# CALLBRIDGE PRO API DOCUMENTATION + EXAMPLES

Version 2.2

06 avril 2018

MCLEED - CALLBRIDGE

# How to use API with CallBridgePro

**CallBridgePro** is a telephony and sms app.

Call orders come either of:

- Your CRM application to which you will have added the use of APIs,
- Our Windows application (CallBridgePro.Win),
- Our Saas portal – Web application (CallBridgeProWeb).

Orders arrive almost immediately through **notifications** on the smartphone(s) having the CallBridgePro app installed .

The notification's *title* (before opening), through its prefix, indicates the action it will produce.

The terminology of prefixes is:

- " **Call.** ": Immediate dialing of the telephone number given in the notification
- " **Call** ": opening of a window allowing the user to check/modify the n° and to ask for its dialing. The Text (optional) can be used to inform the user about the reason of the call.
- " **SMS.** ": Launch the smartphone's native SMS app with the phone number(s) to be contacted + text of the message. On IOS phone user needs to press 'Send'.
- " **SMS** ": Opens a screen allowing the user to check/modify the phone number(s) + text of the SMS and to request its composition. On IOS phone user needs to press 'Send'.

There are also 2 "Services" API, you can use to:

- get the version n° of the WebServices: CallBridgeProApi,
- get the names/nick of smartphone users of the selected account.

API description is available at the following address:
http://164.132.49.2/CALLBRIDGEPROAPI_WEB/awws/CallBridgeProApi.awws?wsdl

For each action on the smartphone there is a service in CallBridgeProApi.

An example of each notification service is given in appendix (+ in javascript):

## CallBridgeProApi/Put_Notification_Sms :

Send SMS to 1 or more recipients.
Variables information :

**pLogin** : Account login
**pPass** : Account password
**pDateTime** : Notification's sending date and time, using format :YYYYMMDDhhmmss *Ex: 20170120093425*
**pImmediate** : 1 = Immediate SMS 0 = Display (for manual user action)
**pRecipients** : User name/nick (on smartphone) =recipient of the notification (userlist retrievable by API Service call, see section)
**pTitle** : Notification's title (visible only on android)
**pPhones** : Phone N° (if multi : separated by a comma)
**pText** : SMS text
**pPost** : Sender

## CallBridgeProApi/Put_Notification_Call :

Dial/start a call on the smartphone**.**

Variables information :

**pLogin** : Account login
**pPass** : Account password
**pDateTime** : Notification's sending date and time, using format :YYYYMMDDhhmmss *Ex: 20170120093425*
**pImmediate** : 1 = Immediate Call 0 = Display (for manual user action)
**pRecipients** : User name/nick (on smartphone) =recipient of the notification (userlist retrievable by API Service call, see section)
**pTitle** : Notification's title (notification's 1st line, visible only on android)
**pPhones** : Phone N° to call
**pText** : Optionnal text (information only, for example : *reason of the call*)
**pPost** : Sender

## CallBridgeProApi/Get_Version :

**Get CallBridgeProApi version number.**

Variables information :

**pLogin** : Account login

**pPass** : Account password

**pDateTime** : Notification's sending date and time, using format :YYYYMMDDhhmmss *Ex: 20170120093425*

## CallBridgeProApi/Get_Devices :

**Get the name (nick) of every smartphone connected to CallBridgePro.**
(this is needed to choose the recipient of each notification)

Variables information :

**pLogin** : Account login
**pPass** : Account password
**pDateTime** : Notification's sending date and time, using format :YYYYMMDDhhmmss *Ex: 20170120093425*

Detail come back through XML with :

IDClientAppareil : Internal ID of smartphone
Nom : Name (nick)
NotifFlag : Kind of smartphone (1=Android 2=Ios)
FlagStop : 1 = Unusable device (stopped/broken/stolen…)
DateCreation : Coonection's Date to CallBridgePro

## Full Javasript example:

For the pDateTime variable/tag, we use TheDateTime()

sending date and time using format : YYYYMMDDhhmmss *Ex: 20170120093425* pour le 20/01/2017 à 09h34m25s

**// Prepare datas to send Soap CallBridgeProApi/Put_Notification_Sms**

```
function Put_Notification_Sms() {
var requete;
requete=”<soap:Envelope xmlns:soap=\”http://schemas.xmlsoap.org/soap/envelope/\”
xmlns:xsi=\”http://www.w3.org/1999/XMLSchema-instance\” xmlns:xsd=\”http://www.w3.org/1999/XMLSchema\”><soap:Body>”;
requete += AjBalise(“pLogin”,”myLogin”);                    // Account Login
requete += AjBalise(“pPass”,”myPassword”);      // Account Password
requete += AjBalise(“pImmediate”,”1”);                      // 1 = SMS immediate    0 = display (for manual actions)
requete += AjBalise(“pDateTime”,TheDateTime() );          // Return as :YYYMMDDhhmmss  Ex: 20170120093425
requete += AjBalise(“pRecipients”,”Patrick S7”); // Name/Nick of the smartphone
requete += AjBalise(“pTitle”,”CallBridgePro”);    // Notification Title
requete += AjBalise(“pPhones”,”0689939667,0623631148”);         // Phone N° receiving SMS
requete += AjBalise(“pText”,”The text of the SMS”);         // Text SMS
requete += AjBalise(“pPost”,”Sender”);                      // device sending the notification
requete += “</soap:Body></soap:Envelope>”;
SendHttp(requete,”Put_Notification_Sms”,CallBackReponse)          // Send XML request and Service name to contact
}
```

