

Τεχνολογία Πολυμέσων

Εργασία 2021-2022

Project : WEBRTC conference.

Tools : Server side Javascript (Node.js) , Client side Javascript , html , css.

WEBRTC architecture: MESH.

Available Services : Camera , Microphone , Files , Chat , Users list.

Tested with: Chrome , Brave.

Authors : Μυρτώ Ελευθέρου 3170046 , Χρύσα Οικονόμου 3170127 , Νίκος Δεληγάννης 3170268

Εγκατάσταση

- 1) Ανοίξτε κάποιον editor (VScode) στον νέο φάκελο (WebRTC)
- 2) Εγκαταστήστε το περιβάλλον node.js στον φάκελο και έπειτα εκτελέστε την εντολή :
nodemon server.js .
- 3) Εφόσον ο server τρέχει στην local IPv4 ή στο localhost με port 3000 είστε έτοιμοι να τρέξετε τον client πληκτρολογώντας στον browser : http://<your_ip>:<port> . Άμα θέλετε να τρέξετε τον server πάνω στην τοπική σας IPv4 address (για να συνδεθούν στον server clients με διαφορετικό device -στο ίδιο δικτυο- από αυτόν πχ.) τότε θα πρέπει να απενεργοποιήσετε το secure origin του browser σας για τις παραμέτρους που τρέχει ο server(http://<your_ip>:<port>) . Περισσότερες πληροφορίες για το πώς θα αντιμετωπίσετε το παραπάνω issue: <https://medium.com/@Carmichaelize/enabling-the-microphone-camera-in-chrome-for-local-unsecure-origins-9c90c3149339>

Η πλατφόρμα μπορεί να υποστηρίξει πολλαπλά δωμάτια την ίδια χρονική στιγμή όπου το κάθε δωμάτιο έχει χωρητικότητα έως και 7 άτομα.

Εγκατάσταση “ngrok”

Εφόσον θέλετε ο server να είναι προσβάσιμος από διαφορετικά δίκτυα τότε θα πρέπει να εγκαταστήσετε στο μηχανημα που τρέχει ο server το λογισμικό ngrok : <https://ngrok.com/download> . Το ngrok θα κάνει δημόσιο το address του server δημιουργώντας http / https σύνδεσμο ο οποίος θα είναι ένα public reference της διεύθυνσης του server.

Βήματα:

- 1) Εγκαταστείτε και εκτελέστε το ngrok.exe αφού κάνετε εγγραφή στην σελίδα(είναι απαραίτητη)
- 2) Τρέξτε την εντολή `ngrok authtoken <token>` στο terminal που θα ανοίξει κατά την εκτέλεση του ngrok.exe , όπου token ένα string το οποίο θα αποκτήσετε με το πέρας της εγγραφής σας στο λογισμικό .
- 3) Ξεκινήστε το tunneling για την διεύθυνση του server . Πχ. Έστω ότι ο server μας τρέχει στην ip localhost και στο port 8080. Η εντολή που θα πρέπει να εκτελέσετε στο τερματικό θα είναι αυτή: `ngrok http 8080` . Το παραπάνω παράδειγμα καλύπτει τον server του project καθώς τρέχει σε localhost:8080.
- 4) Με την προϋπόθεση ότι τα παραπάνω βήματα ολοκληρώθηκαν θα πρέπει να αναγράφονται στο τερματικό δύο συνδέσεις μια http και μια https όπως στο παράδειγμα πιο κάτω . Αντιγράψτε τον σύνδεσμο που αναγράφεται(εμείς χρησιμοποιήσαμε τον https για μεγαλύτερη ασφάλεια) και δημοσιεύστε τον στους υπόλοιπους χρήστες. Τώρα κάθε χρήστης(client) που έχει αυτόν τον σύνδεσμο μπορεί να τον τρέξει στον browser του και να έχει πρόσβαση στον απομακρυσμένο server της εφαρμογής .

```
ngrok by @inconshreveable (Ctrl+C to quit)

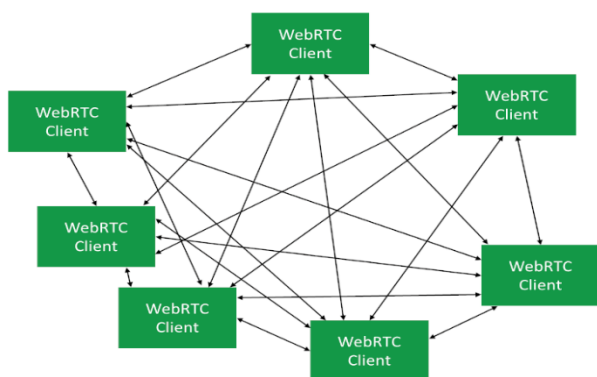
Session Status      online
Account             ndeligiannis8@gmail.com (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://d26f-2a02-587-b802-b55e-c5e0-72b7-8569-e4d3.ngrok.io -> http://localhost:8080
Forwarding           https://d26f-2a02-587-b802-b55e-c5e0-72b7-8569-e4d3.ngrok.io -> http://localhost:8080

Connections         ttl    opn    rt1    rt5    p50    p90
                   197    0      0.00   0.00   0.46   7.58
```

