



WebRTC Manual



WebRTC Manual

Introduction of WebRTC

Web Extensions is easier understanding of WebRTC. WebRTC(Web Real-Time Communication) is an API definition drafted by the World Wide Web Consortium(W3C) that supports browser to browser applications for voice calling without the need of either internal or external plugins. Adopting this innovative WebRTC technology, enterprise can serve their customers with direct voice communication via its website.

Here we will introduce how to install WebRTC and how to use it to make the call. There are two methods for reference:

Method 1:

Embed WebRTC UI on CooVox System

Method 2:

Deploy WebRTC UI on aWebserver

Method 1 Embed WebRTC UI on CooVox System

Objective:

Embedding WebRTC UI on your CooVox system is convenient for users to complete the Web phone functionality by utilizing existing Zycoo CooVox V2 IP PBX. You can make the call without installing any plugins or softphone.

There are two different situations for Method 1.

- 1) WebRTC UI client and IP PBX are in the same LAN
- 2) CooVox IP PBX is behind Router NAT

Here take Google Chrome as example to clarify above setting ways.

1) WebRTC UI client and IP PBX are both in the same LAN

WebRTC UI client and IP PBX are both in the same LAN, so don't need NAT traversal.

Please configure as below steps:

Step 1:

Download [ulmage-md5.u20v2.webrtc_patch](#) (firmware of U20V2),
[ulmage-md5.u50v2.webrtc_patch](#) (firmware of U50V2) and upgrade webrtc-ui on Coovox V2.

Note: Coovox V2 system will automatically reset after upgrading.

Resetting time will be around 1 min.

Step 2:

Once upgrade has successfully completed, open the following link:

<https://192.168.1.65:9999/webrtc/> in Google Chrome browser. See below Figure1:

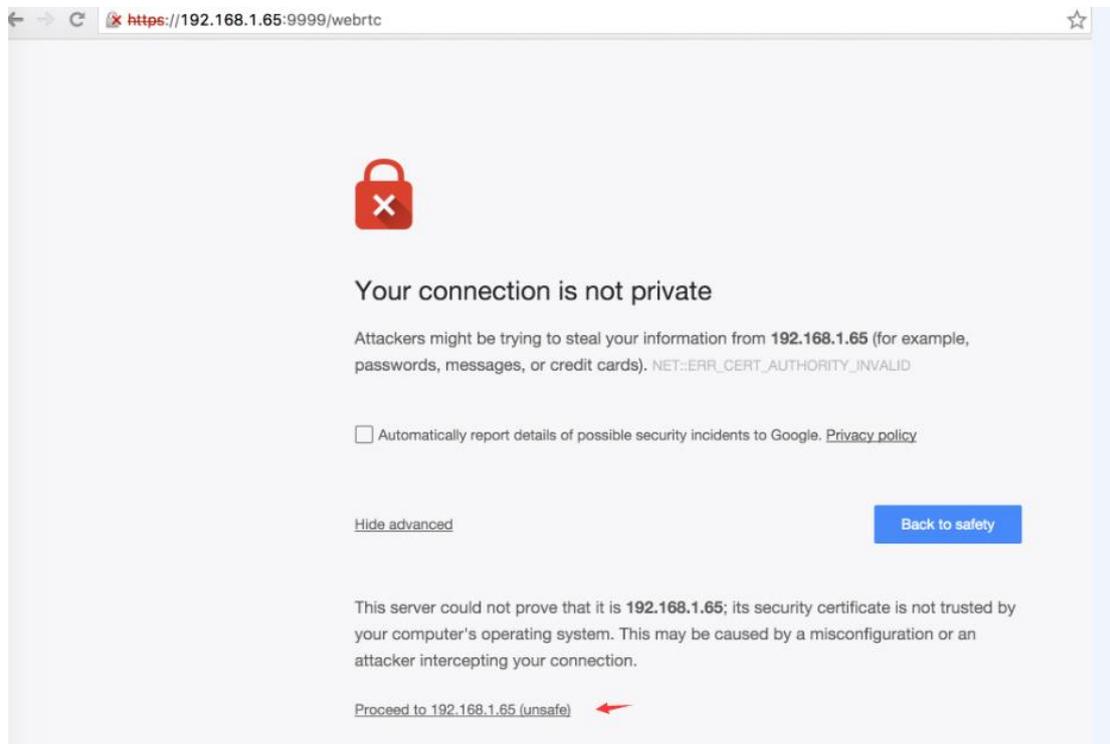


Figure 1. Safety Warning

Chrome will warn you the page is unsafe. To bypass this warning, you must click “Advanced” and then click “Proceed to 192.168.1.65(unsafe)” to continue. You will then be redirected to the web extension register page as below Figure 2.

