

How to use Mizu webphone with PHP

About

This is a basic demonstration about one way the webphone could be used with PHP. Not a great demonstration (we are not very familiar with PHP), but it might ease the start for novice developers who needs some PHP-webphone integration with their website.

The webphone is a modern cross-platform browser softphone, offering integration capabilities with any third-party software. Integrating the mizu webphone into your website/web application you will benefit of VoIP capabilities like voice calls, video calls, conference, call-divert features (hold, transfer, forward), voicemail, chat, SMS, presence, call recording, file transfer, DTMF, offline messages, multi-line support.

In our project we have used the following tools:

Web server: IIS 6.1

Php: PHP Version 7.0.9

Database: mysql 5.7

Database design tool: MySQL Workbench

The screenshot shows the MySQL Workbench interface. The left sidebar contains the 'Navigator' pane with sections for MANAGEMENT, INSTANCE, PERFORMANCE, and SCHEMAS. The 'SCHEMAS' section shows a tree view of databases, including 'mizu'. The 'mizu' database is selected, and the 'tb_cdrs' table is highlighted. The main query editor shows the query: `SELECT * FROM mizu.tb_cdrs;`. The 'Result Grid' displays the query results with columns: id, caller, called, duration, disconnection, and datum. The 'Output' pane at the bottom shows the execution log with 10 rows of results, including the time, action, message, and duration for each query execution.

id	caller	called	duration	disconnection	datum
445	1122	2233	4	Call terminatio...	2017/05/11 08:04:45om
446	1122	2233	8	Bve received (refer received)	2017/05/11 08:05:53om
447	1122	2233	12	Bve received (refer received)	2017/05/11 08:06:37om
448	1122	2233	13	Bve received (refer received)	2017/05/11 08:07:55om
449	1122	2233	0	Busv Here 486 received (No resoonse f...	2017/05/11 08:09:45om
450	1122	2233	13	Bve received (refer received)	2017/05/11 08:11:42om
451	1122	2233	25	Call terminatio...	2017/05/11 08:13:19om
452	1122	2233	7	Bve received (refer received)	2017/05/11 08:14:46om
453	1122	2233	5	Call terminatio...	2017/05/11 08:16:07om
454	1122	2233	8	Call terminatio...	2017/05/11 08:16:32om
455	1122	5129	2	Call terminatio...	2017/05/11 08:16:36om
456	1122	2233	19	Call terminatio...	2017/05/11 08:18:11om

Example

In the below example is described the following scenario:

The user will login to the webpage with his username/password and the webphone will automatically register with his sip username and sip password.

After each call a CDR will be automatically created (tb_cdrs), it will contain informations like caller party, called party, call duration, call datum, disconnect reason.

The users table (tb_users) has a status field, which will contain the users current status (callSetup, callRinging, callConnected, callDisconnected)

In the first step we need to create the tb_users and the tb_cdrs tables.

For this you just need to run the following queries:

```
CREATE TABLE `tb_users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(255) DEFAULT NULL,
  `passwd` varchar(255) DEFAULT NULL,
  `sipusername` varchar(255) DEFAULT NULL,
  `sippassword` varchar(255) DEFAULT NULL,
  `credit` varchar(255) DEFAULT NULL,
  `state` varchar(255) DEFAULT NULL,
  `fullname` varchar(255) DEFAULT NULL,
  `address` varchar(255) DEFAULT NULL,
```

```
`city` varchar(255) DEFAULT NULL,
PRIMARY KEY (`id`),
UNIQUE KEY `id_UNIQUE` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8
```

```
CREATE TABLE `tb_cdrs` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`caller` varchar(255) DEFAULT NULL,
`called` varchar(255) DEFAULT NULL,
`duration` varchar(255) DEFAULT NULL,
`discreason` varchar(255) DEFAULT NULL,
`datum` varchar(255) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=307 DEFAULT CHARSET=utf8
```

When you are ready with the above tables, you will have to connect to your mysql database.

Below you will find the php source code for it:

```
$servername = "YOURSERVERIP";
$username = "DBUSERNAME";
$password = "DBPASSWORD";
$dbname = "DBNAME";
$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
```

After it, you will have to verify that the entered username/password (into the login form) is correct.

In our example they are transferred using the \$_POST method.

If yes, the webphone will start using the sip username and sip password associated to the logged in user.

