JWT Authentication in OpenSIPS

opensips

VOICENTER

About me

VOICENTER

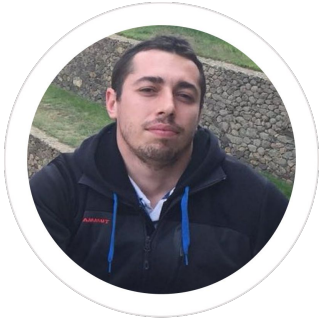
CTO & Founders of Voicenter LTD

Voicenter provides cloud contact center solutions and services for Businesses, enterprises and telcos,
With global coverage using flexible distributed topologies.

VOICENTER ♥ OpenSIPS

More than 15 years of business experience in establishing and managing large scale networks and information technology systems.
Involved with several open source projects.

About me



Opensips Core Developer

Coding, building and integrating OpenSIPS based platforms for more than 10 years

VOICENTER

Agenda

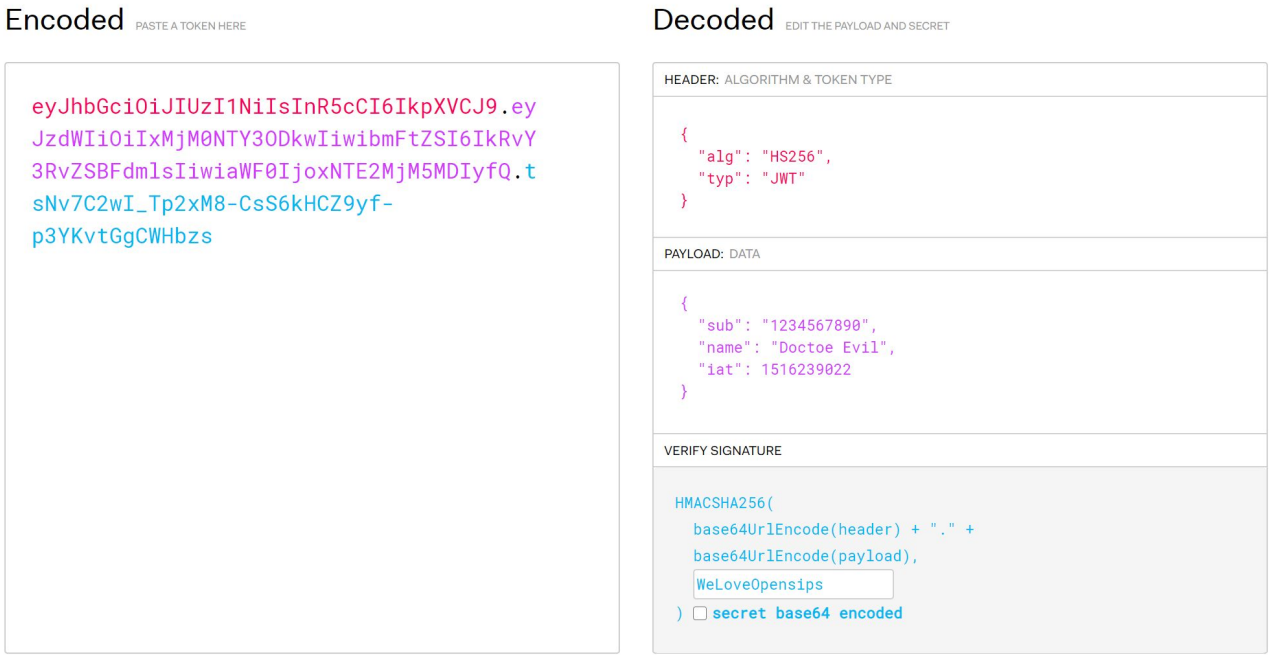
1. Digest Authentication limitations
2. What is JWT ?
3. AUTH_JWT OpenSIPS Module
4. JSSIP JWT implementation
5. Use cases
6. Authentication Provider Integration - OAuth
7. OAuth Demo
8. Questions

Digest Authentication limitations

1. Doing only Authentication not giving us any Authorization
2. Lack of expression
3. MD5 is not secure
4. Subscriber Table sizing problem
5. No safe way for third party auth integration
6. No ability for SSO or Dual-factor authentication.

What is JWT

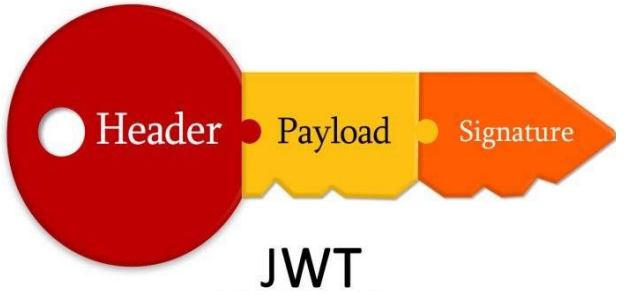
JSON **W**eb **T**okens are an open, industry standard RFC 7519 method for representing claims securely between two parties (jwt.io snapshot)



VOICENTER

What is JWT

A well-formed JWT consists of three concatenated Base64url-encoded strings, separated by dots (.):



1. **JOSE - Header:**
   contains metadata about the type of token and the cryptographic algorithms used to secure its contents.
2. **JWS - payload (set of claims):**
   contains verifiable security statements, such as the identity of the user and the permissions they are allowed.
3. **JWS signature:**
   used to validate that the token is trustworthy and has not been tampered with. When you use a JWT, you must check its signature before storing and using it.

