CSTA

From Uni-wiki

Contents

CSTA API

This is the documentation for Uni-tel CSTA and can be reached through various addresses. For documentation, json has been chosen to explain requests/responses in details.

Addresses

Address	Accepts	Returns
csta.one-connect.dk:7001 (direct)	Accepts multiple xml-requests, separated by \x00	single xml-response
csta.one-connect.dk:7003 (direct)	Accepts single json-requests. socket.io (http://socket.io/) required.	single json-response
csta.one-connect.dk:7005 (http)	Accepts GET-parameters for single request	single json-response
csta.one-connect.dk:7006 (https)	Accepts GET-parameters for single request	single json-response

IP Validation

The API can be access'ed by adding ip-address in one-connect under settings for your pbx.

- Login to one-connect.dk
- Click "Settings" / "Indstillinger".
- Add IP-address under section "API". Please notice the text between "(" and ")".
- Click "Save" / "Gem".

Commands

StartApplicationSession

This command starts a session against CSTA and keep you logged in for a period of time. If timeout, you will be disconnected.

Request

To login, specify your credentials (userID, password and pbxDomain). You can specify the lifetime of your session by setting value (seconds) of requestedSessionDuration.

```
StartApplicationSession: {
   applicationInfo: {
      applicationSpecificInfo: {
         'vendorData:userID': 'test@mydomain',
         'vendorData:password': 'bla.bla',
         'vendorData:pbxDomain': '12345678.pbx.one-connect.dk'
      }
   },
   requestedSessionDuration: '60'
}
```

Response

To help you handling session at client, a session-id will be returned.

```
StartApplicationSessionPosResponse: {
   sessionID: '2e853292-bacf-418e-aa1e-86d9a88ea942',
   actualProtocolVersion: 'http://www.ecma-international.org/standards/ecma-323/csta/ed3',
   actualSessionDuration: '20'
```

```
}
```

ResetApplicationSessionTimer

This is to reset the session-timer, so your session does not timeout.

Request

```
'ResetApplicationSessionTimer': {
   requestedSessionDuration: 123
}
```

Response

```
'ResetApplicationSessionTimerPosResponse': {
   'actualSessionDuration': '123'
}
```

MakeCall

Request

```
'MakeCall': {
   callingDevice: 123,
   calledDirectoryNumber: 321,
   autoOriginate: 'doNotPrompt'
}
```

Response

```
'MakeCallResponse': {
   'callingDevice': {
      'callID': 'SIP/hpbx-000000bc',
      'deviceID': '123'
   }
}
```

SingleStepTransferCall

Request

```
"SingleStepTransferCall": {
   "activeCall": {
      "callID": "SIP/hpbx-0014907d",
      "deviceID": "123"
   },
   "transferredTo": "124"
}
```

Response

```
"SingleStepTransferCallResponse": {
   "transferedCall": {
      "callID": "SIP/hpbx-001490b9",
      "deviceID": "123"
   }
}
```

ConsultationCall

Request

```
"ConsultationCall": {
   "existingCall": {
      "callID": "SIP/hpbx-0014a738",
      "deviceID": "123"
   },
   "consultedDevice": "124"
}
```

Response

```
"ConsultationCallResponse": {
   "callingDevice": {
      "callID": "Local/91890743@transfer-00000bde;2",
      "deviceID": "123"
   }
}
```

SnapShotDevice

Request

```
'SnapShotDevice': {
   snapshotObject: '123@12345678.pbx.one-connect.dk'
}
```

Response

```
"SnapShotDeviceResponse": {
   "serviceCrossRefID": "12345678.pbx.one-connect.dk",
   "snapshotData": {
      "snapshotDeviceResponseInfo": {
         "connectionIdentifier": [
            {
               "callID": "SIP/hpbx-0000214e",
               "deviceID": "123",
               "privateData": {
                  "local_status": "Active",
                  "remote_user": "124"
               }
            }
         ]
      }
   }
}
```

ClearConnection

Request

```
"ClearConnection": {
   "connectionToBeCleared": {
      "callID": "SIP/hpbx-0013365b",
      "deviceID": "123"
   }
}
```

