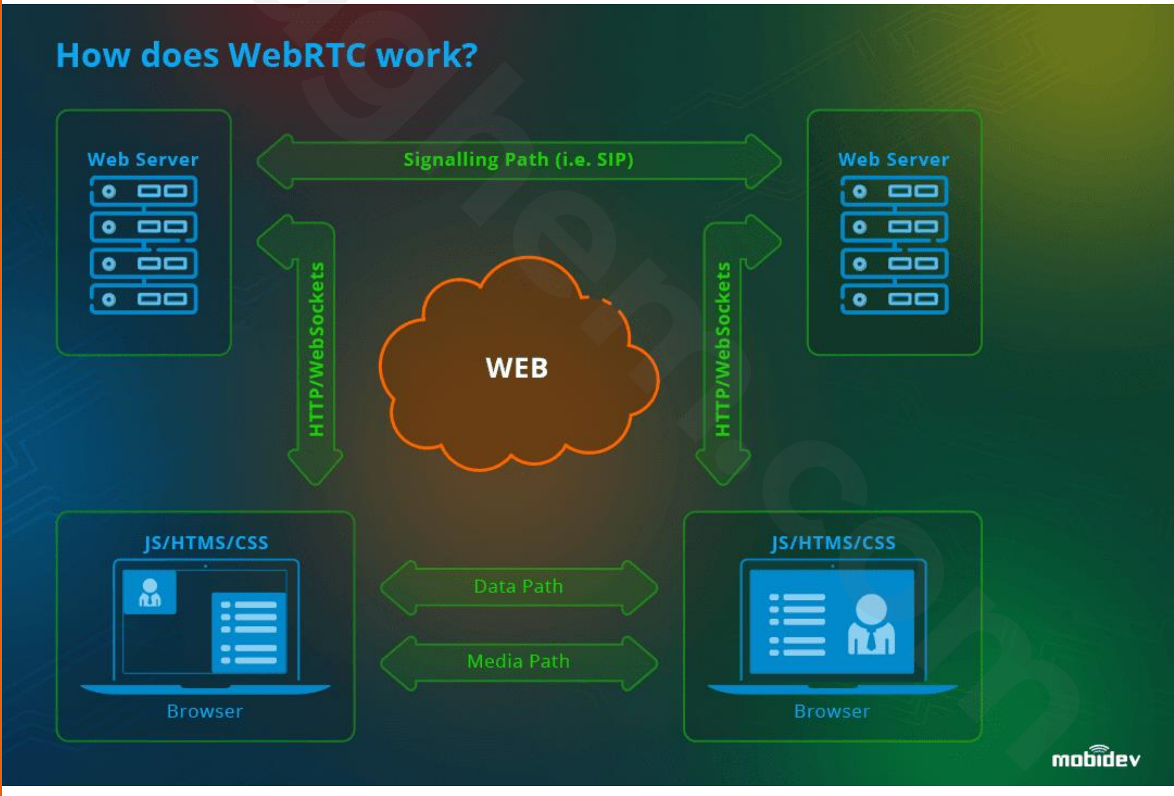
TELEPHONY OVER IP SOLUTION

Travaux pratiques – 2

WebRTC

Classe : E4CCSN