JWT over SIP

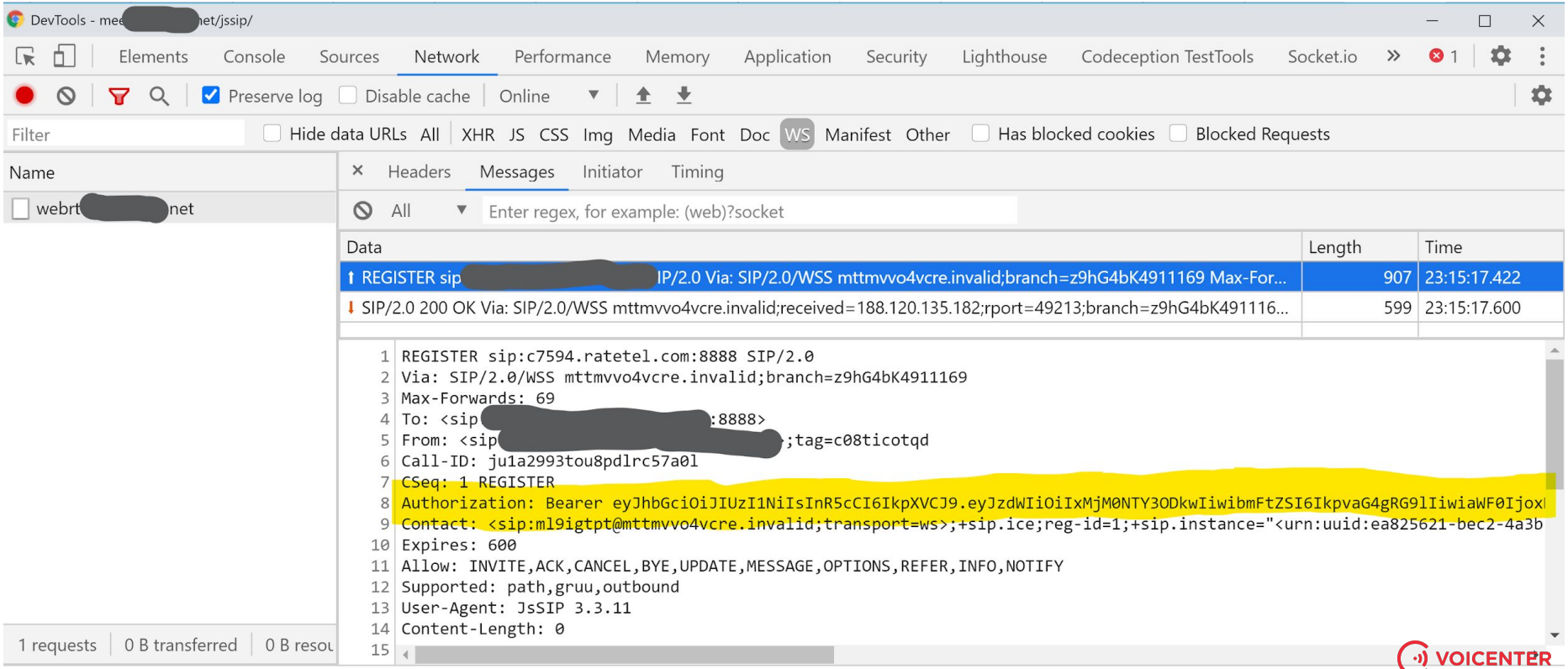
https://tools.ietf.org/id/draft-ietf-sipcore-sip-token-authnz-02.html

SIP Core	R. Shekh-Yusef, Ed.
Internet-Draft	Avaya
Updates: 3261 (if approved)	C. Holmberg
Intended status: Standards Track	Ericsson
Expires: January 8, 2020	V. Pascual
	webrtchacks
	July 7, 2019

Third-Party Token-based Authentication and Authorization for Session Initiation Protocol (SIP)

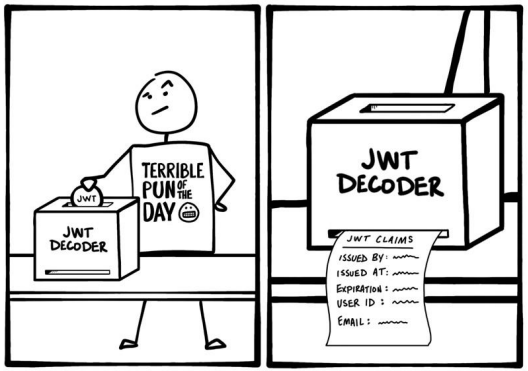
draft-ietf-sipcore-sip-token-authnz-02

JWT over SIP


DevTools - mee[redacted]net/jssip/
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Lighthouse
Codeception TestTools
Socket.io
Preserve log
Disable cache
Online
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
Has blocked cookies
Blocked Requests
Name
webrt[redacted]net
Headers
Messages
Initiator
Timing
All
Enter regex, for example: (web)?socket
Data
Length
Time
REGISTER sip[redacted]IP/2.0 Via: SIP/2.0/WSS mttmvvo4vcre.invalid;branch=z9hG4bK4911169 Max-For...
907
23:15:17.422
SIP/2.0 200 OK Via: SIP/2.0/WSS mttmvvo4vcre.invalid;received=188.120.135.182;rport=49213;branch=z9hG4bK491116...
599
23:15:17.600
1 REGISTER sip:c7594.ratetel.com:8888 SIP/2.0
2 Via: SIP/2.0/WSS mttmvvo4vcre.invalid;branch=z9hG4bK4911169
3 Max-Forwards: 69
4 To: <sip[redacted]:8888>
5 From: <sip[redacted];tag=c08ticotqd
6 Call-ID: ju1a2993tou8pdlrc57a0l
7 CSeq: 1 REGISTER
8 Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0Ijox
9 Contact: <sip:ml9igtpt@mttmvvo4vcre.invalid;transport=ws>;+sip.ice;reg-id=1;+sip.instance="<urn:uuid:ea825621-bec2-4a3b
10 Expires: 600
11 Allow: INVITE,ACK,CANCEL,BYE,UPDATE,MESSAGE,OPTIONS,REFER,INFO,NOTIFY
12 Supported: path,gruu,outbound
13 User-Agent: JsSIP 3.3.11
14 Content-Length: 0
15
1 requests
0 B transferred
0 B resou

AUTH_JWT OpenSIPS modules

- loadmodule “auth_jwt.so”
- modparam("auth_jwt", "db_url","mysql://opensips:opensipsrw@127.0.0.1/opensips")
- modparam("auth_jwt", "profiles_table", "jwt_profiles")
- modparam("auth_jwt", "secrets_table", "jwt_secrets")
- modparam("auth_jwt", "load_credentials", "profile_info")



AUTH_JWT OpenSIPS module - jwt_profiles table

id INT(11)	tag VARCHAR(128)	sip_username VARCHAR(64)	profile_info VARCHAR(2048)
3	space-monkey	tyler.durden	{"FOO": "BAR"}

```
CREATE TABLE opensips.jwt_profiles (
  id int(11) NOT NULL AUTO INCREMENT,
  tag varchar(128) DEFAULT NULL,
  sip username varchar(64) DEFAULT NULL,
  profile info varchar(2048) DEFAULT NULL,
 PRIMARY KEY (id)
)
```