Response

```
"ClearConnectionResponse": {}
```

DirectedPickupCall

Request

```
"DirectedPickupCall": {
   "callToBePickedUp": {
      "callID": "SIP/hpbx-0014a1ec",
      "deviceID": 123
   },
   "requestingDevice": {
      "deviceID": "321"
   }
}
```

Response

```
"DirectedPickupCallResponse": {
   "pickedCall": {
      "callID": "SIP/hpbx-0014a1ec",
      "deviceID": 123
   }
}
```

MonitorStart

Request

```
"MonitorStart": {
      "monitorObject": {
            "deviceObject": 123
      }
}
```

Response

To help you handling monitoring, a reference-id will be returned.


```
"MonitorStartResponse": {
      "monitorCrossRefID": "123-1444294019863"
}
```


Events

"DeliveredEvent" happens when call is delivered to a phone.


```
"DeliveredEvent": {
   "monitorCrossRefID": "123-1444294019863",
   "connection": {
      "callID": "SIP/hpbx-0000038e",
      "deviceID": ["123"]
   },
   "alertingDevice": {
      "deviceIdentifier": ["123"]
   },
   "callingDevice": {
      "deviceIdentifier": "321"
   },
   "calledDevice": {
      "deviceIdentifier": ["123"]
   },
   "lastRedirectionDevice": {
      "deviceIdentifier": null
   },
   "localConnectionInfo": "alerting",
   "cause": "newCall",
   "networkCallingDevice": {
      "deviceIdentifier": ["123"]
   },
   "networkCalledDevice": {
      "deviceIdentifier": "321"
   },
   "associatedCallingDevice": {
      "deviceIdentifier": 14085551212
   }
}
```


"EstablishedEvent" happens when remote is picking up phone.


```
"EstablishedEvent": {
   "monitorCrossRefID": "2314-1444294019863",
   "establishedConnection": {
```

```
      "callID": "SIP/hpbx-0000038d",
      "deviceID": "321"
   },
   "answeringDevice": {
      "deviceIdentifier": "123"
   },
   "callingDevice": {
      "deviceIdentifier": "321"
   },
   "calledDevice": {
      "deviceIdentifier": "123"
   },
   "lastRedirectionDevice": {
      "deviceIdentifier": null
   },
   "localConnectionInfo": "connected",
   "cause": "normal",
   "networkCallingDevice": {
      "deviceIdentifier": null
   },
   "networkCalledDevice": {
      "deviceIdentifier": "123"
   },
   "associatedCallingDevice": {
      "deviceIdentifier": "notKnown"
   }
}
```


"ConnectionClearedEvent" happens when phone ended call.


```
"ConnectionClearedEvent": {
   "monitorCrossRefID": "123-1444294019863",
   "droppedConnection": {
      "callID": "SIP/hpbx-0000038e",
      "deviceID": "123"
   },
   "releasingDevice": {
      "deviceIdentifier": "123"
   },
   "localConnectionInfo": "null",
   "cause": "normalClearing"
}
```


"ApplicationSessionTerminated" happens when connection is closed from remote because of timeout.


```
"ApplicationSessionTerminated": {
   "sessionID": "2e853292-bacf-418e-aa1e-86d9a88ea942",
   "sessionTermReason": {
      "definedTermReason": "sessionTimerExpired"
   }
}
```

XML

XML has the same structure as json. Below is an example of MakeCall, based on previous example (json).

Request

```
<MakeCall>
   <callingDevice>123</callingDevice>
   <calledDirectoryNumber>321</calledDirectoryNumber>
   <autoOriginate>doNotPrompt</autoOriginate>
</MakeCall>
```

Response

```
<MakeCallResponse>
   <callingDevice>
      <callID>SIP/hpbx-000000bc</callID>
      <deviceID>123</deviceID>
   </callingDevice>
</MakeCallResponse>
```

GET (http/https)

To specify parameters as GET is much different that using json/xml since it is called directly via http(s). Following is required.