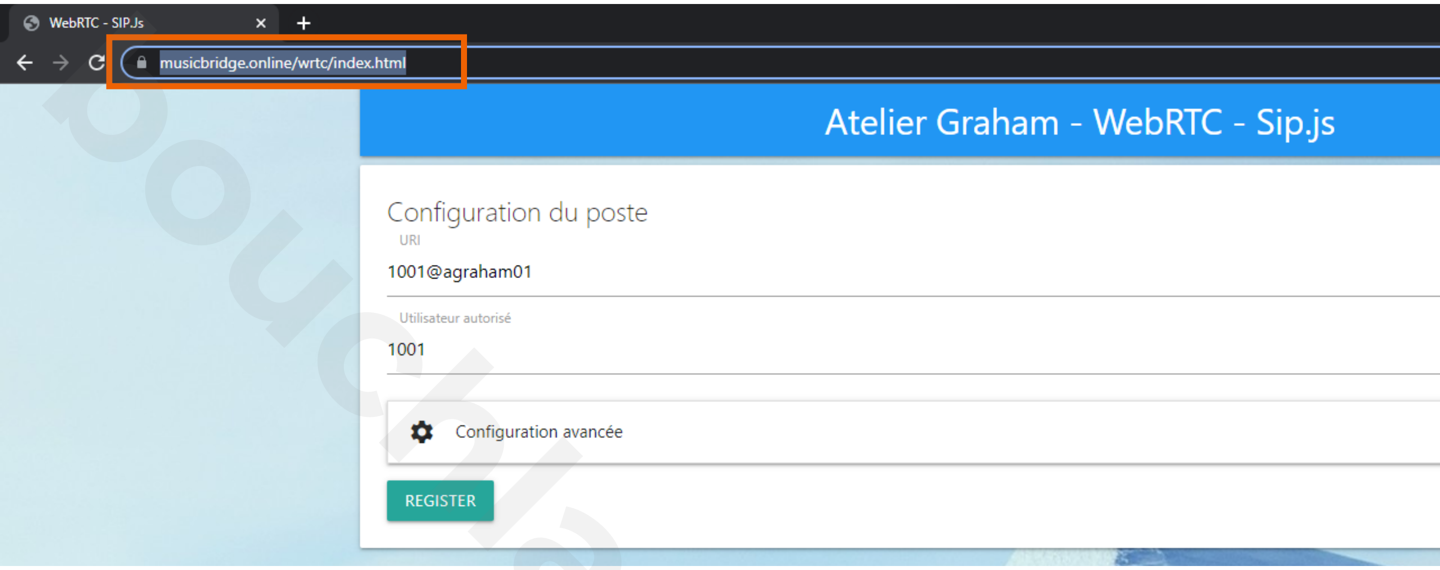


Réalisé par : Mohammed BOUCHLAGHEM

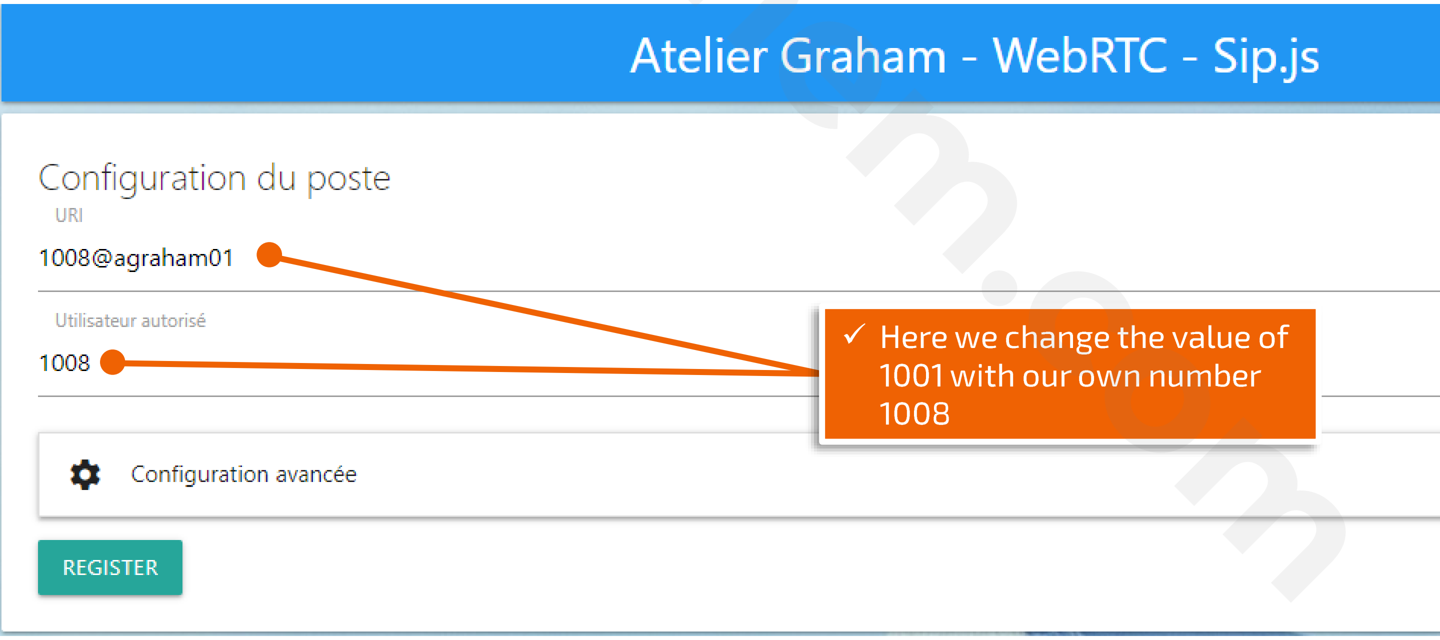
1) launching and configuration :