VOICENTER

AUTH_JWT OpenSIPS module - jwt_secrets table

id INT(11)	corresponding_tag VARCHAR(128)	secret VARCHAR(1024)	start_ts INT(11)	end_ts INT(11)
8	space-monkey	ThefirstruleofFightclubisyoudonottalkaboutfig...	1583239007	2084325407

```
CREATE TABLE opensips.jwt_secrets(
     id int(11) NOT NULL AUTO INCREMENT,
     corresponding tag varchar(128) DEFAULT NULL,
     secret varchar(1024) DEFAULT NULL,
     start ts int(11) DEFAULT NULL,
     end ts int(11) DEFAULT NULL,
   PRIMARY KEY (id)
)
```

VOICENTER

AUTH_JWT OpenSIPS module - Usage example

```
modparam("auth_jwt", "tag_claim", "iss")

If ($hdr(Authorization) =~ "Bearer") {

        $var(jwt_token) = $(hdr(Authorization){s.select,1, }{s.trim});

        if (jwt_authorize("$var(jwt_token)",$avp(decoded_jwt),$avp(out_username))) {

                if ($fU != $avp(out_username)) {

                        send_reply(403,"Forbidden SIP ID");

                        exit;

                }

                xlog("JWT auth - profile info = $avp(profile_info) \n");

        }

}
```

HEADER: ALGORITHM & TOKEN TYPE

```
{
  "alg": "RS256",
  "typ": "JWT"
}
```

PAYLOAD: DATA

```
  "iss": "space-monkey",
  "email": "space-monkey@opensips.org",
  "email_verified": true,
  "pstn_dial" : true
}
```

VOICENTER

AUTH_JWT OpenSIPS module - jwt_authorize function

Jwt_authorize (jwt_token**,** out_decoded_token**,** out_sip_username**)**

The function will read the first param (jwt_token) and extract the tag claim

populates the out_decoded_token pvar with the decoded JWT, the out_sip_username with the SIP username corresponding to that JWT profile, and loads all columns listed in the load_credentials modparam.

```
send_reply("401","Unauthorized");
```

VOICENTER

JSSIP JWT implementation

Version 3.4.0 (released in 2020-03-29)

- Add `authorization_jwt` configuration parameter (#610). Credits to @voicenter.


```
var socket = new window.JsSIP.WebSocketInterface( 'wss://webrtc.space-monkey.com.net');

var configuration = {
        sockets  : [ socket ],
        session_timers: false,
         uri      : 'sip:tyler.durden@webrtc.space-monkey.net,
        authorization_jwt :"eyJhbGciOiJIUz.yyyiamnotstupid.notrealtokenxxx"
};

window.ua = new window.JsSIP.UA(configuration);
window.ua.start();
```

VOICENTER

Use cases - How should we use it ?

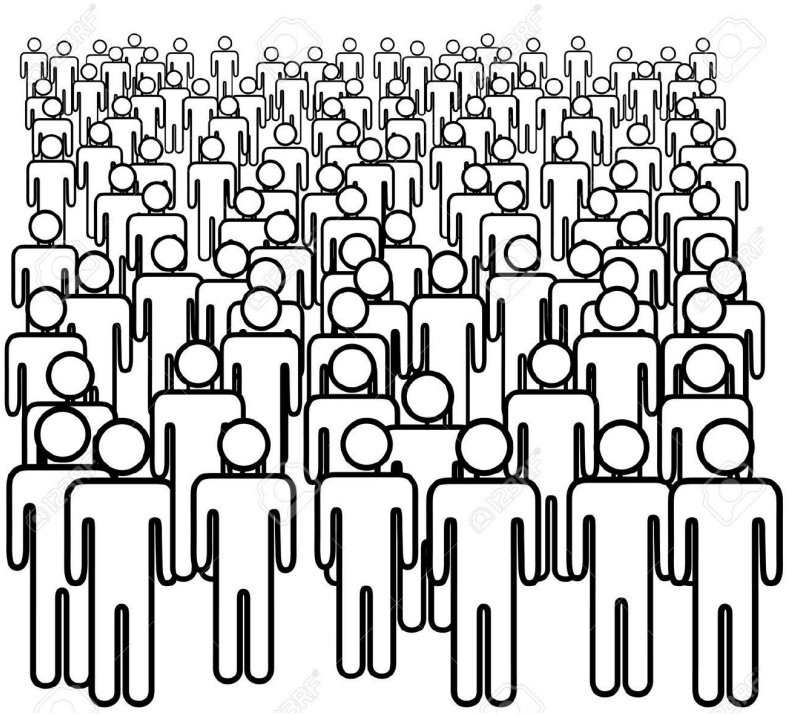
- Webrtc Web systems integration
- Zero provisioning registrations
- S-S-S-O: Sip Single-Sign-On
- Advanced permission management
- Third party permission grant

Existing application SIP integration

- CRM and ERP phone integration
- self service portal: click-to-call customers support
- Mobile app auth mechanism
- online games conferencing integration

Zero provisioning registrations

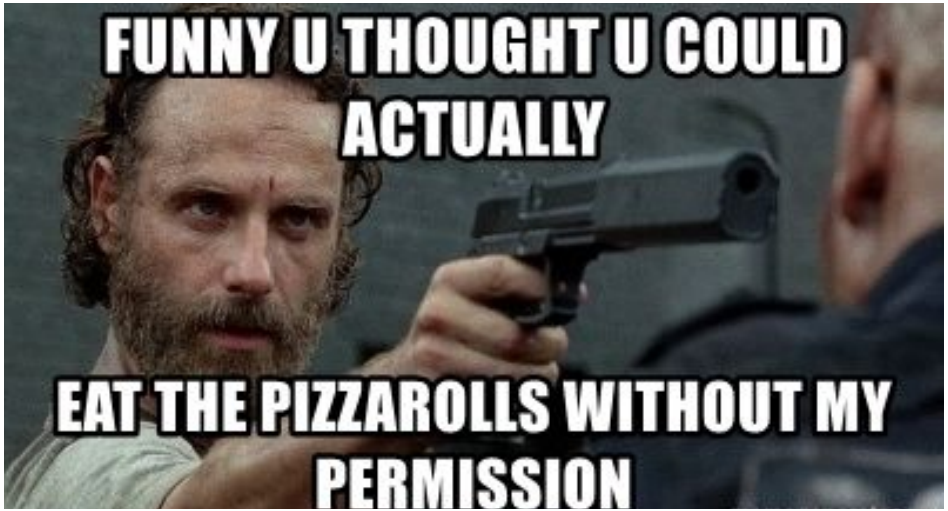
1. Huge amount of potential users
2. Public access with need for authentication
3. Short time user's life cycle



