

MizuTech WebPhone

Quick Start

What is it?

The [WebPhone](#) (or “websipphone”) is very similar with a regular VoIP softphone software, with the following important differences:

- runs in web browsers (but also as a standalone desktop application)
- it can be easily embedded and integrated with any website or application
- it can be easily customized:
 - you can change it after your needs by modifying its parameters including the display language
 - you can create any kind of user interface with simple HTML/CSS or no GUI at all
 - you can control the call logic as you wish with the API (JavaScript API / HTTP API / SDK)
 - use a skin or run it in the background

It is implemented as a platform independent [java applet](#), a full featured sip stack in a single small .jar file compatible with all popular OS (Windows, Linux, MAC) and browsers (Firefox, IE and others). Unlike Flash or HTML5 based solutions, it doesn't need any intermediary server. It acts like a traditional softphone, using the open standard SIP and RTP protocols compatible with all SIP software (server/softswitch/softphone) or device (PBX/IP phone/gateway), connecting directly from the user browser to your SIP server.

All the common VoIP features are supported (udp/tcp/http, reinvoke, hold, forward, transfer, conference, video, call-recording, G729, opus, wideband/HD voice, dtmf, etc).

The webphone covers a wide range of use-case starting from a simple click to call button to a fully featured softphone on the web. It can be easily integrated with any kind of software (such as call center or voice support) or embedded in devices such as a hardware PBX. The webphone is stable, mature software and new versions are released frequently with new features and latest standards.

The webphone can be used as:

- static html: for example as a fully featured pre-customized softphone or click to call button on your website
- dynamic html: applet parameters generated dynamically from your server side script, for example a click to call button with the called number changing depending on the content
- via the API: JavaScript API/ HTTP API/SDK. So instead of server side application logic, you can implement all logic on client side: web or desktop
- a mix of the above
- as a standalone desktop application (with or without API and application parameters)

How do I use it?

Requirements:

You will need one (or more) working SIP account to be able to use the webphone. Free accounts are available at almost all [VoIP service provider](#) (including [Mizutech VoIP service](#)) or you can use your own VoIP server ([Asterisk](#), [Mizutech](#) or [any other](#) SIP compatible).

If you don't have a SIP account, you can still use the webphone to make direct peer to peer calls (webphone to webphone or webphone to any third party SIP softphone) but in this case you will have to type the full URI for the called person (user@ip:port), so the usage via a SIP server is easier.

Deployment:

Copy the webphone.jar to your webserver and refer to it from anywhere from your html with the “applet” tag (see example below or use the Example.html from the demo package).

Set at least the “serveraddress” applet parameter (within the applet tag) to the IP or domain name of your VoIP server.

That's all. Your users now can initiate or accept calls.

Optional:

- Configure: change any of the applet parameters customizing the software after your needs (these are listed in the documentation. The most important are the serveraddress, username, password and callto parameters).
- Customize: Create your own skin, click to call button or full softphone user interface. You can easily build it from scratch or use the skin templates from the Example directory.
- Integrate: if you have some basic development knowledge, then you can also use the API (HTTP, native or JavaScript API) to create your application logic. Have a look in the documentation API section for the details.
- You can also use the webphone from your desktop/terminal/command prompt. Example:
java -jar webphone.jar upperserver=YOURVOIPSERVERADDRESS username=XXX password=YYY callto=ZZZ loglevel=5

Quick test:

1. Deploy the webphone as described above.
2. Make sure that [java](#) is installed on your client PC (the webphone will auto-forward to install, but for test it is better to verify and [install java](#) if necessary)
3. Open the URL of the page where you have deployed the applet in step 1.

Optionally you can also test by running the webphone directly from your desktop. For this just open the Example.html (or any other example or skin from the Example directory. You can also run the webphone from the command line.)

Some more details:

To deploy the webphone on your webserver, all you need is to copy the “webphone.jar” near your html files on your webserver and refer to it from your html page using the “applet” tag (place it anywhere you wish the user interface to appear inside the html body).