1.1 Lunching Firefox and browsing this link:

https://musicbridge.online/wrtc/index.html

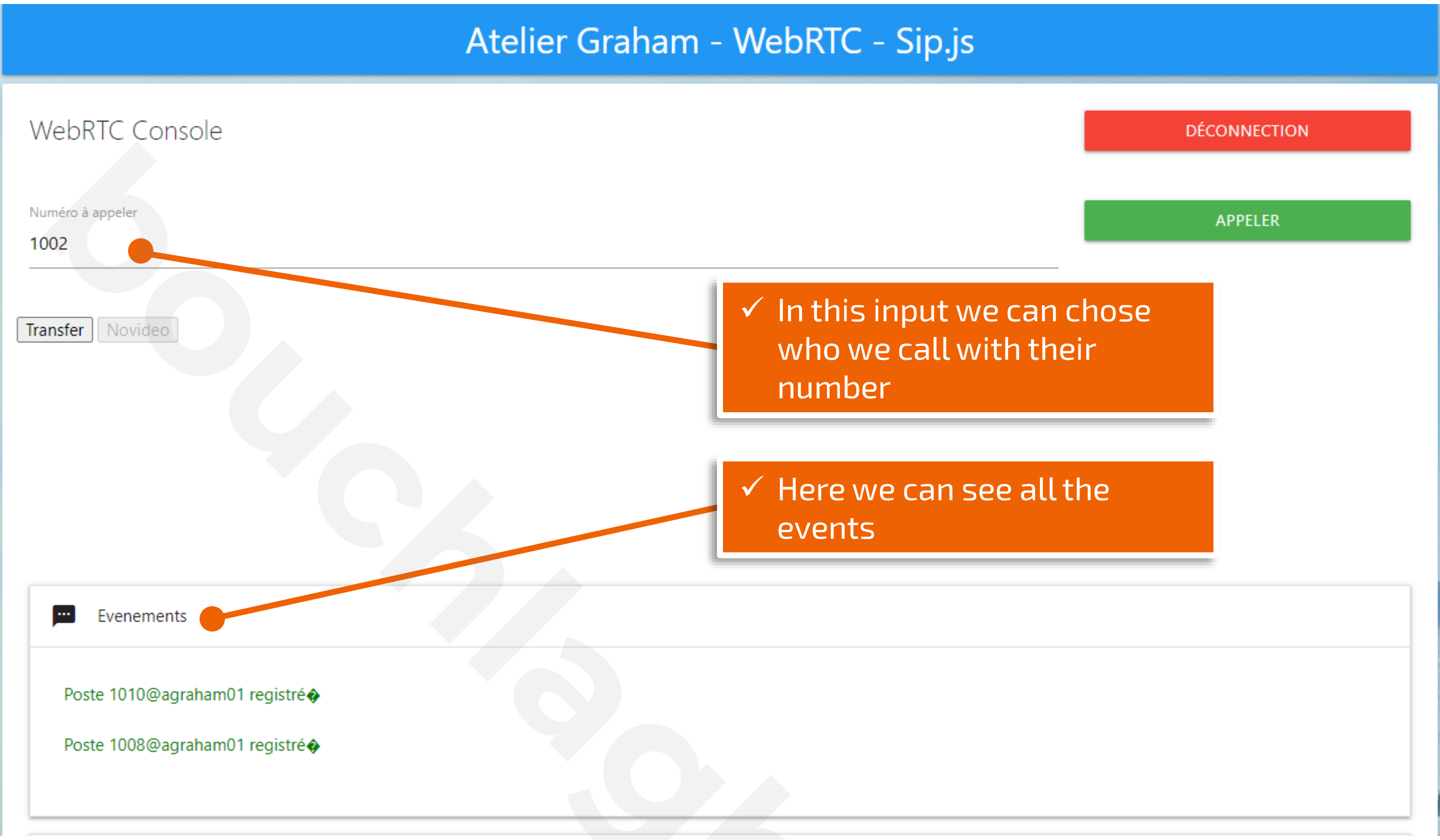


1.2 Login using our account username :

