CloudPBX API

I metodi

metodo: **user login**

questo metodo, date due variabili, restituisce esito positivo e token o esito negativo al login e apertura websocket.
variabili: username, password
esempio:
["user login", {username: "11", password: "1"}]
output in JSON:
["user token", {result: true, token: 4747e18efef3c521a8eb242bf2424a4997b2670ea95f7e8bfb57fb3c5b7c915cf8b1b80ac0ff43045b7b59337f56bc5649b895a0b836a7f68c800d06d1f86759, username: 11}]

nota: nell'utilizzo dei successivi metodi, non è richiesto il token in quanto transitano nello stesso websocket.

== == == == ==

metodo: **CallLog**

questo metodo, date le due variabili, torna un l'elenco delle chiamate effettuate e ricevute dall'interno che ha effettuato la richiesta, nel lasso di tempo definito dalle due variabili passate
variabili : DataStart, DataEnd
esempio:
{ "method":"CallLog", "DataStart":"2016-01-01 10:00:00", "DataEnd":"2016-01-01 20:00:00"}
output in JSON:
[{"callid":8286318,"caller":"340123456","called":"2551","begin":"2016-09-30T09:05:39.000Z","duration":5,"billsec":0},{"callid":8286321,"caller":"11","called":"340123456","begin":"2016-09-30T09:27:05.000Z","duration":9,"billsec":0}]

== == == == ==

metodo: **Dial**

questo metodo genera la chiamata in uscita verso il numero segnalato nella variabile

variabili: Number
esempio:
{ "method":"Dial", "number":"05711738000"}

Dopo avere effettuato il login, sul socket attivato vengono pubblicati i messaggi che tracciano le chiamate in ingresso ricevute dall'interno. I messaggi pubblicati sono i seguenti: Ringing (quando l'interno monitorato inizia a squillare).
output in JSON:
{caller: "3401123456", called: "11", state: "5", statedesc: "Ringing"}

Lato client

abbiamo sviluppato, per test, anche la parte client dell'API;
Un web panel è raggiungibile all'indirizzo https://my.cloudpbx.it:30+idpx (es https://my.cloudpbx.it:30504/)
Si accede con user il numero short dell'interno (nell'esempio dell id pbx 504 e quindi dell'interno 50416, lo user per accedere alle API è 16) e come password la password WEB

Nel riquadro è possibile scrivere i metodi e cliccare sul pulsante SEND

Qui di seguito il codice che può aiutare il cliente a sviluppare il suo lato

main.js

```
$(function() {
  var FADE_TIME = 150; // ms
  var token = false;
  var username = null;
  // Initialize variables
  var $window = $(window);
  // Input for username
  var $usernameInput = $('.usernameInput');
  // Messages area
  var $messages = $('.messages');
  // Input message input box
  var $inputMessage = $('.inputMessage');
  // The login page
  var $loginPage = $('.login.page');
  // The chatroom page
  var $chatPage = $('.chat.page');
  var socket = io.connect('http://my-dev.cloudpbx.it:30504/');
  // Adds a message element to the messages and scrolls to the bottom
```

```
  // el - The element to add as a message
  // options.fade - If the element should fade-in (default = true)
  // options.prepend - If the element should prepend
  //   all other messages (default = false)
  function addMessageElement (el, options) {
    var $el = $(el);
    // Setup default options
    if (!options) {
      options = {};
    }
    if (typeof options.fade === 'undefined') {
      options.fade = true;
    }
    if (typeof options.prepend === 'undefined') {
      options.prepend = false;
    }
    // Apply options
    if (options.fade) {
      $el.hide().fadeIn(FADE_TIME);
    }
    if (options.prepend) {
      $messages.prepend($el);
    } else {
      $messages.append($el);
    }
    $messages[0].scrollTop = $messages[0].scrollHeight;
  }
  function printDataNewstate(data) {
    console.log(data);
    var $el = '';
    $el += `caller: ${data.caller}<br/>`;
    $el += `called: ${data.called}<br/>`;
    $el += `state: ${data.state}<br/>`;
    $el += `statedesc: ${data.statedesc}<br/>`;
//    $el += `uniqueid: ${data.uniqueid}<br/>`;
    $el += `<hr><br/><br/>`;
    $messages.append($el);
  }
  function printDataHangup(data) {
    console.log(data);
    var $el = '';
    $el += `caller: ${data.caller}<br/>`;
    $el += `called: ${data.called}<br/>`;
    $el += `statedesc: Hangup<br/>`;
    $el += `cause: ${data.cause}<br/>`;
    $el += `causedesc: ${data.causedesc}<br/>`;
//    $el += `uniqueid: ${data.uniqueid}<br/>`;
    $el += `<hr><br/><br/>`;
    $messages.append($el);
  }
  function printDataCallLog(data) {
    console.log(data);
//    var output = JSON.parse(data);
```

```
      var output = data;
//    console.log(output);
//    $messages.append(data);
    output.rows.forEach(function(element){
      var $el = '';
      $el += `callid: ${element.callid}<br/>`;
      $el += `caller: ${element.caller}<br/>`;
      $el += `called: ${element.called}<br/>`;
      $el += `begin: ${element.begin}<br/>`;
      $el += `duration: ${element.duration}<br/>`;
      $el += `billsec: ${element.billsec}<br/>`;
      $el += `<hr><br/>`;
      $messages.append($el);
    });
  }
/*  function filterByUsername(data){
    if(data.called === username){
      return true;
    }
    return false;
  }
  function filterByAccountCode(data){
    if(data.accountcode === ("504-" + username)){
      return true;
    }
    return false;
  }
  function filterByChannel(data){
    if(data.channel.includes("SIP/504" + username)){
      return true;
    }
    return false;
  }
*/
  function registerListeners() {
    socket.on('pbx Newstate', function (data) {
//      if(filterByUsername(data) && filterByChannel(data)){
        printDataNewstate(data);
//      }
    });
    socket.on('pbx Hangup', function (data) {
//      if(filterByUsername(data) && filterByAccountCode(data)){
        printDataHangup(data);
//      }
    });
    socket.on('pbx CallLog', function (data) {
      printDataCallLog(data);
    });
  }
  socket.on('user token', function (data) {
    if(data.result) {
      token = data.token;
      $loginPage.fadeOut();
```

```
      $chatPage.show();
      $loginPage.off('click');
      $currentInput = $inputMessage.focus();
      username = data.username;

      registerListeners();
    }
  });
  // Focus input when clicking anywhere on login page
  $(".submitInput").click(function () {

//    console.log($(".usernameInput").val());
//    console.log($(".passwordInput").val());
    socket.emit('user login', {
      username: $(".usernameInput").val(),
      password: $(".passwordInput").val()
    });
  });
   $(".submitMessage").click(function () {
    var action = JSON.parse($inputMessage.val())
    socket.emit('pbx action', action);
  });
});
```

