

JavaScript Example

This example requires a valid token. To find out how to get a valid token, please read the [Obtain Authorization](#) page.

To run this example, you need to install socket.io and socket.io-client modules on your local machine.

- [socket.io](#)
- [socket.io-client](#)

For Socket.io documentation, click [here](#).

```
var io = require('socket.io-client');

var host = "https://CHANGEME";
var accessToken = "CHANGEME";

var deviceId = "CHANGEME";
var instanceId = "CHANGEME";

var serviceEndpoint = "/hubgetsb/ws/" + deviceId + "/" + instanceId;

//for self signed SSL certificate
process.env.NODE_TLS_REJECT_UNAUTHORIZED = "0";

var socket = io.connect(host, {
  query: "access_token=" + accessToken,
  transports: ['websocket'],
  path: serviceEndpoint
});
socket.on('connect', function(data){
  console.log('connected !\nwaiting for messages...');
});
socket.on('message', function (data) {
  var output= "[" + data.timestamp + "]" \n Context: "+data.content.context+"\n Event: "+data.content.
event+"\n FromApp: "+data.content.fromApp ;
  if(data.content.context=="sys.un"){
    output += '\n';
    output+=" Category: " + data.content.category + '\n';
    output+=" Event: " + data.content.event + '\n';
    output+=" Action: " + data.content.action + '\n';
    output+=" Important: " + data.content.important + '\n';
  }
  if(data.content.context=="sys.phonecall"){
    output += '\n';
    output+=" Status: " + data.content.payload.status + '\n';
    output+=" phoneCallId: " + data.content.payload.phoneCallId +'\n';
    output+=" phoneCallViewId: " + data.content.payload.phoneCallViewId+'\n';
    output+=" Extesion/DID: " + JSON.stringify(data.content.payload.extension)+'\n' ;
    if (data.content.payload.flow=="outgoing") {
      output+=" Dialed: " + data.content.payload.dialed+'\n';
    }
    if (data.content.payload.flow=="incoming") {
      output+=" CalerId: " + JSON.stringify(data.content.payload.callerid)+'\n';
    }
    output+=" Answered: " + data.content.payload.answered +'\n';
    output+=" Started: " + data.content.payload.started +'\n';
    output+=" Recorded: " + data.content.payload.recorded+'\n' ;
    if (data.content.payload.status!="destroyed"){
      output+=" Flow: " + data.content.payload.flow+'\n';
    }
    if (data.content.payload.disposition){
      output+=" Disposition: " + data.content.payload.disposition+'\n';
    }
    if (data.content.payload.hold=="on") {
      output+=" Hold: " + data.content.payload.hold +'\n';
    }
    if (data.content.payload.parked=="on") {
      output+=" Parked: " + data.content.payload.parked +'\n';
    }
  }
}
```

```

    }
    if(data.content.context=="sys.extension"){
        if(data.content.event=="register"){
            output += '\n';
            output+=" Extension: " + data.content.payload.extensionNumber + '\n';
            output+=" Expire: " + data.content.payload.registerExpire + '\n';
            output+=" FromIp: " + data.content.payload.deviceIP+'\n';
        }
        if(data.content.event=="create"){
            output += '\n';
            output+=" Alias: " + data.content.payload.alias + '\n';
            output+=" ExtensionId: " + data.content.payload.extensionId + '\n';
            output+=" Agent: " + data.content.payload.agentDevice+'\n';
            output+=" Preferred: " + data.content.payload.preferred+'\n';
            if (data.content.payload.recActive){
                output+=" RecState: " + data.content.payload.recActive+'\n';
            }
            if (data.content.payload.voicemailActive){
                output+=" Voicemail: " + data.content.payload.voicemailActive+'\n';
            }
        }
        if(data.content.event=="update"){
            output += '\n';
            output+=" Alias: " + data.content.payload.alias + '\n';
            output+=" ExtensionId: " + data.content.payload.extensionId + '\n';
            output+=" Agent: " + data.content.payload.agentDevice+'\n';
            output+=" Preferred: " + data.content.payload.preferred+'\n';
            if (data.content.payload.recActive){
                output+=" RecState: " + data.content.payload.recActive+'\n';
            }
            if (data.content.payload.voicemailActive){
                output+=" Voicemail: " + data.content.payload.voicemailActive+'\n';
            }
        }
        if(data.content.event=="delete"){
            output += '\n';
            output+=" Removed Extension Ids: " + JSON.stringify(data.content.payload.extensionIds) + '\n';
        }
    }

    console.log(output);
});
socket.on('disconnect', function(){
    console.log('disconnected');
});
socket.on('error', function(err) {
    console.log('error: ' + err);
});

```