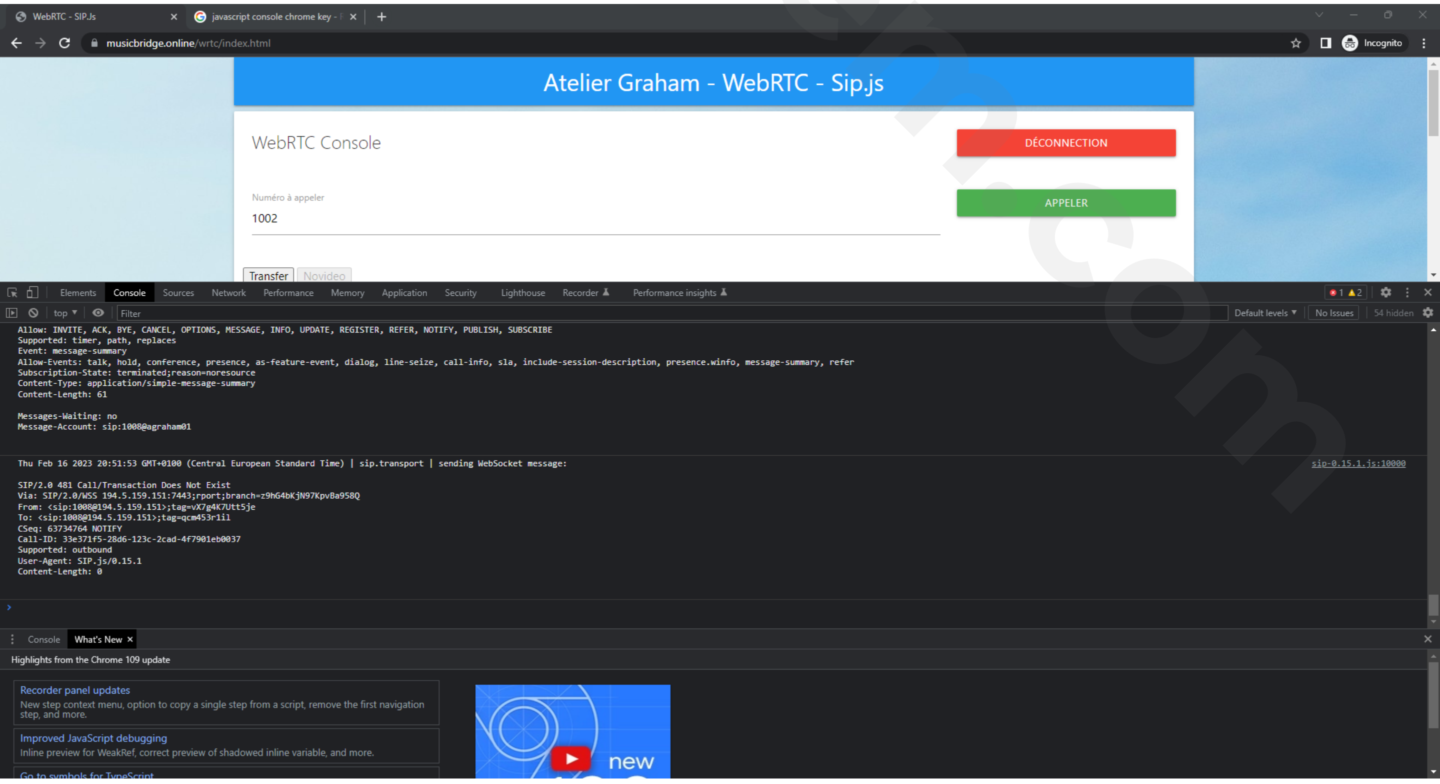


1) launching and configuration :

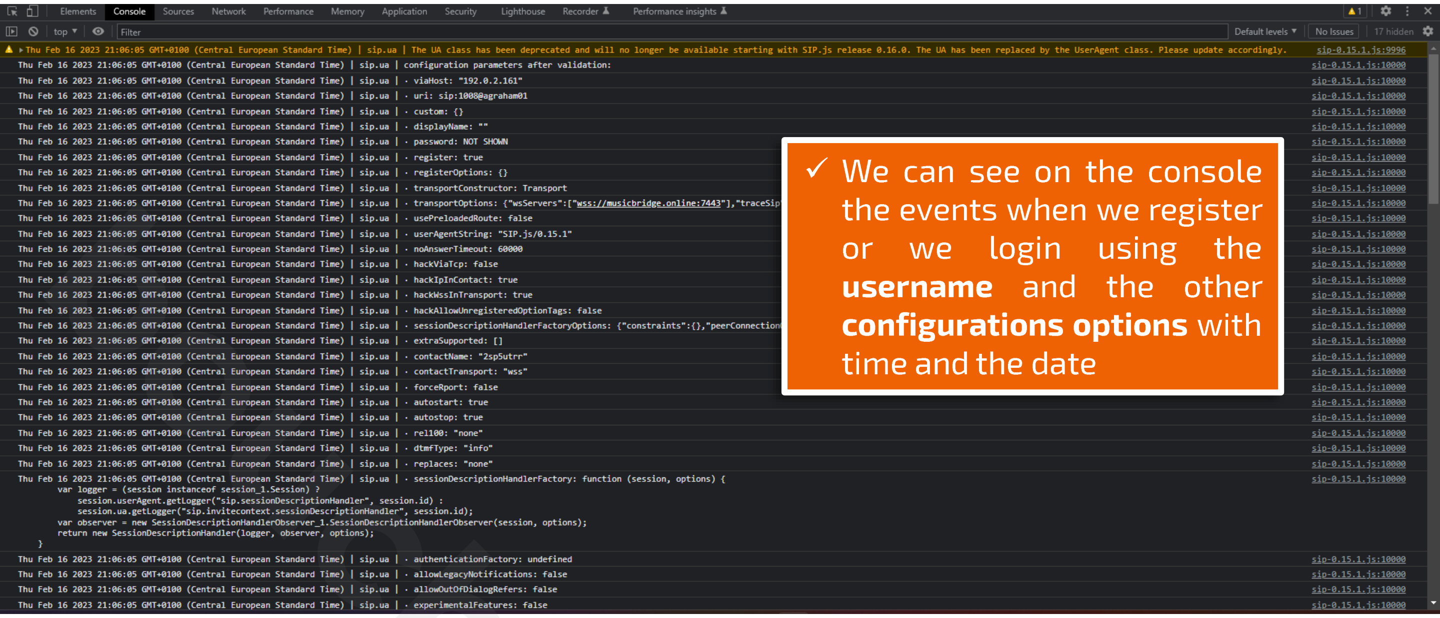
1.3 First look into the WebRTC Console :