! Make sure that your webserver doesn't block .jar files. You can test it easily by typing the exact URL to the webphone.jar file into your browser address bar. If works ,then you should be able to download the jar file. Otherwise you will have to add the .jar mime type to the webserver allowed list.

Example (copy-paste this into your html):

```
<applet
archive = "webphone.jar"
codebase = "."
code = "webphone.webphone.class"
name = "webphone"
width = "300"
height = "330"
hspace = "0"
vspace = "0"
align = "middle"
mayscript = "true"
scriptable = "true"
alt="Enable or install java: http://www.java.com/en/download/index.jsp"
>
<param name = "serveraddress" value = "VOIP_SERVER_ADDRESS">
<param name = "username" value = "">
<param name = "password" value = "">
<param name = "callto" value = "">
<param name = "loglevel" value = "1">
<param name = "other" value = "insert other optional parameters like these lines or you can pass it also via API">
<param name = "MAYSCRIPT" value = "true">
<param name = "scriptable" value = "true">
<param name = "pluginspage" value = "http://java.com/download/">
<param name = "permissions" value = "all-permissions">

<b>You must enable java or install if not already installed from <a href="http://www.java.com/en/download/index.jsp"> here </a> </b>
</applet>
```

You will have to set at least the “serveraddress” applet parameter. If you don't have a VoIP account (or your own server) then you can register to one of the thousands of available VoIP service providers from all around the world (try [ours](#) or [search google](#)).

In certain situations you might have to preset the webphone with a sip account and a number to call (for example if you wish to implement a simple click-to-call button). In this case you have to set the following applet parameters accordingly: username, password, autocal, callto.

Otherwise you can let the users to type their login username/password. It can be used also without a VoIP server address by making calls directly to other SIP endpoints (in this case you will have to use full SIP URI of the called party).

In case if you wish to create a custom skin (as simple as a button or as complex as a full featured softphone) you will need to use the webphone JavaScript API. This means that you will have to embed the webphone as we have described above, but you will hide it's default user interface (set the applet width and height to 1), display your own html user interface and control the webphone instance (running in the background) with JavaScript function calls. For this you can use one of the skins what we include with the webphone or create your own skin from scratch. Also it is possible to modify on of the existing skins after your needs, or just use the wp_common.js to simplify the interaction with the JavaScript API.

Please note that it is perfectly fine to run the webphone without any html skin. You can also run it in the background driven by your logic, embed it in hardware devices or use its API directly from your applications (Java lib/SDK).

For example making a call from javascript is as simple as this:

- insert the webphone applet tag to wherever you wish into your html (like the example above)
- obtain the webphone handle (see the initcheck function in the skin examples or just use document.applets[0] as the webphonehandle)
- optionally add a "webphonetojs" javascript function where you can handle notifications from the webphone such as "ringing"
- call webphonehandle.API_Register(server,username,password) (the webphone can also auto-register, so this step is also optional)
- call webphonehandle.API_Call(-1,numbertocall) to initiate the outgoing call
- call webphonehandle.API_Hangup(-1) to disconnect the call.

Explore

You should check the [webphone homepage](#) if you haven't done it yet for a longer presentation.

Have a look at the [demo pages](#) or [screenshots](#) to see how it looks like.

Download the [demo package](#) and start to use it on your website or from your application. The demo package contains everything you need to get started with the webphone (the webphone.jar itself, some additional modules, the documentation and the examples/skins).

In the [documentation](#) you can find:

- more details about the webphone (features, usage)
- list of the applet parameters (in case if you wish to change/fine-tune the default settings)
- JavaScript API details (if you wish to create your own skin or for custom call logic)

The demo version has some limitations such as a 100 second call limit after some calls and a trial limit, but it should be perfectly fine during development.

You will need a license to be able to use the webphone in production. The pricing can be found [here](#). On your payment we will send your webphone build within one work-day.

For any enquiries (including help, sales or technical questions) write to us: websipphone@mizu-voip.com.