In the followings please find the php source code:

```
$myusername = $_POST['myusername'];
$mypassword = $_POST['mypassword'];
$sql = "SELECT username,passw,sipusername,sippasword FROM tb_users WHERE username= '$myusername' and passw= '$mypassword'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    // output data of each row
    echo "SUCESSFULLY LOGED IN";
    echo "<br><br>";
    while ($row = mysqli_fetch_assoc($result)) {
        $sipuser = $row["sipusername"];
        $sippasw = $row["sippasword"];
        header ("Location:
http://127.0.0.1:88/YOURPATH/techdemo_example.html?wp_serveraddress=88.150.183.67&wp_username=$sipuser&wp_password=$sippasw");
    }
} else {
    echo "LOGIN FAILED";
}
mysqli_close($conn);
```

At this moment the webphone is registered, so you are able to initialize and receive calls.

The next goal is to create a CDR (call-detail-record) after each call. The key function to achieve this is the **onCdr**.

Below you can find the Javascript source code, where the parameters (caller, called, duration, reason) are sent to the cdrs.php page using AJAX method, where they will be inserted into the database.

```
webphone_api.onCdr(function (caller, called, connecttime, duration, direction, peerdisplayname, reason)
{
    var ajaxRequest; // The variable that makes Ajax possible!
    try
    {
        // Opera 8.0+, Firefox, Safari
```

```

ajaxRequest = new XMLHttpRequest();
}catch (e)
{
// Internet Explorer Browsers
try
{
ajaxRequest = new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
try
{
ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
} catch (e){
// Something went wrong
alert("Your browser broke!");
return false;

ajaxRequest.open("GET", "cdrs.php?q="+caller+"&w="+called+"&z="+duration+"&x="+reason,true);
ajaxRequest.send();
});

```

Once the cdrs.php page will receive the above parameters, they will be inserted into the table tb_cdrs.

Except this parameters, which are received by AJAX, the current date is also inserted.

Here is the PHP source code of it:

```

$datum = date("Y/m/d h:i:sa");
$caller = $_GET['q'];
$called = $_GET['w'];
$duration = round($_GET['z']/1000);
$reason = $_GET['x'];

$sql="INSERT INTO tb_cdrs (caller, called, duration, discreason, datum) VALUES ('$caller','$called','$duration','$reason','$datum')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
mysqli_close($conn);

```

To get the users status, you need to use the onCallStateChange function and update the tb_users table state field for the caller and called party.

Please find below the Javascript source code for it:

```

var currentuser = "";
var username = "";
webphone_api.onCallStateChange(function (status, direction, peername, peerdisplayname)
{
    var ajaxRequest; // The variable that makes Ajax possible!

    try
    {
        // Opera 8.0+, Firefox, Safari
        ajaxRequest = new XMLHttpRequest();
    }catch (e)
    {
        // Internet Explorer Browsers
        try
        {
            ajaxRequest = new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch (e)

```

```

{
try
{
ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
} catch (e){
// Something went wrong
alert("Your browser broke!");
return false;
}
}
}
ajaxRequest.onreadystatechange = function()
{
if (ajaxRequest.readyState == 4)
{
currentuser = window.location.href;
pos = currentuser.search("wp_username=");
username = currentuser.substring(pos+12,pos+16);
}
}
ajaxRequest.open("GET","state.php?s="+status+"&p="+peername+"&v="+username,true);
ajaxRequest.send();
});

```

The username variable will contain the caller user, the peername variable will contain the called party and the status variable will contain their current status. They are sent to the state.php page through AJAX. There is performed the users status update, please find below the PHP source code:

//here is missing the connection to DB, which is described above in this tutorial

```

$status = $_GET['s'];
$peername = $_GET['p'];
$caller = $_GET['v'];

$sql="UPDATE tb_users SET state='$status' where sipusername='$peername' or sipusername='$caller'";

if (mysqli_query($conn, $sql)) {
    echo "OK";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
mysqli_close($conn);

```

How to use Mizu webphone for call center purposes

In the above example we have successfully implemented the Mizu webphone into a PHP based website.

Now let's complicate a little bit the things. Suppose that we need to provide a ready to use solution for an outgoing call center, where the agents need to be as efficient as possible.

The Mizu webphone will automatically call (in a configurable time interval) the customers from a predefined database and will offer informations for the agents about the called customers (name, age, address, interests, etc.)

and also the agents will be able to save (to the database) by a simple click their conversation result. (if the called customer wants to buy the promoted product or not, the customer is agree to participate to the advertised event or not etc.)

In our example the customers will be stored in the tb_customers table.

