

 eBook Gratuit

APPRENEZ

webrtc

eBook gratuit non affilié créé à partir des
contributeurs de Stack Overflow.

#webrtc

Table des matières

À propos	1
Chapitre 1: Démarrer avec webrtc	2
Remarques.....	2
Exemples.....	2
Configuration d'un système de communication basé sur WebRTC.....	2
Introduction au WebRTC.....	3
Accédez à vos fichiers audio et vidéo en utilisant l'API getUserMedia (), Bonjour WebRTC!.....	3
Chapitre 2: Utilisation de getUserMedia () pour demander l'accès à la caméra et au microph	5
Exemples.....	5
Utiliser getUserMedia ().	5
Pour quoi getUserMedia() est-il utilisé?.....	5
Navigateurs pris en charge pour getUserMedia().....	5
Fichiers requis pour utiliser getUserMedia.....	5
Chapitre 3: WebRTC exemples simples	7
Paramètres.....	7
Exemples.....	7
Obtenir la permission de la caméra et du microphone et prévisualiser l'affichage sur la pa.....	7
Crédits	9

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [webrtc](#)

It is an unofficial and free webrtc ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official webrtc.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec webrtc

Remarques

WebRTC est un projet ouvert et gratuit qui fournit aux navigateurs et aux applications mobiles des fonctionnalités de communication en temps réel (RTC) via des API simples. Les composants WebRTC ont été optimisés pour mieux servir cet objectif.

Commander les liens suivants pour obtenir plus d'informations sur WebRTC

webrtc.org

[architecture webrtc](#)

[vérifier la démo en direct ici](#)

Exemples

Configuration d'un système de communication basé sur WebRTC

Pour configurer un système de communication WebRTC, vous devez disposer de trois composants principaux:

1. Un serveur de signalisation WebRTC

Pour établir une connexion WebRTC, les homologues doivent contacter un serveur de signalisation, qui fournit ensuite les informations d'adresse dont les homologues ont besoin pour configurer une connexion entre homologues. Les serveurs de signalisation sont par exemple:

- [signalmaster](#) : serveur de signalisation léger basé sur JavaScript
- [NextRTC](#) : serveur de signalisation basé sur Java
- [Kurento](#) : framework WebRTC complet
- [Janus](#) : Passerelle WebRTC à usage général

2. Une application client WebRTC

Le client accède à l'implémentation WebRTC d'un navigateur via une API JavaScript ou utilise une bibliothèque WebRTC (par exemple, dans le cadre d'une application de bureau ou d'une application mobile). Pour établir la connexion avec un homologue, le client doit d'abord se connecter au serveur de signalisation. Exemples de clients WebRTC:

- [Plusieurs projets Kurento](#)
- [OpenWebRTC](#) , un client multi-plateforme avec focus mobile
- [Peer.js](#) Un client basé sur un navigateur (Peer.js fournit également un serveur léger)
- [Exemples de démos Janus](#)

3. Un serveur STUN / TURN

[Session Traversal Utilities pour NAT \(STUN\)](#) permet aux homologues d'échanger des informations d'adresse même s'ils sont derrière des routeurs utilisant la [traduction d'adresses réseau \(NAT\)](#) . Si des restrictions de réseau empêchent les homologues de communiquer directement, le trafic est acheminé via un serveur [Traversal Using Relays Around NAT \(TURN\)](#) . Vous trouverez une explication graphique et détaillée de STUN et TURN à l' [adresse http://www.avaya.com/blogs/archives/2014/08/understanding-webrtc-media-connections-ice-stun-and-turn.html](http://www.avaya.com/blogs/archives/2014/08/understanding-webrtc-media-connections-ice-stun-and-turn.html) . Les exemples de serveurs WebRTC STUN / TURN sont les suivants:

- [Coturn](#) combine STUN et TURN et fait généralement partie d'une infrastructure WebRTC à part entière.
- [La passerelle WebRTC de Janus](#) est fournie avec un serveur STUN / TURN intégré.

Introduction au WebRTC

WebRTC est un framework ouvert pour le Web qui active les communications en temps réel dans le navigateur. Il comprend les éléments fondamentaux pour des communications de haute qualité sur le Web, tels que les composants réseau, audio et vidéo utilisés dans les applications de conversation vocale et vidéo.

