

```
// sending to sender-client only
socket.emit('message', "this is a test");

// sending to all clients, include sender
io.emit('message', "this is a test");

// sending to all clients except sender
socket.broadcast.emit('message', "this is a test");

// sending to all clients in 'game' room(channel) except sender
socket.broadcast.to('game').emit('message', 'nice game');

// sending to all clients in 'game' room(channel), include sender
io.in('game').emit('message', 'cool game');

// sending to sender client, only if they are in 'game' room(channel)
socket.to('game').emit('message', 'enjoy the game');

// sending to all clients in namespace 'myNamespace', include sender
io.of('myNamespace').emit('message', 'gg');

// sending to individual socketid (server-side)
socket.broadcast.to(socketid).emit('message', 'for your eyes only');

// join to subscribe the socket to a given channel (server-side):
socket.join('some room');

// then simply use to or in (they are the same) when broadcasting or emitting (server-side)
io.to('some room').emit('some event');

// leave to unsubscribe the socket to a given channel (server-side)
socket.leave('some room');
```