5)

WebRTC Υλοποίηση

Για την υλοποίηση του διαμοιρασμού εικόνας , ήχου και αρχείων προσεγγίσαμε την webRTC αρχιτεκτονική MESH (rtcpeerconnection[]) . Για το chat χρησιμοποιήσαμε τον signaling server μας ο οποίος επικοινωνεί με τον client μέσω του socket.io api , δηλαδή υπεύθυνος για την αποστολή των μηνυμάτων είναι αποκλειστικά ο server.



Αρχιτεκτονική: MESH

Η αρχιτεκτονική MESH αναφέρεται σε ένα WEBRTC σύστημα το οποίο αποτελείται από πολλούς clients όπου κάθε client έχει μια ανοικτή σύνδεση(peerconnection) για κάθε χρήστη που βρίσκεται στο ίδιο δωμάτιο. Στην πλατφόρμα μας αυτό επιτυγχάνεται με την αποθήκευση των peerconnections σε λίστα , με αποτέλεσμα κάθε νέος χρήστης που συνδέεται να δημιουργεί συνδέσεις με όλους τους άλλους χρήστες , ενώ οι ήδη

συνδεδεμένοι χρήστες ενημερώνονται από τον signaling server μας ότι ένας νέος χρήστης συνδέθηκε και προσθέτουν στην υπάρχουσα λίστα τους τη νέα σύνδεση με το id του νέου χρήστη. Επίσης αξιο αναφοράς είναι πως ο signaling server αποθηκεύει σε δομή τα socket ids σε συνάρτηση με τα usernames των clients και τα στέλνει στους clients ενός συγκεκριμένου δωματίου όταν ένας νέος χρήστης συνδέεται στο δωμάτιο ή όταν χρήστης αποσυνδέεται από το δωμάτιο. Με αυτόν τον τρόπο ενημερώνεται η λίστα των peerconnections του κάθε webRtc Client. Κατα τα άλλα τα peers συνδέονται με τον κλασικό τρόπο (offer – answer). Για την υλοποίηση των συνδέσεων ανάμεσα στα peers βασιστήκαμε στη παρακάτω πηγή: <https://webRTC.ventures/2018/07/tutorial-build-video-conference-application-webrtc-2/> .

Streams

Όπως αναφέραμε η πλατφόρμα μας εξυπηρετεί διαμοιρασμό εικόνας , ήχου και αρχείων μέσω του WebRTC api . Η εικόνα και ο ήχος στέλνονται ανάμεσα στους χρήστες όταν ένας νέος client συνδεθεί και σετάρει το localstream του με την βοήθεια της συνάρτησης addstream(localstream) όπου ουσιαστικά γίνεται triggered η συνάρτηση onaddstream σε κάθε άλλον client ο οποίος εμφανίζει το localstream του νέου χρήστη. Όσων αφορά τον νέο χρήστη κάνει ακριβώς το ίδιο αφού οι ήδη συνδεδεμένοι clients έχουν σετάρει και αυτοί τα localstream τους και έτσι τα εμφανίζει.

DataChannels

Τα datachannels χρησιμοποιούνται για τον διαμοιρασμό αρχείων ανάμεσα στους χρήστες. Βασίζονται πάνω στις συνδέσεις των peers και δημιουργούνται πριν σταλθεί το offer από το offerer peer.