You can easily create it by running the following query:

```

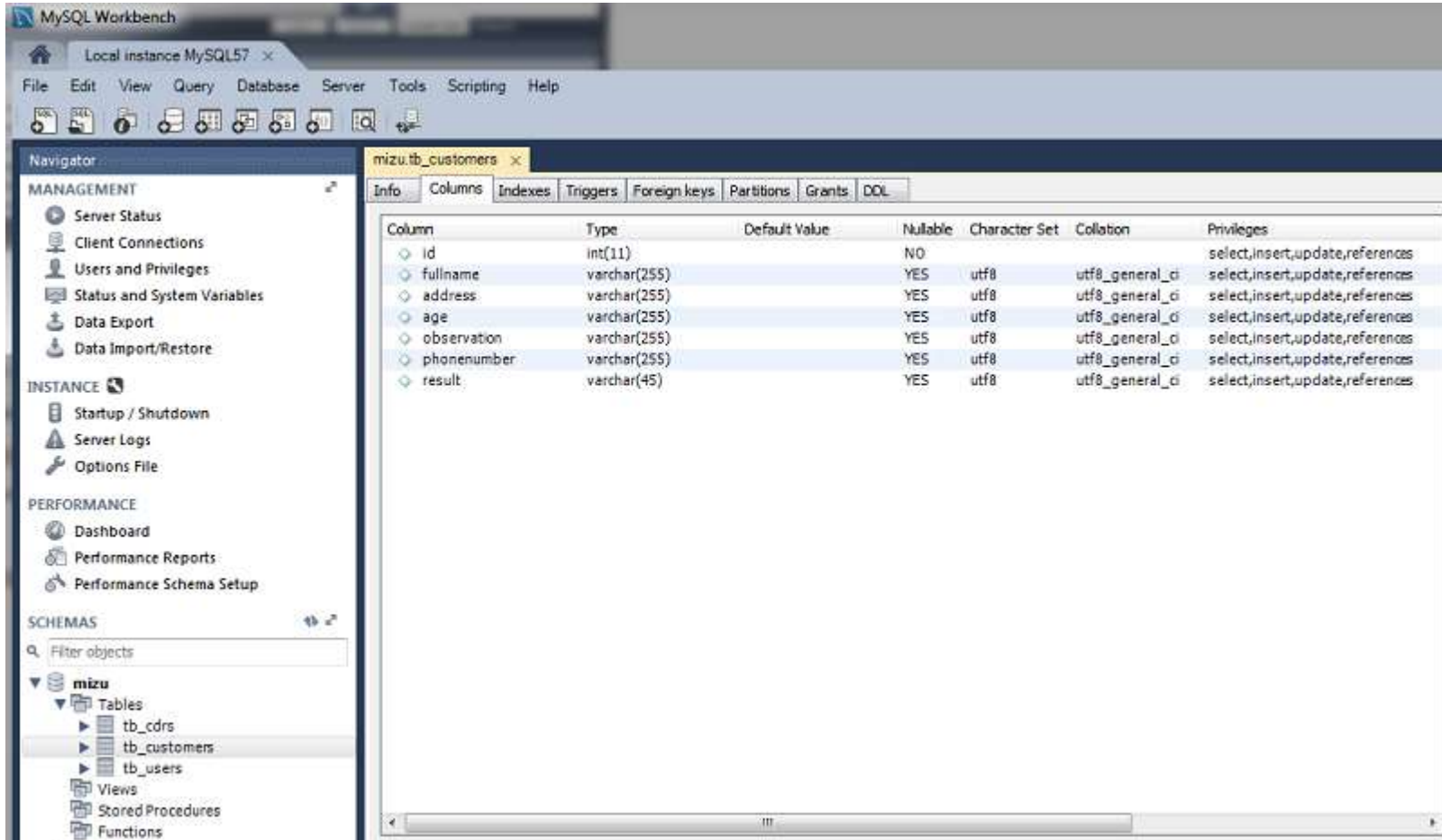
CREATE TABLE `tb_customers` (
  `id` int(11) NOT NULL AUTO_INCREMENT,

```

```

`fullname` varchar(255) DEFAULT NULL,
`address` varchar(255) DEFAULT NULL,
`age` varchar(255) DEFAULT NULL,
`observation` varchar(255) DEFAULT NULL,
`phonenummer` varchar(255) DEFAULT NULL,
`result` varchar(45) DEFAULT NULL,
PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8

```



The whole mechanism will begin, when the agent will press the “Start” button, on which the `predictivedialer()` javascript function is called:

```

function predictivedialer()
{
  StartCalls();
  myVar = setInterval(alertFunc, 10000);
}

function alertFunc()
{
  if (webphone_api.isincall() === false)
  {
    document.getElementById('callresult').value="";
    document.getElementById('customer').innerHTML="";
    StartCalls();
  }
}

```

In the `predictivedialer` function is called the `StartCalls` function (about it we will discuss later) and in every 10 seconds (10000 ms) is called the `alertFunc` function.

In case that the webphone is not in call (we will know it from the isincall function), we will empty the input field where will be stored the result about the next call and the textarea, where will be displayed the informations about the next called customer.