Headers

Following headers are required:

Header	Value
username	This is the same name as you use when you log in on one-connect.dk
password	This is the same password as you use when you log in on one-connect.dk
pbxdomain	This is the full domain of pbx. Example: 69102200.pbx.one-connect.dk

Parameters

The syntax of a parameter is "command[parameter]=value". So to make call, based on previous example (json), the query-string should be formed as

```
MakeCall[callingDevice]=123&MakeCall[calledDirectoryNumber]=321&MakeCall[autoOriginate]=doNotPrompt
```

[ ] indicates that it is a property. In this example, the property "callingDevice" is set on "MakeCall". To have nested properties, you need to specify it as "command[property][property]".

TEST

It is possible to do API-calls without executing any commands and return static content instead. This is to help/test API and for development.

To do a test, simple add a parameter called "test" and set the value to either "ok" or "error".

```
* "ok" will return a static response indicating that the request was executed normally.
* "error" will return a static response indicating that the request was not executed normally.
```

Example of a test to make call, based on previous example (json), request/response could look like this.

Request (error-response)

```
'MakeCall': {
    callingDevice: 123,
    calledDirectoryNumber: 321,
    autoOriginate: 'doNotPrompt',
    test: 'ok'
}
```

Response

```
"CSTAErrorCode": {
    "security": "securityInfoViolated",
    "details": "Explanation of error"
}
```

Scenarios

Transfer call unattended

To transfer an existing call, you need to use 2 x CSTA commands ("SnapShotDevice" and "SingleStepTransferCall").

The process is following.

- Use "SnapShotDevice" to request snapshot on existing call from CSTA .
- Call CSTA.
- Use "deviceID" and "callID" from snapshot response, to generate request for "SingleStepTransferCall".
- Call CSTA.

Example (https)

- /csta/?SnapShotDevice[snapshotObject]=123@12345678.pbx.one-connect.dk
- /csta/?SingleStepTransferCall[transferredTo]=124&SingleStepTransferCall[activeCall][deviceID]=123&SingleStepTransferCall[activeCall][callID]=SIP%2Fhpbx-00002142

Transfer call attended

To transfer an existing call, you need to use 2 x CSTA commands ("SnapShotDevice" and "ConsultationCall").

The process is following.

- Use "SnapShotDevice" to request snapshot on existing call from CSTA .
- Call CSTA.
- Use "deviceID" and "callID" from snapshot response, to generate request for "ConsultationCall".
- Call CSTA.
- Hang up call on "middle"-device to transfer call.

Example (https)

- /csta/?SnapShotDevice[snapshotObject]=123@12345678.pbx.one-connect.dk
- /csta/?ConsultationCall[consultedDevice]=123&ConsultationCall[existingCall][deviceID]=123&ConsultationCall[existingCall][callID]=SIP%2Fhpbx-00002146

Example

This example connects to CSTA, make a call and starting a monitor.

```
var http = require('http');
var socket = require('socket.io-client')('http://csta.one-connect.dk:7003');
var util = require('util');
socket.on('connect', function () {
    util.log("Connected");
    socket.send(
        {
            "StartApplicationSession": {
                applicationInfo: {
                    applicationSpecificInfo: {
                        "vendorData:userID": "test@mydomain.dk",
                        "vendorData:password": "bla.bla",
                        "vendorData:pbxDomain": "69102200.pbx.one-connect.dk"
                    }
                },
                requestedSessionDuration: 100
            }
        }
    );


});

socket.on("StartApplicationSessionPosResponse", function () {
    socket.send(
        {
            "MonitorStart": {
                monitorObject: {
                    deviceObject: 123
                }
            }
        }
    );

    socket.send(
        {
            "MakeCall": {
                callingDevice: 123,
                calledDirectoryNumber: 321,
                autoOriginate: "doNotPrompt"
            }
        }
    );
});

socket.once("EstablishedEvent", function (event) {
    var callID = event.EstablishedEvent.establishedConnection.callID;
    socket.send(
        {
            "SingleStepTransferCall": {
                activeCall: {
                    callID: callID
```

```
                },
                transferredTo: 12345678
            }
        }
    );
});

socket.on("message", function (message) {
    util.log(util.inspect(message));
});

socket.on("disconnect", function () {
    util.log("Lost connection");
});

socket.on("reconnecting", function ($number) {
    util.log("Trying to reconnect. Attempt number " + $number);
});

socket.on("reconnect_error", function () {
    util.log("Failed to reconnect");
});

socket.on("end", function () {
    process.exit();
});
```