Ces composants, lorsqu'ils sont implémentés dans un navigateur, sont accessibles via une API JavaScript, permettant aux développeurs d'implémenter facilement leur propre application Web RTC.

L'effort WebRTC est en cours de normalisation au niveau de l'API au niveau du [W3C](#) et au niveau du protocole à l' [IETF](#) .

- Un facteur clé du succès du Web est que ses technologies de base, telles que HTML, HTTP et TCP / IP, sont ouvertes et librement implémentables. Actuellement, il n'existe pas de solution complète, gratuite et de haute qualité permettant la communication dans le navigateur. WebRTC permet cela.
- Déjà intégré aux meilleurs moteurs vocaux et vidéo déployés sur des millions de terminaux au cours des 8 dernières années. Google ne facture pas de redevances pour WebRTC.
- Inclut et résume la technologie de traversée de NAT et de pare-feu, en utilisant STUN, ICE, TURN, RTP-over-TCP et la prise en charge de proxy.
- S'appuie sur la force du navigateur Web: WebRTC extrait la signalisation en offrant une machine à états de signalisation qui mappe directement à PeerConnection. Les développeurs Web peuvent donc choisir le protocole choisi pour leur scénario d'utilisation (par exemple, mais sans s'y limiter, SIP, XMPP / Jingle, etc.).

En savoir plus sur WebRTC d' [ici](#)

**Accédez à vos fichiers audio et vidéo en utilisant l'API `getUserMedia()`,
Bonjour WebRTC!**

`navigator.mediaDevices` est la méthode courante adaptée dans Chrome et FF pour `getUserMedia`

dès maintenant.

Un rappel basé sur les promesses qui renvoie le flux local en cas de succès

```
navigator.mediaDevices.getUserMedia({ audio: true, video: true })
  .then(stream => {
    // attach this stream to window object so you can reuse it later
    window.localStream = stream;
    // Your code to use the stream
  })
  .catch((err) =>{
    console.log(err);
  });
```

Vous pouvez transmettre des **contraintes** audio et vidéo à `getUserMedia` pour contrôler les paramètres de capture tels que la résolution, la fréquence d'images, les préférences de périphérique, etc.

Joindre le flux à un élément vidéo

```
// Set the video element to autoplay to ensure the video appears live
videoElement.autoplay = true;
// Mute the local video so the audio output doesn't feedback into the input
videoElement.muted = true;
// attach the stream to the video element
```

arrêter à la fois vidéo et audio

```
localStream.getTracks().forEach((track) => {
  track.stop();
});
```

arrêter uniquement l'audio

```
localStream.getAudioTracks()[0].stop();
```

arrête seulement la vidéo

```
localStream.getVideoTracks()[0].stop();
```

[Démonstration en direct](#)

Lire Démarrer avec webrtc en ligne: <https://riptutorial.com/fr/webrtc/topic/4623/demarrer-avec-webrtc>

Chapitre 2: Utilisation de `getUserMedia ()` pour demander l'accès à la caméra et au microphone

Exemples

Utiliser `getUserMedia ()`

Comme nous le savons, WebRTC est basé sur le développement et le codage JavaScript et pour plus d'informations et d'exemples, veuillez vous référer [ici](#) et [ici](#) .

Et maintenant, laissez-moi vous montrer un exemple très simple d'utiliser `getUserMedia ()` ;

Pour quoi `getUserMedia ()` est-il utilisé?

`getUserMedia ()` permet de détecter la caméra et le microphone de l'utilisateur / visiteur.

Navigateurs pris en charge pour `getUserMedia ()`

- Mozilla Firefox 22 (PC) ou supérieur.
 - Microsoft Edge 21 (PC) ou supérieur.
 - Google Chrome 23 (PC) ou supérieur.
 - Opera 18 (PC) ou supérieur.
 - Google Chrome 28 (Android) ou supérieur.
 - Mozilla Firefox 24 (Android) ou supérieur.
 - Opera Mobile 12 (Android) ou supérieur.
 - iOS (**Bowser**).
 - Chrome OS
 - Firefox OS
 - Navigateur BlackBerry 10 par défaut.
-