The screenshot shows a login form for a web extension. The form is titled "Webphone" and contains four input fields, each with a red question mark icon to its right. The fields are: "Name" (with example text "i.e. Homer Simpson"), "SIP URI" (with example text "i.e. sip:homer@your-domain.com"), "SIP password", and "WS URI" (with example text "i.e. wss://your-domain.com:8089/ws"). Below the "WS URI" field is a link for "advanced settings".

Figure 2. Login Extension

Note:

1. You are able to login and use the Webphone provided that you have created Web Extension on your Coovox IP PBX. Your Webphone user name and password must match that you set on your Web Extension (Web Extensions are not the SIP/IAX2 extensions on Coovox).
2. There are two possible options to choose from when setting up your connection. First option is WS (WebSocket) Protocol which is an independent TCP-based protocol providing full-duplex communication channels over a single TCP connection. The WebSocket protocol was standardized by the IETF as RFC 6455 in 2011, and the WebSocket API in Web IDL is being standardized by the W3C.
3. Option 2 is WSS (WebSockets over SSL/TLS), like HTTPS, WSS is encrypted and we strongly recommend the secure wss:// protocol over the insecure ws:// transport. A variety of attacks against WebSockets are almost impossible if the transport is secured.

Create Web Extension

See below Figure 3: Select “WSS” for Transport when adding Web Extension, then save and activate the settings.

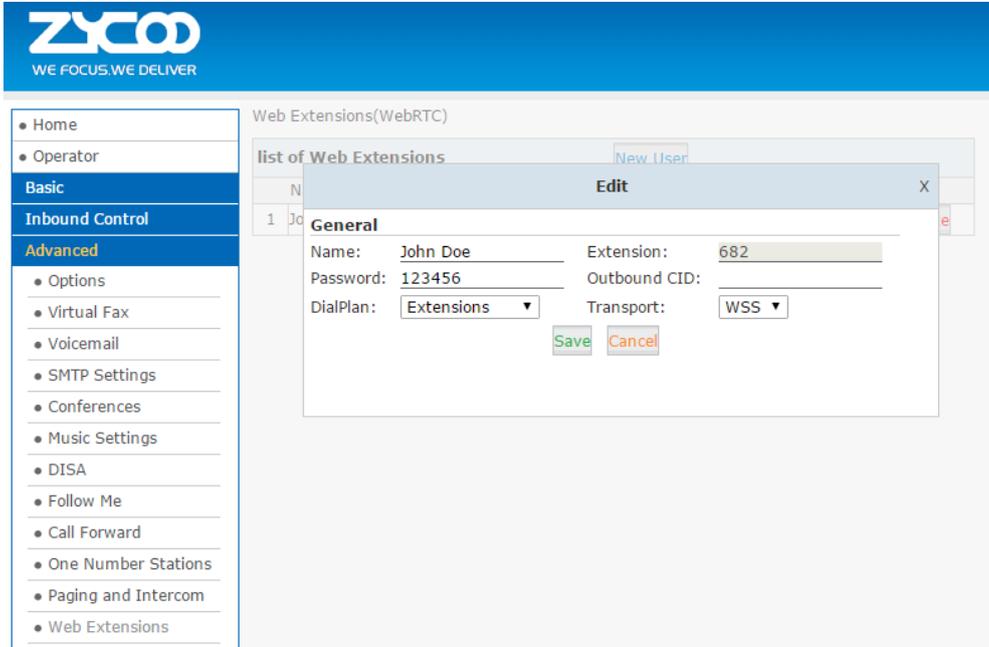


Figure 3. Create Web Extension

Step 3:

Input Web Extension(Name), registration address(SIP URI), registration port of WebRTC, use WSS(wss://192.168.1.65:8089/ws) by default in WS URI (192.168.1.65 is the IP of IP PBX). See below Figure 4.



Figure 4. Registration

Step 4:

Once registration has completed a user can make and receive calls. See below Figure 5.