VOICENTER

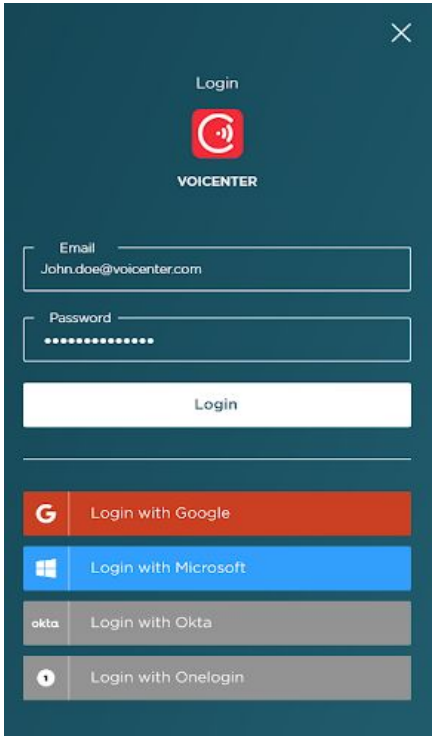
Advanced permission management

1. Call center advance call action
   a. Eavesdropping
   b. Whispering
   c. Conferencing
2. International Calls restriction
3. IP based restriction
4. Conference rooms restriction



S-S-S-O - Sip Single Sign On

1. Save on login time
2. Enjoy Auth provider security policy
3. Save on IT deployment cost
4. SSO became a standard in the business application
5. Reduce human-factor related risks involved credentials handling.



VOICENTER

Third party permission grant

1. Get list of contact for caller name and contact integrations
2. Store a Record on storage provider cloud on clients buckets
3. Create activity in CRM's on the call made with a client in the client activity log