**// Prepares data to send SOAP CallBridgeProApi/Put_Notification_Call**

```
function Put_Notification_Call() {
var requete;
requete="<soap:Envelope xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:xsi=\"http://www.w3.org/1999/XMLSchema-instance\" xmlns:xsd=\"http://www.w3.org/1999/XMLSchema\"><soap:Body>";
requete += AjBalise("pLogin","myLogin");                    // Account Login
requete += AjBalise("pPass","myPassword");      // Account Password
requete += AjBalise("pImmediate","1");                      // 1 = immediate call    0 = display only
requete += AjBalise("pDateTime",TheDateTime() );          // Return as :YYYMMDDhhmmss  Ex: 20170120093425
requete += AjBalise("pRecipients","Patrick S7"); // Name/Nick of the smartphone
requete += AjBalise("pTitle","CallBridgePro");    // Notification Title (1st lign on android)
requete += AjBalise("pPhones","0689939667"); // Phone N° to call
requete += AjBalise("pText","The text explanatory"); // Optionnal Text optionnel for the caller
requete += AjBalise("pPost","Sender");                      // Device sending the notification
requete += "</soap:Body></soap:Envelope>";
SendHttp(requete,"Put_Notification_Call","CallBackReponse") // Send XML request and Service name to contact
}
```

**// List of all my devices (smartphones)**

```
function Get_Devices() {
var requete;
requete="<soap:Envelope xmlns:soap=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:xsi=\"http://www.w3.org/1999/XMLSchema-instance\" xmlns:xsd=\"http://www.w3.org/1999/XMLSchema\"><soap:Body>";
requete += AjBalise("pLogin","myLogin");                    // Account Login
requete += AjBalise("pPass","myPassword");                  // Account Password
requete += AjBalise("pDateTime",TheDateTime() );            // Return as :YYYMMDDhhmmss  Ex: 20170120093425
requete += AjBalise("pAll","0");                            // all devices     0=only active       1=all
requete += "</soap:Body></soap:Envelope>";
SendHttp(requete,"Get_Devices","CallBackReponse") // Send XML request and Service name to contact
}
```

```
// Send Request to CallBridgeProApi
function SendHttp(pRequete,pService,pCallBack) {
var params = 'xml='+pRequete+'&action='+"urn:CallBridgeProApi/"+pService;
var xmlhttp;
xmlhttp = new XMLHttpRequest();
xmlhttp.open("POST","https://www.mcleed.net/CallBridgeProApi_WEB/awws/CallBridgeProApi.awws",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded; charset=UTF-8");
xmlhttp.setRequestHeader("Content-length",params.length);
xmlhttp.setRequestHeader("Connection","close");
xmlhttp.onreadystatechange = function() {
        if (xmlhttp.readyState==4 && (xmlhttp.status==200)) {
        var lRetour = xmlhttp.responseText;
        var tChaine = lRetour.split("Result>");
        var tReponse = tChaine[1].split("</");
        if (pCallBack) { pCallBack(tReponse[0]) };
        }       }
xmlhttp.send(params);
}

 // Process the response ( pCallBack )
function CallBackReponse(pText){
 var sText=pText.replace("<br>",String.fromCharCode(10))
 alert("callbridge",sText);
}
// Return date local time as  YYYYMMDDhhmmss
function TheDateTime(){
var ladate=new Date();
var Y=ladate.getFullYear();
var M=ladate.getMonth()+1;
if (M<10) {M = "0" + M};
var D=ladate.getDate();
if (D<10) {D = "0" + D};
 var h=ladate.getHours();
if (h<10) {h = "0" + h};
var m=ladate.getMinutes();
if (m<10) {m = "0" + m};
var s=ladate.getSeconds();
if (s<10) {s = "0" + s};
return Y+""+M+""+D+""+h+""+m+""+s;
}
  // return beacon and value as : <Balise  xsd:type="xsd:string">Value</Balise>
function AjBalise(pBalise,pValeur) {
var sRetour = "<"+pBalise+"  xsd:type=\"xsd:string\" >"+StringtoUTF8(pValeur)+"</"+pBalise+">"
return sRetour
}
```

```
// for special charaters to UTF8
function StringToUTF8(pTexte){
   sEot=String.fromCharCode(4);
   var sText0="", sText1='', sText2=pTexte, sText3='', i=10, t='';
   while (sText0!=sText2 && i>0){
      i--;sText0=sText2;sText1=sText2;
      sText2=sText1.replace("È","è");sText1=sText2;
      sText2=sText1.replace("É","é");sText1=sText2;
      sText2=sText1.replace("Ê","ê");sText1=sText2;
      sText2=sText1.replace("&amp;","¤");sText1=sText2;
      sText2=sText1.replace("€","\$");sText1=sText2;
      sText2=sText1.replace(" "," ");sText1=sText2;
      sText2=sText1.replace(sEot,"?");sText1=sText2;
      sText2=sText1.replace("  "," ");sText1=sText2;
   }
   var sortie=unescape(encodeURIComponent(sText2));
   return sortie;
}
```

*END OF THE DOUCMENTATION*