Figure 5. Make Calls from Web Extension

Notice: If you want to use transport protocol WS(not recommended), configure as Figure 6.

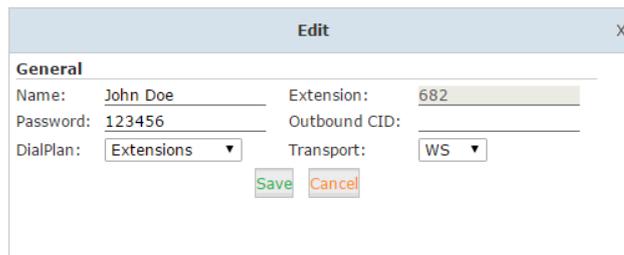


Figure 6. WS Transport Protocol

When you register a web extension using WS transport protocol then you must set the “WS URI” to “ws://192.168.1.65:8088/ws” as below Figure 7.



Webphone

Name	i.e. Homer Simpson
John Doe	?
SIP URI	i.e. sip:homer@your-domain.com
682@192.168.1.65	?
SIP password	
.....	?
WS URI	i.e. ws://your-domain.com:8088/ws
ws://192.168.1.65:8088/ws	?

[advanced settings](#)

Figure 7. WS Registration

WS does not require encryption, however, Google Chrome has its security mechanism configured so that you have to manually set Google Chrome to load unsafe scripts to proceed. This is achieved by first clicking the shield icon on the right side of the address bar, and then click “Load unsafe scripts” as below figure 8.

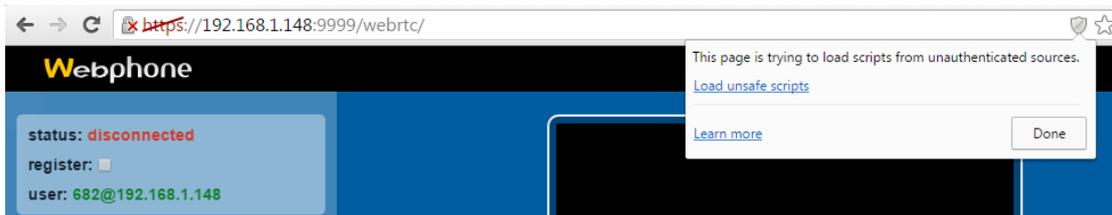


Figure 8. Load Unsafe Scripts

The page will now reload and you have to complete the registration process again after which you’ll be able to register.

2) Coovox PBX is behind Router NAT

Under this deployment scenario, WebRTC requires the support of ICE server. In the WebRTC UI firmware, ICE client has been configured through the coordination of Google(stun server) and Zycoo(turn server). If you have your own ICE server, then configure through “Advanced Settings” on the login page. See below Figure 9:

Read the documentation of the advanced settings [here](#). ✖

authorization_user	<input type="text"/>
register	<input checked="" type="checkbox"/>
register_expires	<input type="text" value="600"/>
registrar_server	<input type="text"/>
no_answer_timeout	<input type="text" value="60"/>
session_timers	<input type="checkbox"/>
peerconnection_config (use JSON)	<input type="text" value='{"iceServers": [{"urls": ["stun:stun4.l.google.com'/>
use_preloaded_route	<input type="checkbox"/>
connection_recovery_min_interval	<input type="text" value="2"/>
connection_recovery_max_interval	<input type="text" value="30"/>
hack_via_tcp	<input type="checkbox"/>
hack_via_ws	<input type="checkbox"/>
hack_ip_in_contact	<input checked="" type="checkbox"/>

ws://zycoopbx_ip:8088/ws

advanced settings

Figure 9. Modify ICE server settings

Please use below settings as an example(only one line of code is used for Zycoco test), you need to change this based on your iceServers parameters which retaining this format. Close after completing this field.

```
{ "iceServers": [ { "urls":  
["stun:stun1.l.google.com:19302"]}, {"url": "turn:117.176.159.163", "username": "toto", "credential":  
:"password"} ], "gatheringTimeout": 2000 }
```

As you are behind a router, you are required to configure the port on NAT router mapping to WebRTC registration port 8089 and Web visiting port 9999(default) of CooVox IP PBX.