index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CloudPBX Example</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <ul class="pages">
    <li class="chat page">
      <div class="chatArea">
        <ul class="messages"></ul>
      </div>
      <div style="float:left; width:90%;">
        <!-- textarea class="inputMessage" placeholder="Type
here..."></textarea -->
        <textarea class="inputMessage" placeholder="Type
here...">{"method":"CallLog", "DataStart":"2017-02-24 00:00:00",
"DataEnd":"2017-02-28 00:00:00"}</textarea>
      </div>
      <div style="float:left;  width:10%;">
        <input class="submitMessage" type="button" value="Send"/>
      </div>
    </li>
    <li class="login page">
      <div class="form">
        <h4 class="title">Username</h4>
        <input class="usernameInput" type="text" maxlength="14" />
```

```
        <h4 class="title">Password</h4>
        <input class="passwordInput" type="password" maxlength="14" />
        <br />
        <input class="submitInput" type="button" value="Send"/>
      </div>
    </li>
  </ul>
  <script src="https://code.jquery.com/jquery-1.10.2.min.js"></script>
  <script src="<%= ENV.PROT %>://<%= ENV.DNS %>:<%= ENV.PORT
%>/socket.io/socket.io.js"></script>
  <script type="text/javascript">
    var PROT = <%- JSON.stringify(ENV.PROT) %>;
    var DNS = <%- JSON.stringify(ENV.DNS) %>;
    var PORT = <%- JSON.stringify(ENV.PORT) %>;
  </script>
  <script src="main.js"></script>
</body>
</html>
```

style.css

```
/* Fix user-agent */
* {
  box-sizing: border-box;
}
html {
  font-weight: 300;
  -webkit-font-smoothing: antialiased;
}
html, input {
  font-family:
    "HelveticaNeue-Light",
    "Helvetica Neue Light",
    "Helvetica Neue",
    Helvetica,
    Arial,
    "Lucida Grande",
    sans-serif;
}
html, body {
  height: 100%;
  margin: 0;
  padding: 0;
}
ul {
  list-style: none;
  word-wrap: break-word;
}
/* Pages */
.pages {
  height: 100%;
  margin: 0;
  padding: 0;
  width: 100%;
}
.page {
```

```
  height: 100%;
  position: absolute;
  width: 100%;
}
/* Login Page */
.login.page {
  background-color: #000;
}
.login.page .form {
  height: 100px;
  text-align: center;
  top: 50%;
  width: 100%;
}
.login.page .form .usernameInput, .login.page .form .passwordInput {
  background-color: transparent;
  border: none;
  border-bottom: 2px solid #fff;
  outline: none;
  padding-bottom: 15px;
  text-align: center;
  width: 400px;
}
.login.page .title {
  font-size: 200%;
}
.login.page .usernameInput, .login.page .passwordInput {
  font-size: 200%;
  letter-spacing: 3px;
}
.login.page .title, .login.page .usernameInput, .login.page .passwordInput
{
  color: #fff;
  font-weight: 100;
}
.submitInput {
  margin-top: 50px;
  padding: 10px;
  font-size: 150%;
  letter-spacing: 3px;
  border: none;
  border-bottom: 2px solid #fff;
}
/* Chat page */
.chat.page {
  display: none;
}
/* Font */
.messages {
  font-size: 100%;
}
.inputMessage {
  font-size: 100%;
}
.log {
  color: gray;
  font-size: 70%;
```

```
  margin: 5px;
  text-align: center;
}
/* Messages */
.chatArea {
  height: 80%;
  padding-bottom: 60px;
}
.messages {
  height: 100%;
  margin: 0;
  overflow-y: scroll;
  padding: 10px 20px 10px 20px;
}
.message.typing .messageBody {
  color: gray;
}
.username {
  font-weight: 700;
  overflow: hidden;
  padding-right: 15px;
  text-align: right;
}
/* Input */
.inputMessage {
  border: 10px solid #000;
  bottom: 0;
  height: 160px;
  left: 0;
  outline: none;
  padding-left: 10px;
  right: 0;
  width: 100%;
}
.submitMessage {
  width: 100%;
  height: 160px;
}
```