1.4 To access to the JavaScript Console we press : Ctrl + Shift + J



2) Observation of the console :



2) Observation of the source codes :

DevTools - musicbridge.online/wrtc/index.html

Elements | Console | Sources | Network | Performance | Memory | Application | Security | Lighthouse | Recorder | Performance insights

Page | Filesystem

- top
  - musicbridge.online
    - wrtc
      - Images
      - ① index.html
      - renderer.js
      - ② sip-0.15.1.js
  - ajax.googleapis.com
  - cdnjs.cloudflare.com
  - fonts.googleapis.com
  - fonts.gstatic.com

sip-0.15.1.js | index.html ×

```html
            <!--Answer / Refuse call-->
            <div class="row">
                    <a id="yes_answer_btn" class="waves-effect waves-light btn disabled green col s5 hide">Accepter appel</a>
                        <div class="col s2"></div>
                    <a id="no_answer_btn" class="waves-effect waves-light btn disabled red col s5 hide">Refuser appel</a>
            </div>

            <!--Hang Up Call-->
            <div id="hangupRow" class="row hide">
                    <a id="hangup_btn" class="waves-effect waves-light btn red col s4 offset-s4" >Raccrocher</a>

            </div>

            <!-- Adio/Video elements for calls -->
            <div class="row">
                <video id="remoteVideo" class="show"></video>
                    <div class="col s2"></div>
                <video id="localVideo" class="show"></video>
            </div>

            <!-- Global logs -->
            <ul class="collapsible">
                    <li class="active">
                      <div class="collapsible-header"><i class="material-icons">textsms</i>Evenements</div>
                        <div class="collapsible-body">
                                <div id="phone_log" style="max-height:200px;overflow:auto;">
                                </div>
                        </div>
                    </li>
            </ul>

            <!-- Log SIP -->
            <ul class="collapsible">
                    <li>
                      <div class="collapsible-header"><i class="material-icons">textsms</i>Log SIP</div>
                        <div class="collapsible-body">
                                <div id="log_display" style="max-height:200px;overflow:auto;">
                                </div>
                        </div>
                    </li>
            </ul>

        </div>
      </div>
    </div>


  </div>

  <!--Include JS script-->
  <script src="./renderer.js"></script>
② <script src="./sip-0.15.1.js"></script>

</body>

</html>
```