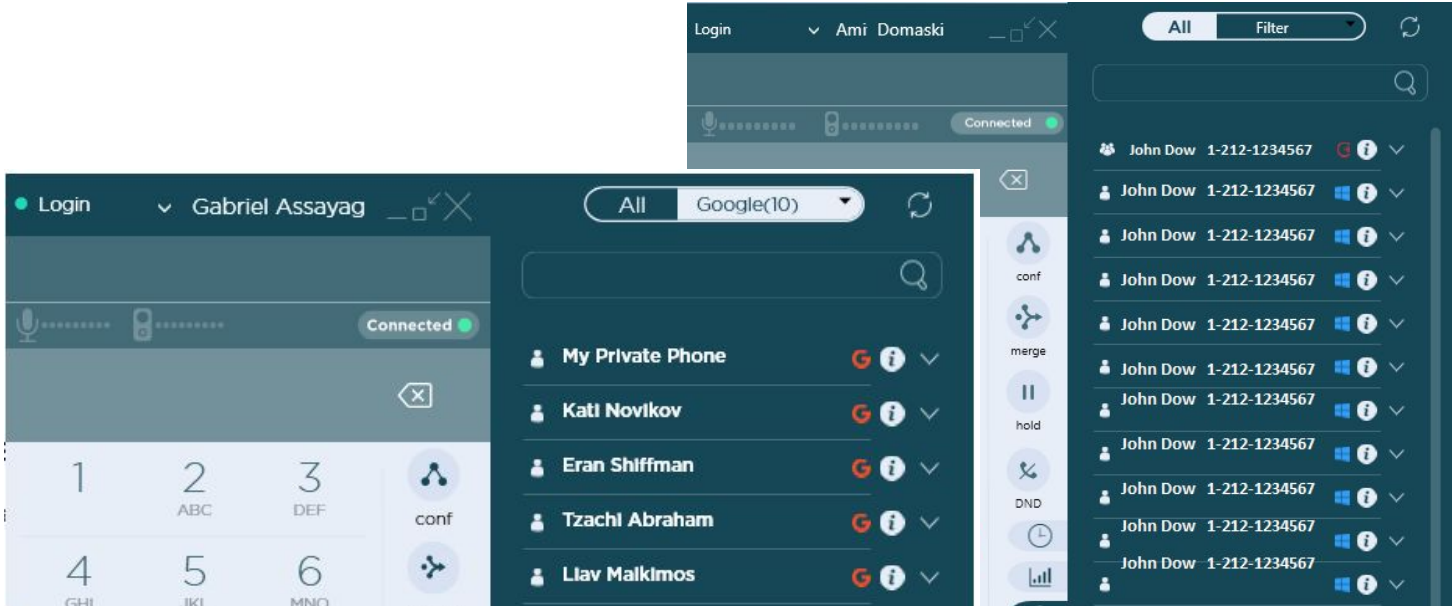
Microsoft Dynamics CRM

ZOHO

salesforce

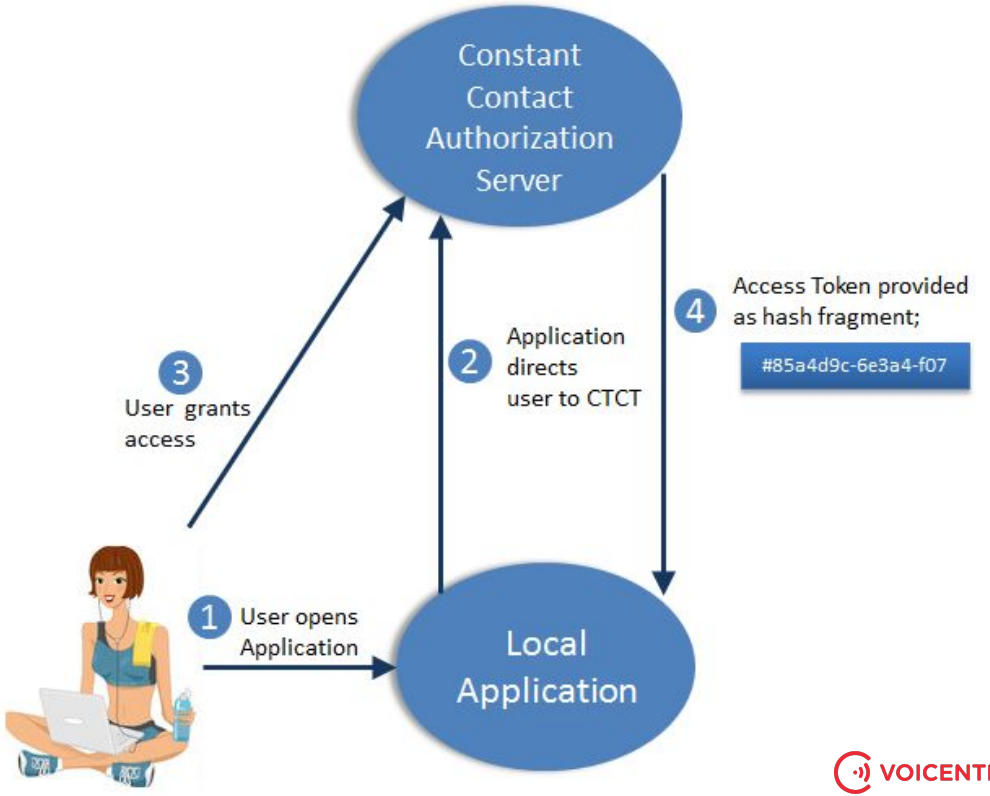
vtiger

And many more...



Authentication Provider Integration - OAuth

OAuth is an open-standard authorization protocol or framework that describes how unrelated servers and services can safely allow authenticated access to their assets without actually sharing the initial, related, single logon credential.



VOICENTER

SIP OAuth Flow

Step 1 – The user shows intent

- **Joe (user):** “Hi Voicenter. I would like to make a call and save its recording”
- **Voicenter (user application):** “Great! Let me go to Google to ask for permission.”

Step 2 – Voicenter (user application) gets permission

- **Voicenter (user application):** “Hello Google , i have a user that would like to make a call and save its recording at your storage. Can I have a request token?”
- **Google Cloud Storage (Service Provider):** “Sure. Here’s a token and a secret.”

The secret is used to prevent request forgery. The application uses the secret to sign each request so that the service provider can verify it is actually coming from the consumer application.

VOICENTER

SIP OAuth Flow

Step 3 – The User Is Redirected to the Service Provider

- **Voicenter:** “OK, Joe. I’m sending you over to Google so you can approve. Take this token with you.”
- **Joe:** “OK!”

<Voicenter directs Joe to Google for authorization>

Step 4 – The User Gives Permission

- **Joe:** “Google, I’d like to authorize this request token that Voicenter gave me.”
- **Google:** “OK, just to be sure, you want to authorize Voicenter **to make a call** and store the call recording file in your Google storage account?”
- **Joe:** “Yes!”
- **Google:** “OK, you can go back to Voicenter callback ,and tell them there that they have permission to use that token ”

Google marks the request token as “good-to-go,” so when the Application requests access, it will be accepted (so long as it’s signed using their shared secret).

VOICENTER

SIP OAuth Flow

Step 5 – The Application obtains an access token

- **Joe:** “Google gave me that token , can i register my phone ?”
- **Google:** “Sure, i can see it was sing by google , you are 200 Super OK ,”

Step 6 – The application accesses the protected resource

- **Voicenter:** “I’d like to to store this call on your storage . Here’s my JWT token!”
- **Google:** “Done!”

OAuth SIP cake recipe

- 1 Login HTML page
- 1 Phone Html page
- 3 js functions (GetConfig, Register ,Call) JSSIP implementation
- 1 Web Server backend OAuth login
- 1+ Auth provider application config
- 2 Auth provider public signature
- 1 shot of espresso

VOICENTER

Login HTML page

/public/Login.js

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
<form action="/connect/google" method="POST" accept-charset="utf-8">
  <fieldset>
    <div>
      <button>Submit</button>
    </div>
  </fieldset>
</form>
</body>
</html>
```

VOICENTER

Phone HTML page

/public/Phone.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <script src="https://jssip.net/download/releases/jssip-3.5.7.js"></script>
  <script src="call.js"></script>
  <title>Title</title>
</head>
<body>
<button onClick="Register()"> Register</button>
<button onClick="Call()"> Call</button>
</body>
</html>
```

VOICENTER

3 JS functions (GetConfig,Register ,Call) JSSIP implementation

/public/call.js

```
function GetConfig() {
        const urlParams = new URLSearchParams(window.location.search);
        if(urlParams.get('token'))token=urlParams.get('token');
        console.log("token",token)
        var configuration = {
          sockets  : [ new window.JsSIP.WebSocketInterface('wss://webrtc.officering.net:8888') ],
          session_timers: false,
          uri      : 'sip:oauth@webrtc.officering.net',
          authorization_jwt :"Bearer "+token
        };
        return configuration
}
function Register() {
        window.ua = new window.JsSIP.UA(GetConfig());
        window.ua.start();
}
function Call() {
        window.ua.activeCall = window.ua.call('sip:6001@webrtc.officering.net',{'mediaConstraints': {'audio': true, 'video': false}});
        window.ua.activeCall.connection.addEventListener('addstream', (e) => {
          console.log("addstream",e);
          var audio = document.createElement('audio');
          audio.srcObject = e.stream;
          audio.play();
        });
}
```