Sonnyke phone

Speaking

SIP Server address: 88.150.183.67

Websocket address: Websocket address

ICE Servers: ICE Servers

Username: 1122

Display name: Displayname

Password: *****

Register

Start

Destination: Call to

Result: callresult

Save

Call Hangup Video Chat

Last called customer
Name:Nagy Vilma
Address:Kapornyas, Allomas utca 10 szam
Age:56
Observation:Shopping
Phone number:3333

Below you can find the source code of the Startcalls js function. By AJAX method it will get the necessary informations about the next called customer (from the customers.php), this infos will be displayed into a textarea (with id=testarea) and the user will be called (webphone_api.call(username) function).

```
function StartCalls()
{
    var ajaxRequest; // The variable that makes Ajax possible!
    var result,pos,len,username;
    try
    {
        // Opera 8.0+, Firefox, Safari
        ajaxRequest = new XMLHttpRequest();
    }catch (e)
    {
        // Internet Explorer Browsers
        try
        {
            ajaxRequest = new ActiveXObject("Msxml2.XMLHTTP");
        }
        catch (e)
        {
            try
            {
                ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
            }
            catch (e)
            {
                // Something went wrong
                alert("XMLHttpRequest is not supported");
            }
        }
    }
    // Create the XMLHttpRequest object
    ajaxRequest.open("GET", "customers.php", true);
    // Send the request
    ajaxRequest.send();
    // Get the response
    result = ajaxRequest.responseText;
    // Parse the response
    pos = result.indexOf("Name:");
    len = result.indexOf("Address:");
    username = result.substring(pos+7, len);
    // Call the user
    webphone_api.call(username);
    // Display the customer information
    document.getElementById("testarea").innerHTML = result;
}
```



```

{
    ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
} catch (e){
    // Something went wrong
    alert("Your browser broke!");
    return false;
}}
ajaxRequest.onreadystatechange = function()
{
    if (ajaxRequest.readyState == 4)
    {
        document.getElementById('testarea').innerHTML = ajaxRequest.responseText;
        result = ajaxRequest.responseText;
        pos = result.search("Phone number:");
        len = result.length;
        username = result.substring(pos+13,len);
        document.getElementById('ccc').innerHTML = username;
        webphone_api.call(username);
    }
}
ajaxRequest.open("GET","customers.php",true);
ajaxRequest.send();
}

```

Here you can find the source code of the customers.php, where is randomly selected a customer from the tb_customers table and his informations are displayed for the call center agent.

```

//here is missing the connection to DB, which is described above in this tutorial
$sql = "SELECT * FROM tb_customers ORDER BY rand() LIMIT 1;";
$result = mysqli_query($conn, $sql);

```

```

if (mysqli_num_rows($result) > 0)
{
    while($row = mysqli_fetch_assoc($result)) {

        echo "Last called customer";
        echo "\r\n";
        echo "Name:" . $row["fullname"];
        echo "\r\n";
        echo "Address:" . $row["address"];
        echo "\r\n";
        echo "Age:" . $row["age"];
        echo "\r\n";
        echo "Observation:" . $row["observation"];
        echo "\r\n";
        echo "Phone number:" . $row["phonenumber"];

    }
} else {
    echo "LOGIN FAILED";
}
mysqli_close($conn);

```

In the last part of this tutorial we will discuss about the call result storage. It is performed by the SaveResults js function, which is sending by AJAX the result and customers phonenumber to the result.php, where the tb_customers table is updated with the transmitted values.

Javascript source code:

```

function SaveResults() { //saving the call result
    var ajaxRequest; // The variable that makes Ajax possible!

```



```

var phonenumber;
var result;
try
{
// Opera 8.0+, Firefox, Safari
ajaxRequest = new XMLHttpRequest();
} catch (e)
{
// Internet Explorer Browsers
try
{
ajaxRequest = new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
try
{
ajaxRequest = new ActiveXObject("Microsoft.XMLHTTP");
} catch (e){
// Something went wrong
alert("Your browser broke!");
return false;
}
}
}
ajaxRequest.onreadystatechange = function()
{
if (ajaxRequest.readyState == 4)
{
document.getElementById('customer').innerHTML = ajaxRequest.responseText;
}
}
result = document.getElementById('callresult').value;
phonenumber = document.getElementById('ccc').innerHTML;
ajaxRequest.open("GET", "result.php?q="+result+"&w="+phonenumber,true);
ajaxRequest.send();
}

```

PHP source code:

```

$result = $_GET['q'];
$phonenr = $_GET['w'];

$sql="UPDATE tb_customers SET result='$result' where phonenumber='$phonenr'";

if (mysqli_query($conn, $sql)) {
    echo "The result was saved successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}

```

The source code for the above examples can be downloaded from [here](#).