HTML File

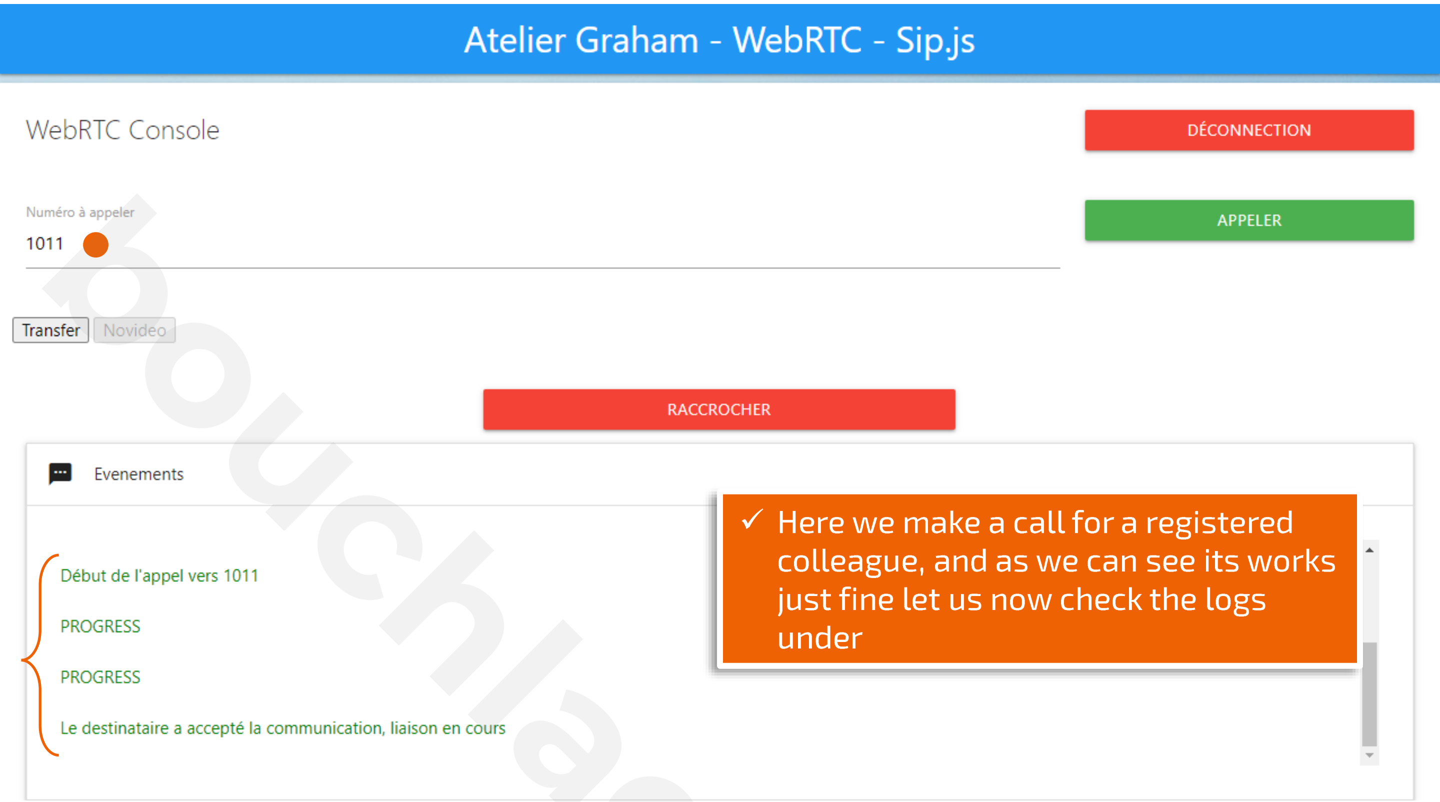
- ✓ Here we have the source code :
- ✓ **HTML (index.html) :** the html code ① that implements the difference options and interactions
- ✓ And in the HTML file there is a part of that include a **JavaScript** file source : **(./sip-0-15.1.js").** ②

sip-0.15.1.js × | index.html

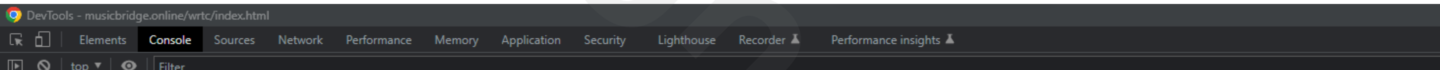
```javascript
     * Parse the given header on the given index.
     * @param name - header name
     * @param idx - header index
     * @returns Parsed header object, undefined if the
     *   header is not present or in case of a parsing error.
     */
    IncomingMessage.prototype.parseHeader = function (name, idx) {
        if (idx === void 0) { idx = 0; }
        name = utils_1.headerize(name);
        if (!this.headers[name]) {
            // this.logger.log("header '" + name + "' not present");
            return;
        }
        else if (idx >= this.headers[name].length) {
            // this.logger.log("not so many '" + name + "' headers present");
            return;
        }
        var header = this.headers[name][idx];
        var value = header.raw;
        if (header.parsed) {
            return header.parsed;
        }
        // substitute '-' by '_' for grammar rule matching.
        var parsed = grammar_1.Grammar.parse(value, name.replace(/-/g, "_"));
        if (parsed === -1) {
            this.headers[name].splice(idx, 1); // delete from headers
            // this.logger.warn('error parsing "' + name + '" header field with value "' + value + '"');
            return;
```

- ✓ The JavaScript file contain all the functions that make the website dynamic and interactive

3) Call a registered colleague :



✓ Here we make a call for a registered colleague, and as we can see its works just fine let us now check the logs under