Then complete the registration with public IP address, see below Figure 10:



Figure 10. Register with PublicIP Address

Method 2 Deploy WebRTC UI on WebServer

Requirement:

Deploying in this scenario is only recommended for highly experienced technicians.

Objective:

Deploying WebRTC UI on your own Website is a convenient way for customers to ask for direct support /service from the website.

Step1:

Download original reference code ClickCall.tar.gz, note that you are required to modify some code. Please see below assets directory structure in Figure 11.

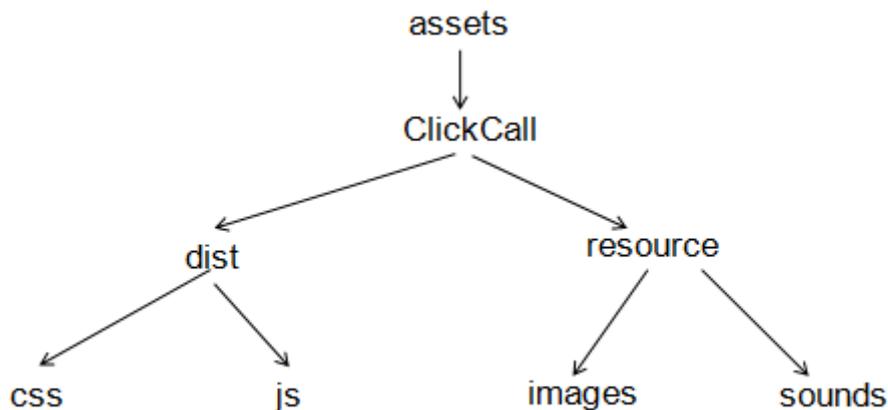


Figure 11.Reference code directory structure

Modify the file under css directory as required to create Web format; you can use UltraEdit to modify the file "style.css" under css directory and "clickCall.js"," gui.js","init.js" three files under js directory then complete registration information and links.

1. Modify file clickCall.js: set www.zycoo.com as your Web IP or domain.

```
urls.rtcninja =  
'http://www.zycoo.com/assets/ClickCall/dist/js/rtcninja.js';  
urls.jquery =  
'http://www.zycoo.com/assets/ClickCall/dist/js/jquery.js';  
urls.i18n =  
"http://www.zycoo.com/assets/ClickCall/dist/js/jquery.i18n.js";  
urls.messagestore =  
"http://www.zycoo.com/assets/ClickCall/dist/js/jquery.i18n.messagesto
```

```
re.js";
urls.jssip = 'http://www.zycoo.com/assets/ClickCall/dist/js/jssip.js';
urls.init = 'http://www.zycoo.com/assets/ClickCall/dist/js/init.js';
urls.gui = 'http://www.zycoo.com/assets/ClickCall/dist/js/gui.js';
urls.css = 'http://www.zycoo.com/assets/ClickCall/dist/css/style.css';
```

2. Modify registration information in the file init.js (at the end of the file).

```
var user = local_config.user || {
    username: '682',
    password: '123456',
    host: '117.176.159.157',
    wsPort: '8088',
    wsProtocol: 'ws'
};
createUA(user);
phone_call_button.click(function (event) {
    console.log(event);
    varivr_num = local_config.ivr_num || '910';
    GUI.phoneCallButtonPressed(ivr_num);
});
```

Note: Modify username as Web Extension; password as Web Extension's password and also modify host as IP address of Coovox IP PBX. Keep other settings unchanged.

If Coovox IP PBX is behind router NAT, the port 8088 should be mapped to the internal IP PBX'IP.

(Above settings is just for reference)

3. Modify the link information in gui.js file: Search www.zycoo.com in the file and change it to your Web server IP or domain.

Modify the click icon as you required: search icon-register.png in the file and set it as your required icon under image directory. E.g.: icon-register-en.png, icon-offline-en.png

4. Under distcss directory, modify the www.zycoo.com in the style.css file as your Web IP or domain.(Same way as previous)

Note: After modification, place the assets directory under the DocumentRoot of your Web server

