http#server,-opts:object):server)Server#attach(srv:http#Server, opts:Object):Server

Attaches the Server to an engine.io instance on srv with the  
supplied opts (optionally).

[#](http://socket.io/docs/server-api/#server#attach(port:number,-opts:object):server)Server#attach(port:Number, opts:Object):Server

Attaches the Server to an engine.io instance that is bound to port  
with the given opts (optionally).

[#](http://socket.io/docs/server-api/#server#listen)Server#listen

Synonym of Server#attach.

[#](http://socket.io/docs/server-api/#server#bind(srv:engine#server):server)Server#bind(srv:engine#Server):Server

Advanced use only. Binds the server to a specific engine.io Server  
(or compatible API) instance.

[#](http://socket.io/docs/server-api/#server#onconnection(socket:engine#socket):server)Server#onconnection(socket:engine#Socket):Server

Advanced use only. Creates a new socket.io client from the incoming  
engine.io (or compatible API) socket.

[#](http://socket.io/docs/server-api/#server#of(nsp:string):namespace)Server#of(nsp:String):Namespace

Initializes and retrieves the given Namespace by its pathname  
identifier nsp.

If the namespace was already initialized it returns it right away.

[#](http://socket.io/docs/server-api/#server#emit)Server#emit

Emits an event to **all connected clients.** The following two are equivalent:

var io = require('socket.io')();

io.sockets.emit('an event sent to all connected clients');

io.emit('an event sent to all connected clients');

For other available methods, see Namespace below.

[#](http://socket.io/docs/server-api/#server#use)Server#use

See Namespace#use below.

[#](http://socket.io/docs/server-api/#namespace)Namespace

Represents a pool of **sockets** connected under a given scope identified by a pathname (eg: /chat).

By default the client always connects to /.

Events

* connection / connect. Fired upon a connection.

Parameters:

* + Socket **the incoming socket.**

[#](http://socket.io/docs/server-api/#namespace#name:string)Namespace#name:String

The namespace identifier property.

[#](http://socket.io/docs/server-api/#namespace#connected:object)Namespace#connected:Object

Hash of Socket objects that are connected to this namespace indexed  
by id.

[#](http://socket.io/docs/server-api/#namespace#use(fn:function):namespace)Namespace#use(fn:Function):Namespace

Registers a middleware, which is a function that gets executed for  
every incoming Socket and receives as parameter the socket and a  
function to optionally defer execution to the next registered  
middleware.

var io = require('socket.io')();

io.use(function(socket, next){

if (socket.request.headers.cookie) return next();

next(new Error('Authentication error'));

});

Errors passed to middleware callbacks are sent as special error  
packets to clients.

[#](http://socket.io/docs/server-api/#socket)Socket

A Socket is the fundamental class for interacting with browser  
clients. A Socket belongs to a certain Namespace (by default /)  
and uses an underlying Client to communicate.

[#](http://socket.io/docs/server-api/#socket#rooms:array)Socket#rooms:Array

A list of strings identifying the rooms this socket is in.

[#](http://socket.io/docs/server-api/#socket#client:client)Socket#client:Client

A reference to the underlying Client object.

[#](http://socket.io/docs/server-api/#socket#conn:socket)Socket#conn:Socket

A reference to the underyling Client transport connection (engine.io  
Socket object).

[#](http://socket.io/docs/server-api/#socket#request:request)Socket#request:Request

A getter proxy that returns the reference to the request that  
originated the underlying engine.io Client. Useful for accessing  
request headers such as Cookie or User-Agent.

[#](http://socket.io/docs/server-api/#socket#id:string)Socket#id:String

A unique identifier for the socket session, that comes from the  
underlying Client.

[#](http://socket.io/docs/server-api/#socket#emit(name:string[,-…]):socket)Socket#emit(name:String[, …]):Socket

Emits an event to the socket identified by the string name. Any  
other parameters can be included.

All datastructures are supported, including Buffer. JavaScript  
functions can’t be serialized/deserialized.

var io = require('socket.io')();

io.on('connection', function(socket){

socket.emit('an event', { some: 'data' });

});

[#](http://socket.io/docs/server-api/#socket#join(name:string[,-fn:function]):socket)Socket#join(name:String[, fn:Function]):Socket

Adds the socket to the room, and fires optionally a callback fn  
with err signature (if any).

**The socket is automatically a member of a room identified with its  
session id (see Socket#id).**

The mechanics of joining rooms are handled by the Adapter  
that has been configured (see Server#adapter above), defaulting to  
[socket.io-adapter](https://github.com/socket.io/socket.io-adapter).

[#](http://socket.io/docs/server-api/#socket#leave(name:string[,-fn:function]):socket)Socket#leave(name:String[, fn:Function]):Socket

Removes the socket from room, and fires optionally a callback fn  
with err signature (if any).

**Rooms are left automatically upon disconnection**.

The mechanics of leaving rooms are handled by the Adapter  
that has been configured (see Server#adapter above), defaulting to  
[socket.io-adapter](https://github.com/socket.io/socket.io-adapter).

[#](http://socket.io/docs/server-api/#socket#to(room:string):socket)Socket#to(room:String):Socket

[#](http://socket.io/docs/server-api/#socket#in(room:string):socket)Socket#in(room:String):Socket

Sets a modifier for a subsequent event emission that the event will  
only be *broadcasted* to sockets that have joined the given room.

To emit to multiple rooms, you can call to several times.

var io = require('socket.io')();

io.on('connection', function(socket){

socket.to('others').emit('an event', { some: 'data' });

});

[#](http://socket.io/docs/server-api/#client)Client

The Client class represents an incoming transport (engine.io) connection. A Client can be associated with many multiplexed Socket  
that belong to different Namespaces.

[#](http://socket.io/docs/server-api/#client#conn)Client#conn

A reference to the underlying engine.io Socket connection.

[#](http://socket.io/docs/server-api/#client#request)Client#request

A getter proxy that returns the reference to the request that  
originated the engine.io connection. Useful for accessing  
request headers such as Cookie or User-Agent.

SOCKET.IO IS OPEN-SOURCE (MIT). RUN BY [CONTRIBUTORS](https://github.com/Automattic/socket.io/graphs/contributors).