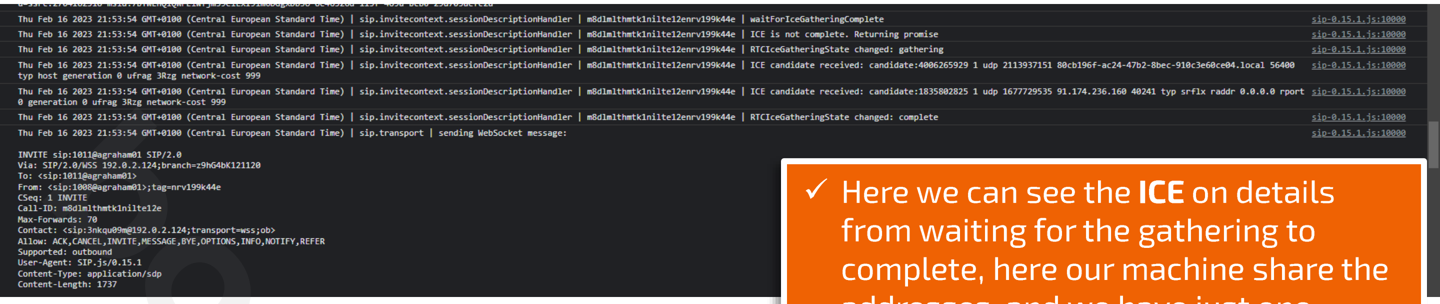



```
Thu Feb 16 2023 21:54:19 GMT+0100 (Central European Standard Time) | sip.transport | received WebSocket text message:

SIP/2.0 200 OK
Via: SIP/2.0/WSS 192.0.2.124;branch=z9hG4bK7316876;received=91.174.236.160;rport=38447
From: <sip:1008@agraham01>;tag=nrv199k44e
To: <sip:1011@agraham01>;tag=6SQ40y7UNHK3D
Call-ID: m8dlmlthmtk1nilte12e
CSeq: 2 INVITE
Contact: <sip:1011@194.5.159.151:5060;transport=udp>
User-Agent: FreeSWITCH-mod_sofia/1.10.8-release.14~64bit
Allow: INVITE, ACK, BYE, CANCEL, OPTIONS, MESSAGE, INFO, UPDATE, REGISTER, REFER, NOTIFY, PUBLISH, SUBSCRIBE
Supported: timer, path, replaces
Allow-Events: talk, hold, conference, presence, as-feature-event, dialog, line-seize, call-info, sla, include-session-description, presence.winfo, message-summary, refer
Content-Type: application/sdp
Content-Disposition: session
Content-Length: 892
Remote-Party-ID: "Outbound Call" <sip:3gvcub37@agraham01>;party=calling;privacy=off;screen=no

v=0
o=FreeSWITCH 1676558025 1676558026 IN IP4 194.5.159.151
s=FreeSWITCH
c=IN IP4 194.5.159.151
t=0 0
a=msid-semantic: WMS DA18QSzNpZM4N1FkXNFL6FqyYwORnkeR
m=audio 22810 UDP/TLS/RTP/SAVPF 111 110
a=rtpmap:111 opus/48000/2
a=fmtp:111 useinbandfec=1; minptime=10
a=rtpmap:110 telephone-event/48000
a=ptime:20
a=fingerprint:sha-256 98:BA:1C:40:5E:A2:3C:1D:DB:28:EB:85:B6:24:3F:87:FD:A0:1C:8B:C5:89:9B:31:C2:88:E9:9D:27:5D:F3:A6
a=setup:active
a=rtcp-mux
a=rtcp:22810 IN IP4 194.5.159.151
a=ice-ufrag:qohnUxLJTqr7AvOb
a=ice-pwd:LttIynWtU41DHK9YZgbKDNjn
a=candidate:9587866711 1 udp 2130706431 194.5.159.151 22810 typ host generation 0
a=end-of-candidates
a=ssrc:1207823003 cname:KpPr7f4n4GiISNTF
a=ssrc:1207823003 msid:DA18QSzNpZM4N1FkXNFL6FqyYwORnkeR a0
a=ssrc:1207823003 mslabel:DA18QSzNpZM4N1FkXNFL6FqyYwORnkeR
a=ssrc:1207823003 label:DA18QSzNpZM4N1FkXNFL6FqyYwORnkeRa0

Thu Feb 16 2023 21:54:19 GMT+0100 (Central European Standard Time) | sip.invitecontext.sessionDescriptionHandler | m8dlmlthmtk1nilte12enrv199k44e | ICE Connection State changed to iceConnectionChecking
Thu Feb 16 2023 21:54:19 GMT+0100 (Central European Standard Time) | sip.invitecontext.sessionDescriptionHandler | m8dlmlthmtk1nilte12enrv199k44e | track added
Thu Feb 16 2023 21:54:19 GMT+0100 (Central European Standard Time) | sip.invite-dialog | INVITE dialog m8dlmlthmtk1nilte12enrv199k44e6SQ40y7UNHK3D sending ACK request
Thu Feb 16 2023 21:54:19 GMT+0100 (Central European Standard Time) | sip.transport | sending WebSocket message:

ACK sip:1011@194.5.159.151:5060;transport=udp SIP/2.0
Via: SIP/2.0/WSS 192.0.2.124;branch=z9hG4bK9835296
To: <sip:1011@agraham01>;tag=6SQ40y7UNHK3D
From: <sip:1008@agraham01>;tag=nrv199k44e
CSeq: 2 ACK
Call-ID: m8dlmlthmtk1nilte12e
Max-Forwards: 70
Supported: outbound
User-Agent: SIP.js/0.15.1
Content-Length: 0

Uncaught (in promise) DOMException: The play() request was interrupted by a new load request. https://goo.gl/LdLk22
Thu Feb 16 2023 21:54:19 GMT+0100 (Central European Standard Time) | sip.invitecontext.sessionDescriptionHandler | m8dlmlthmtk
```

