

# 10.0websocket

websoketfrwebsocket

frwebsocketsocket.io 2.0

com/fr/web/socketio/SocketIOServerFactory.class

com/fr/third/socketio/SocketIOServer.class

frwebsocketfrwebsocketnamespace"/system"js

frwebsocketclientsclientsfrwebsocket

socket.iohtml

```
<script type="text/javascript"
  src="${fineServletURL}/file?path=/com/fr/web/socketio/dist/socket.io.js&type=plain&parser=plain"></script>
frwebsocketjswebsocket
```

com/fr/web/ui/materials.min.js

```
/**
 * socket
 * @param {*} options socket
 * @param {*} cb
 */
initSocket: function (options, cb) {
  var ports = options.webSocketPort, portIndex = 0, isInit = true;
  var socket = null, callback = BI.isFunction(cb) ? cb : BI.emptyFn;

  function reconnect (v) {
    if (v >= BI.size(ports)) {
      isInit = false;
      callback("fail");
      return;
    }
    socket.io.uri = options.requestUrl + ":" + ports[v] + options.webSocketNameSpace;
    socket.io.connect();
    socket.connect();
  }

  try {
    socket = io.connect(options.requestUrl + ":" + ports[portIndex] + options.webSocketNameSpace,
BI.extend({
      query: options.query,
      path: options.webSocketContextName
    }, options.options));

    // ,
    socket.on("connect_error", function () {
      isInit && reconnect(++portIndex);
    });

    socket.on("connect", function () {
      isInit = false;
      callback("success");
    });
    return socket;
  } catch (e) {
    throw new Error(e);
  }
}
};
```

token tokenfr fine\_auth\_token

```
var token = "${fine_auth_token}";

var t = window.location.protocol + "://" + window.location.hostname
var config = BI.extend({
  requestUrl: t,
  query: {
    token: token
  }
}, {
  websocketContextName: "/socket.io",
  websocketNameSpace: "/system",
  websocketPort: [38888, 39888, 38889],
  websocketProtocol: "plain"
});
//socketsocket.io
var socket = window.ssocket = BI.initSocket(config, function (e) {
  console.info("socket ");
  if (socket) {
    socket.emit("joinDP", "success")
    console.info(" ");
  }
})
```

socketsocket.ioon emit namespace/system.namespace

websocket

SocketIOServerFactory.getBroadcastOperations().sendEvent("", "");  
socketon

```
socket.on("", function(msg){
  alert(""+msg)
})
```

SocketIOServer addEventListener

```
socketIOServer.getNamespace("/system").addEventListener("joinDP", String.class, (client, data, ackReq) -> {
  FineLoggerFactory.getLogger().info(":{", client);
});
```

socketIOServer getNamespace /systemjoinDPString.class

SocketIOServerFactory.getBroadcastOperations().sendEvent("", data); SocketIOClient client

1.frclients

2.websockettokenclients