VOICENTER

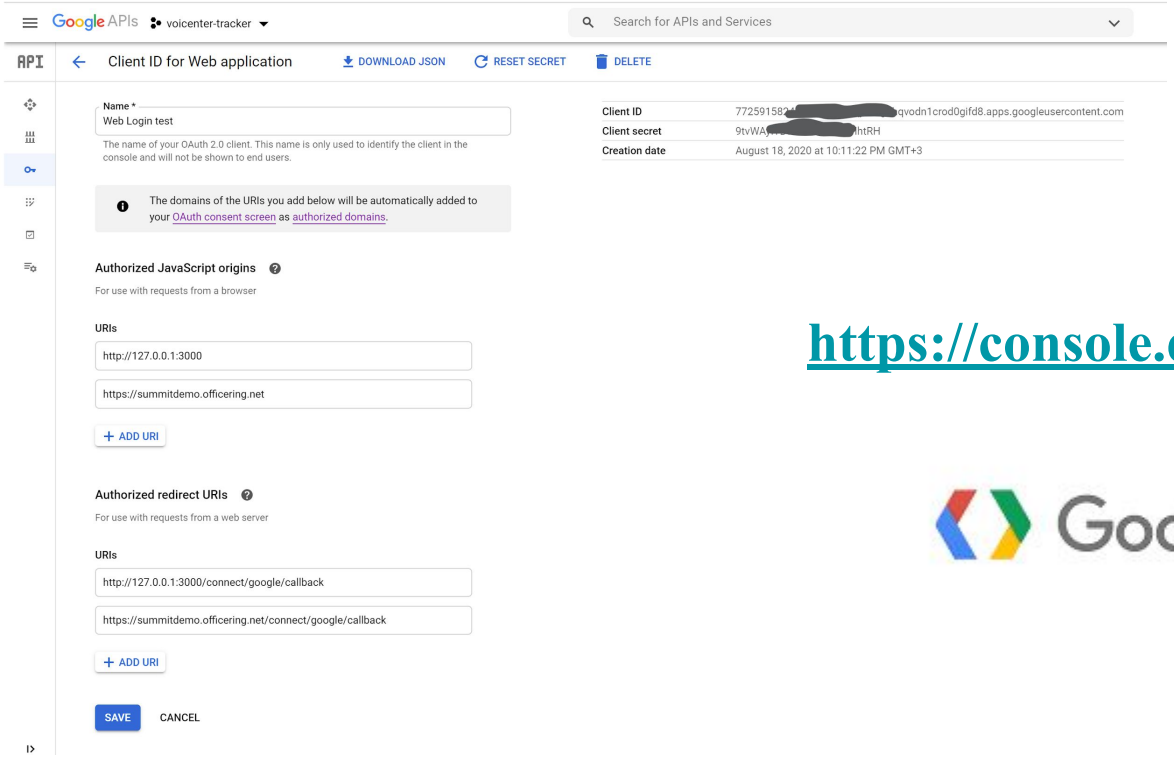
Web Server backend OAuth login

/server.js

```
var fastify = require('fastify')
var cookie = require('fastify-cookie')
var session = require('fastify-session')
var parser = require('fastify-formbody')
var grant = require('grant').fastify()
const path = require('path')
fastify()
  .register(cookie)
  .register(session, {secret: '01234567890123456789012345678912', cookie: {secure: false}})
  .register(parser)
  .register(grant(require('./config')))
  .register(require('fastify-static'), {root: path.join(__dirname, 'public'), prefix: '/'  })
  .route({method: 'GET', path: '/', handler: async (req, res) => {
      if(req.session && req.session.grant && req.session.grant.response){
          res.redirect("/Phone.html?token="+req.session.grant.response.raw.id_token);
      }else{
          res.redirect("/Login.html");
      }
}})
  .listen(3001)
```

VOICENTER

Google Oauth application config



https://console.developers.google.com/

Google Developers

VOICENTER

Google Oauth application config

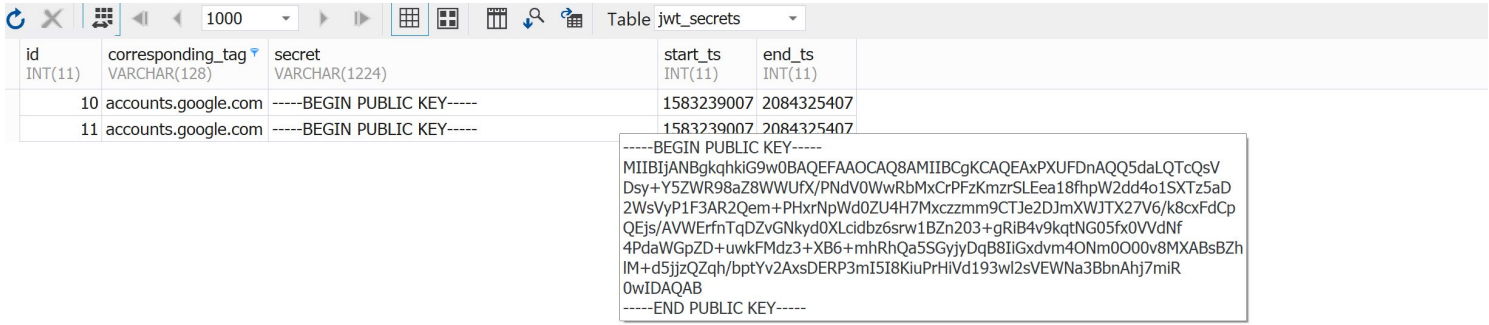
/config.js

```
{
 "defaults": {
   "origin": "http://localhost:3001",
   "transport": "session"
 },
 "google": {
   "key": "772**********************************************d8.apps.googleusercontent.com",
   "secret": "9t*****************htRH",
   "dynamic": ["scope"],
   "scope": ["openid","email","https://www.googleapis.com/auth/admin.directory.user.alias.readonly"],
   "response": ["raw", "jwt"],
   "callback": "/"
 }
}
```