Fichiers requis pour utiliser `getUserMedia`

- <https://github.com/webrtc/samples/blob/gh-pages/src/content/getusermedia/gum/js/test.js>
 - <https://github.com/webrtc/samples/blob/gh-pages/src/content/getusermedia/gum/js/main.js>
 - <https://github.com/webrtc/samples/blob/gh-pages/src/js/adapter.js>
 - <https://github.com/webrtc/samples/blob/gh-pages/src/js/common.js>
 - <https://github.com/webrtc/samples/blob/gh-pages/src/js/lib/ga.js>
-

1. Commencez à coder dans un fichier HTML normal.
2. Dans les balises `<body></body>` , incluez les fichiers API WebRTC requis:

Je ne peux pas utiliser le code de l'éditeur Stackoverflow, voici le code:

<http://pastebin.com/2fQzJhuG>

Bon travail! vous êtes juste un bon débutant maintenant et cela devrait fonctionner normalement.

Dernier code:

```
<html>
<body>
<script src="js/adapter.js"></script>
<script src="js/common.js"></script>
<script src="js/main.js"></script>
<script src="js/lib/ga.js"></script>
<body>
</html>
```

C'était très facile, non?

Lire Utilisation de `getUserMedia ()` pour demander l'accès à la caméra et au microphone en ligne: <https://riptutorial.com/fr/webrtc/topic/6134/utilisation-de-getusermedia----pour-demander-l-acces-a-la-camera-et-au-microphone>

Chapitre 3: WebRTC exemples simples

Paramètres

Paramètres getUserMedia ()	La description
Contraintes	Il consiste en un tableau qui nous permet de spécifier les périphériques multimédia à utiliser, par exemple audio ou vidéo ou les deux.
Rappel de succès	Créer une fonction de rappel de succès qui vous fournira le flux que vous obtenez de vos périphériques multimédia
Erreur de rappel	Ce rappel est appelé quand il y a un problème, comme il n'y a pas de périphériques multimédias, ou l'utilisateur a refusé la permission de les utiliser

Exemples

Obtenir la permission de la caméra et du microphone et prévisualiser l'affichage sur la page Web

Pour commencer à utiliser WebRTC, vous devez obtenir la permission de la caméra et du microphone. Pour ce qui suit:

1. `adapter.js`, vous pouvez l'obtenir d' [ici](#)
2. Une page Web HTML avec une balise vidéo et un peu de code js

L'adaptateur.js est un outil JavaScript pour WebRTC, géré par Google avec l'aide de la [communauté WebRTC](#), qui résume les préfixes des fournisseurs, les différences entre les navigateurs et les modifications apportées aux spécifications.

Maintenant, une fois que vous avez ce fichier, créez un fichier HTML avec le code suivant:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My first webrtc example</title>
    <script src="adapter.js"></script>
    <script type="text/javascript">
      function gotStream(stream) {
        window.AudioContext = window.AudioContext || window.webkitAudioContext;
        var audioContext = new AudioContext();

        // Create an AudioNode from the stream
        var mediaStreamSource = audioContext.createMediaStreamSource(stream);
```

```
        // Connect it to destination to hear yourself
        // or any other node for processing!
        mediaStreamSource.connect(audioContext.destination);
        var video = document.querySelector('video');
        var videoTracks = stream.getVideoTracks();
        window.stream = stream; // make variable available to browser console
        video.srcObject = stream;
    }
    function onfail(error) {
        console.log("permission not granted or system don't have media
devices."+error.name);
    }
    navigator.getUserMedia({audio:true,video:true}, gotStream,onfail);

</script>
</head>
<body>
    Welcome to webrtc
    <video id="local" autoplay=""></video>
</body>
</html>
```

Une fois terminé, enregistrez ce fichier et lancez-le dans le navigateur. Lorsque vous lancez le navigateur, vous êtes invité à autoriser cette page Web à accéder à votre webcam et à votre microphone. Autorisez-le et whola !, vous verrez l'aperçu sur la page Web.

Lire WebRTC exemples simples en ligne: <https://riptutorial.com/fr/webrtc/topic/5641/webrtc-exemples-simples>

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec webrtc	AJ. , Community , Griffin , Ichigo Kurosaki , mpromonet , Olga Khylkouskaya , Sasi Varunan , Timotheus.Kampik , xdumaine
2	Utilisation de getUserMedia () pour demander l'accès à la caméra et au microphone	protld
3	WebRTC exemples simples	Ichigo Kurosaki