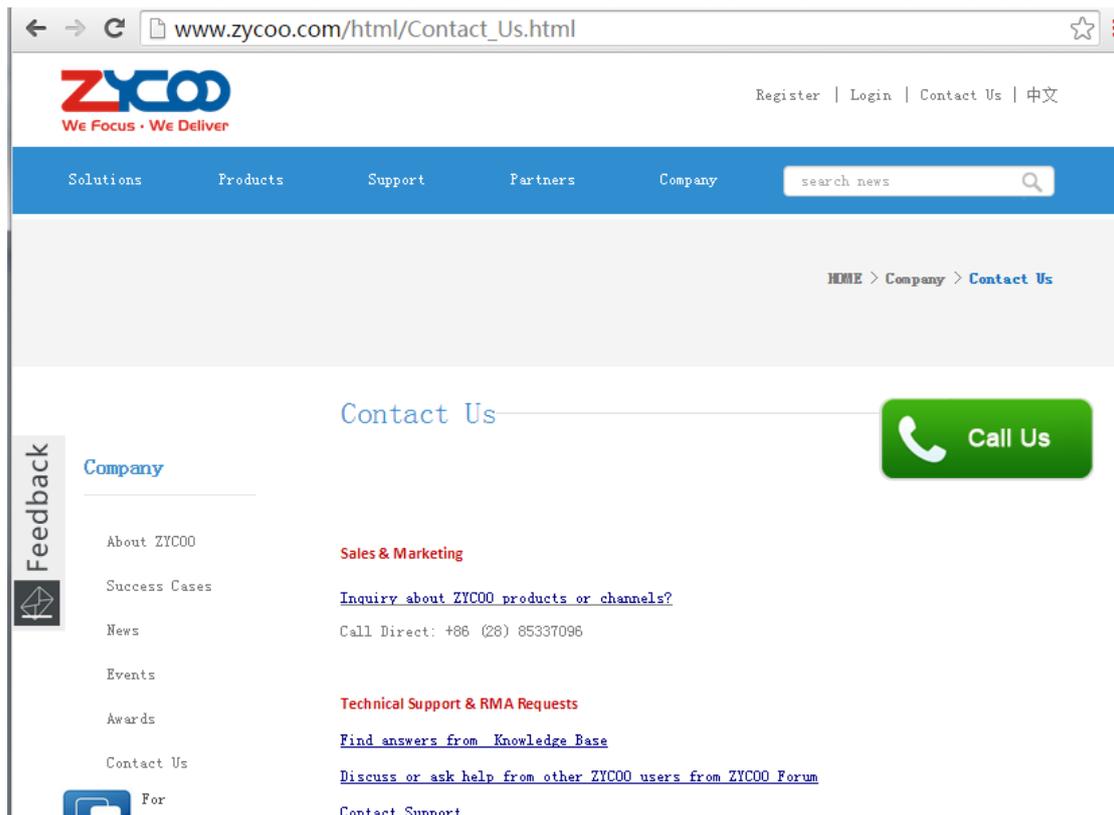
Step2:

Embed modified codes to the appropriate place on your Web page <body></body> . Please see below as reference.

```
<script>
document.write('<scriptsrc="http://www.zycoo.com/assets/ClickCall/dist/js/clickCall.js">/script>')
</script>
```

Step 3:

Visit your Website. E.g.: Zycoo Website and you can see the icon Call Us.



Appendix: Browsers Support WebRTC

Zycoco Test Result:

Browser	Version	Test Condition(Register, dial, pickup, hangup, hold, second dial)---http	Test Result
Google Browser (Andriod Mobile)	38.0	OK	Success
	40.0	OK	Success
	45.0	OK	Success
	46.0	OK	Success
	47.0	Register succeeded, but inbound/outbound call failed	Fail
Google Browser (PC)	38.0	OK	Success
	39.0	OK	Success
	40.0	OK	Success
	41.0	OK	Success
	42.0	Outbound call-single pass and no voice on Web Extension side; No auto-hangup, inbound call is normal for both sides.	Fail
	43.0	OK	Success
	44.0	OK	Success
	45.0	OK	Success
	46.0	OK	Success
Opera Browser (PC)	27	OK	Success
	28	OK	Success
	29	No voice for outbound call whatever internal or external calls. Single pass; inbound call is normal.	Fail
	30	OK	Success
	31	OK	Success
	32	OK	Success
	33	OK	Success
Firefox Browser	All version	Test result is not good enough. Many bugs existing in Firefox while supporting WebRTC. Firefox is not recommended.	Fail
Opera Browser (Andriod Mobile)	All version	Doesn't Support WebRTC	Fail
IE Browser	All version	Doesn't Support WebRTC	Fail