VOICENTER

Google JWT public signature

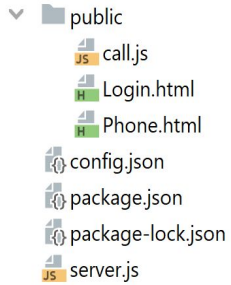

```
{
    3f332b3e9b928bfe51bcf4f8da543cc4bd9d4724: "-----BEGIN CERTIFICATE----- MIIDJjCCAg6gAwIBAgIIAS+xfbAfX+AwDQYJKoZIhvcNAQEFBQAwNjE0MDIGA1UE AxMrZmVkZXJhdGVkLXNpZ25vbi5zeXN0ZW0uZ3NlcnZpY2VhY2NvdW50LmNvbTAe
    Fw0yMDA5MDUwNDI5MzhaFw0yMDA5MjExNjQ0MzhaMDYxNDAyBgNVBAMTK2ZlZGVy YXRlZC1zaWdub24uc3lzdGVtLmdzZXJ2aWNlYWNjb3VudC5jb20wggEiMA0GCSqG SIb3DQEBAQUAA4IBDwAwggEKAoIBAQCj0r0dYYEzCe9IlYZhcS0qC72nsqrHqofm
    3lajNLWQPv2kzLjKIbJRcHAdsizsrGfHCMHPrn9C4u2IVAaGkqR7xqp2P6/ZNBQ9 BQfPRDradYzR/rfokuhwwmFFEFZmwRCFp3nEr5XLyC+40qlvLft1zmDHKMconHkO 35dAfI/mqVdDDziWsY0YmDd3yPd+sCLpwwlIQqvvTF5qgBY2Xg04pP3h96ko67Ep
    1XNkjVGKqDJk8DX3QGQrlRyeOIzOD+h2BhJlaqn2lM0EqPQIewWqXfKdSJKjWNxc DooINbYAw2044u7ODR+rVRj5KZWdT3gMqS0P0MS0wV7A/QXwLo9FAgMBAAGjODA2 MAwGA1UdEwEB/wQCMAAwDgYDVR0PAQH/BAQDAgeAMBYGA1UdJQEB/wQMMAoGCCsG
    AQUFBwMCMA0GCSqGSIb3DQEBBQUAA4IBAQCHurmF/teBGhqR2mT+/XdH/quQ+hDW wbWBRFGQ9JvfENZH2UhWaIV8RRk1VxjeWeIeEwEK+0LeQMepib8l6Pvzp++i6App QTm7d8QDBJJPOi1HiwdaJOcS4frisWoXCTd2iy8SwLC2MNdVP0qE2qTBuhl5l+fK
    MarJcNHWRQiKCXe/MZFwF0I0Be6CoxkK1grBTML1myrKWJh3a8IA0b96kA2yQis2 x7i1O5MhYuKVZQ+RYBdWhNbfZYMZ7eILRZ7PznguE2U3KbXgv1ep7lOmZ0B4ob+i oGB8eUc427Z9mXrehtAy6y9WWoasymt8VPbL7Fj0Gp92/YDzL4jxAoGv -----END CERTIFICATE----- ",
    bc49530e1ff9083dd5eeaa06be2ce437f49c905e: "-----BEGIN CERTIFICATE----- MIIDJjCCAg6gAwIBAgIIeq+tQVfEhGYwDQYJKoZIhvcNAQEFBQAwNjE0MDIGA1UE AxMrZmVkZXJhdGVkLXNpZ25vbi5zeXN0ZW0uZ3NlcnZpY2VhY2NvdW50LmNvbTAe
    Fw0yMDA4MjgwNDI5MzhaFw0yMDA5MTMxNjQ0MzhaMDYxNDAyBgNVBAMTK2ZlZGVy YXRlZC1zaWdub24uc3lzdGVtLmdzZXJ2aWNlYWNjb3VudC5jb20wggEiMA0GCSqG SIb3DQEBAQUAA4IBDwAwggEKAoIBAQDE9dQUOcBBDl1otBNxCxUOzL5jllZH3xpn
    xZZR9f8811XRbBFszEKs8XMqbOtIsR5rXx+GlbZ13ijVJdPPloPZaxXI/UXcBHZB 6b48fGs2lZ3RlTgfszFzPOab0JMl7YMmZdYlNFbtXr+TxzEV0KlASOz8BVYSt+dO oNm8Y2TJ3RctyJ1vPqyvDUFmfbTf6BGIHi/2Sq00bTl/HRVV01/g91pYalkP67CQ
    Ux3Pf5cHr6aFGFBrlIbKPIOoHwiIbF2+bg42bQ7TS/wxcAGwFmGUz53mOPNBmqH9 um1i/YDGwMRE/eYjkjwqK4+seJV3X3fCXaxURY1rcFucCGPuaJHTAgMBAAGjODA2 MAwGA1UdEwEB/wQCMAAwDgYDVR0PAQH/BAQDAgeAMBYGA1UdJQEB/wQMMAoGCCsG
    AQUFBwMCMA0GCSqGSIb3DQEBBQUAA4IBAQBUZXgvGpvy7gF1OMlORTGKHhZ2oR/d 1PctSNCxA0P7tv//ti1d8XyMOxpyoLmvaAGaafvXhVlJp0plPQXJ7FJ9SvDUWhzh c1Lr/otFJbJqWoks58yA5XHGcIQ33VOtNZwS6QxjPekxRv9Ql8Pt+r6FQ/hjH6sW
    9E3qHdngWKEGlHW2Ka8Hl6YKeVXE2hrew7HDFS4W+9BiehVRbNZddbgc2n3AV4S8 vlQ60gAYHOmix+e39Af+h13xrmjiepwzf1YUEylLxyMbv6QolAQ3Y9q67kTLL3fw FFztghwqYw8Sl5h8i1v8zMKORCIG/hruoNplg5/vggcrETEnP6taK4EY -----END CERTIFICATE----- "
}
```

VOICENTER

Instruction checklist

- **Step 1 - Make a new folder and run**
  - i. **> npm init -y**
  - ii. **> npm install -s grant fastify-formbody fastify-session fastify-cookie fastify fastify-static**
- **Step 2 - Create 5 files**
  - i. **public/Login.html**
  - ii. **public/Phone.html**
  - iii. **public/call.js**
  - iv. **/config.js**
  - v. **/server.js**
- **Step 3 - Create Google Oauth application config , and add the key and secret on /config.js**
  https://console.developers.google.com/
  https://developers.google.com/identity/protocols/oauth2
- **Step 4 - Import Google JWT public signature**
- **Step 5 - Run the server.js file :**
  > node server.js