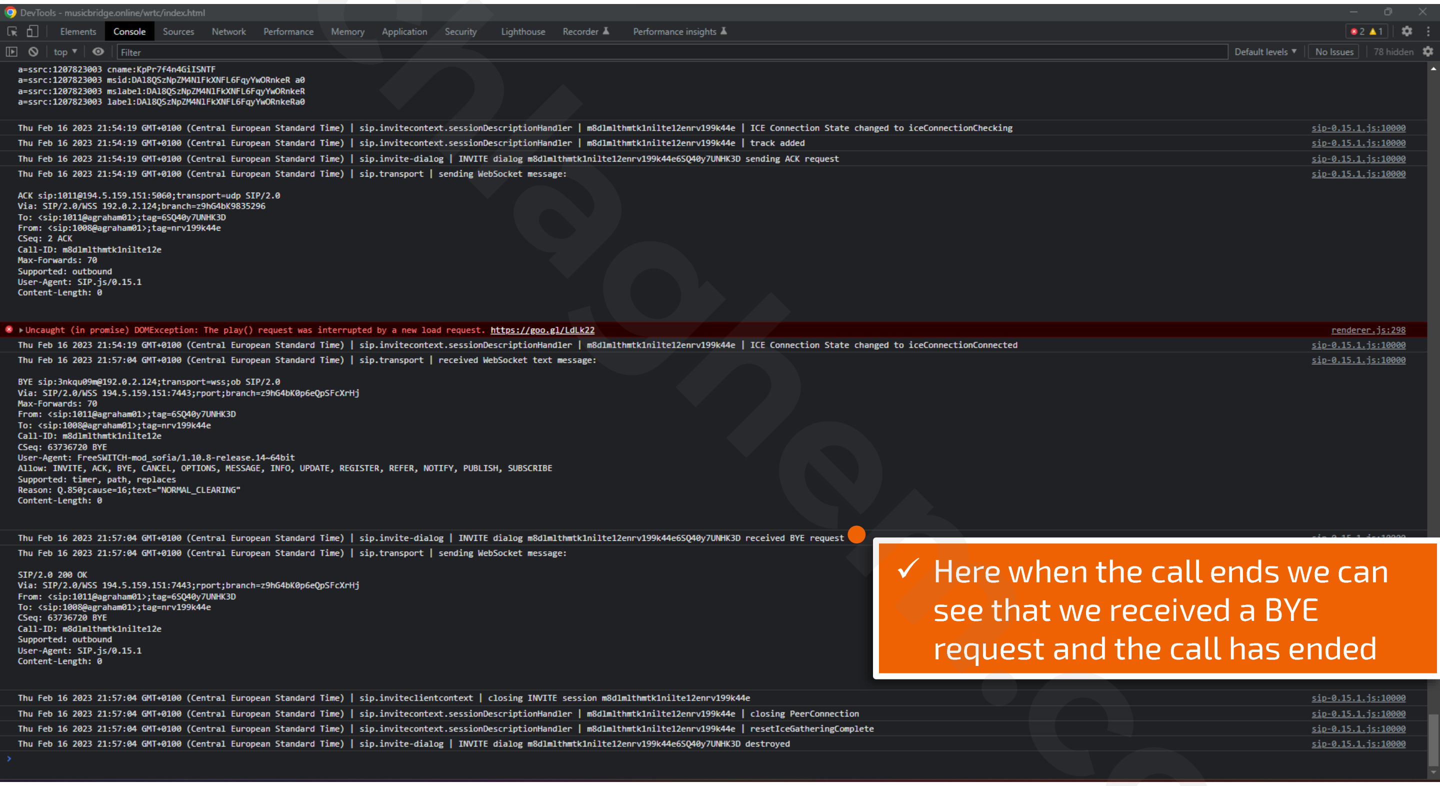
✓ Here we see that the call start via the SIP/2.0/WSS 192.0.124 From 1008 to 1011 and with all the configuration and the codecs that used on the call

✓ The exchange between the server and the client the ICE Connection State and the ACK request with the sent of the WebSocket message

3) Call a registered colleague :



- ✓ Here we can see the **ICE** on details from waiting for the gathering to complete, here our machine share the addresses, and we have just one candidate the server



- ✓ Here when the call ends we can see that we received a BYE request and the call has ended

4) Where does the media flow go ?:

✓ In this schema we have a resume of the media flow or all the exchanges that start from a client that is using a web browser and interacted with the page here the JavaScript works and send it to the server than the server check for the number call if it available than it start the call to the receiver so its goes from our machine to our router than our router route it to the internet in the direction of the web server than to the colleague