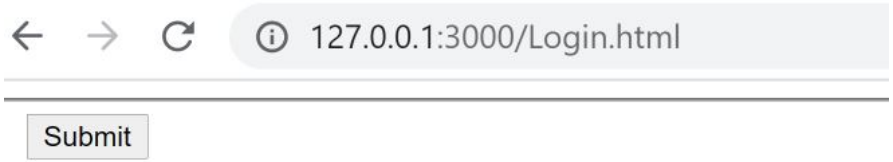


Enjoy your cake over http://127.0.0.1:3000

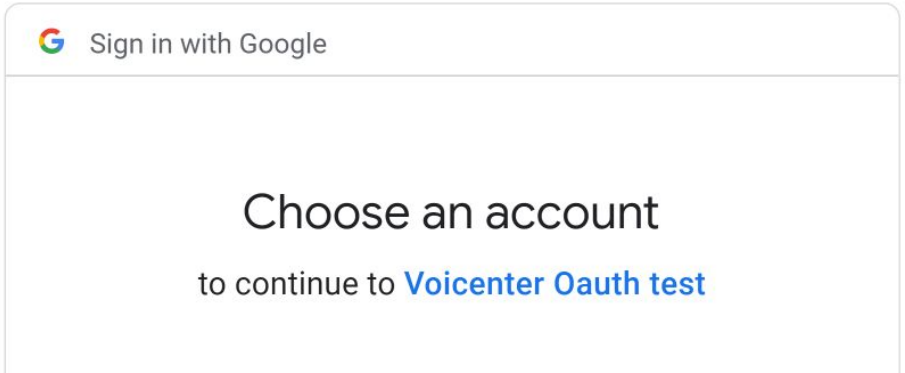
VOICENTER

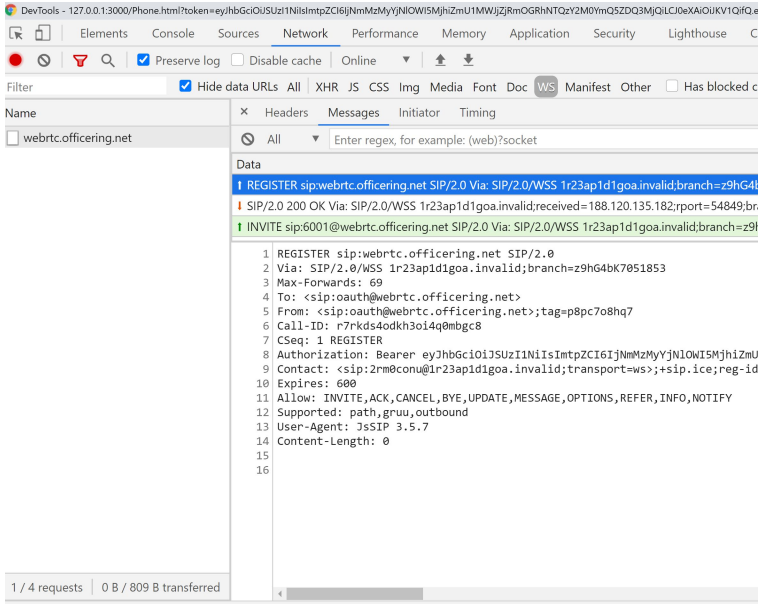
How it will look

1>

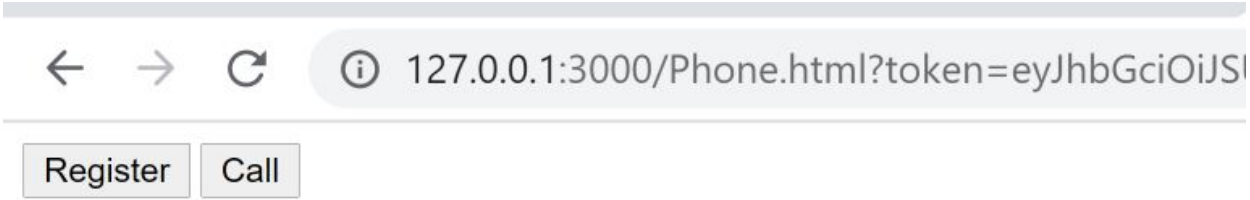


2>





3>



VOICENTER

Invite the media server

```
if ($json(jwt_body) != NULL && $json(jwt_body/email) != NULL) {
        xlog("XXXX - OAuth email $json(jwt_body/email) \n");
        append_hf("X-OAUTH: $json(jwt_body/email) \r\n");
        $avp(email)= $json(jwt_body/email);
        $avp(emailName)=$(avp(email){s.select,0,@});
        $avp(emailUri)="sip:" + $avp(emailName) +"@" + $fd;
          uac_replace_from("$avp(emailName)","sip:$avp(emailName)@$fd");
 }
```

```
                                                  Call flow for ge9keo90e3kq688hmf0e (Color by Request/Response)
                                                  xINVITE sip:6001@webrtc.officering.net SIP/2.0
        192.168.201.231:5060          192.168.201.27:5060xRecord-Route: <sip:192.168.201.231;r2=on;lr;ftag=q9g1n8fkso;did=eb4.b2d4b093>
       qqqqqqqqqqwqqqqqqqqq          qqqqqqqqqqwqqqqqqqqqxRecord-Route: <sip:209.163.136.231:8888;transport=wss;r2=on;lr;ftag=q9g1n8fkso;did=eb4.b2d4b093>
13:13:05.531497   x        INVITE (SDP)         x        xVia: SIP/2.0/UDP 192.168.201.231:5060;branch=z9hG4bKa6b7.7036447.0;cid=202;i=4dd490d7
      +0.001161   x qqqqqqqqqqqqqqqqqqqqqqqqqqq> x        xVia: SIP/2.0/WSS ebepftjppjqo.invalid;rport=63009;received=188.120.135.182;branch=z9hG4bK7493643
13:13:05.532658   x         100 Trying          x        xMax-Forwards: 68
      +1.013622   x <qqqqqqqqqqqqqqqqqqqqqqqqqqq x        xTo: <sip:6001@webrtc.officering.net>
13:13:06.546280   x        200 OK (SDP)         x        xFrom: shlomi <sip:shlomi@webrtc.officering.net>;tag=q9g1n8fkso
      +0.188710   x <qqqqqqqqqqqqqqqqqqqqqqqqqqq x        xCall-ID: ge9keo90e3kq688hmf0e
13:13:06.734990   x            ACK              x        xCSeq: 196 INVITE
                  x qqqqqqqqqqqqqqqqqqqqqqqqqqq> x        xContact: <sip:at7jjlqp@188.120.135.182:63009;transport=ws;ob>
                  x                             x        xContent-Type: application/sdp
                  x                             x        xAllow: INVITE,ACK,CANCEL,BYE,UPDATE,MESSAGE,OPTIONS,REFER,INFO,NOTIFY
                  x                             x        xSupported: ice,replaces,outbound
                  x                             x        xUser-Agent: JsSIP 3.5.7
                  x                             x        xContent-Length: 1035
                  x                             x        xX-OAUTH: shlomi@starkey.co.il
                  x                             x        xX-APP-NAME : Conference
```

VOICENTER

Thank you ....

VOICENTER