Apprendre à construire un assistant virtuel en Node.js avec Watson

Par IBM

Date de publication : 12 juin 2019

Le but de ce tutoriel est de vous apprendre, de façon pratique, à construire un assistant virtuel qui répond à un mot déclencheur. Il s'agira de créer une application client dans Node.js qui envoie de l'audio au service « Speech to Text » de Watson lorsqu'un « mot de réveil » est entendu.

Un espace de travail vous est proposé sur le forum. N'hésitez pas à apporter vos avis. **Commentez**

I - Présentation du projet....3
I-A - Objectifs d'apprentissage....3
I-B - Conditions préalables....3
I-C - Temps estimé....3
II - Mise en œuvre du projet....3
II-A - Étape 1 : Créer des services Watson....3
II-B - Étape 2 : Créer votre environnement d'exécution....6
II-C - Étape 3 : Charger le code de l'assistant virtuel....6
II-D - Étape 4 : Exécuter l'application....9
III - Dépannage....10
IV - Conclusion....10
Remerciements Developpez.com....11

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

I - Présentation du projet

Les assistants personnels sont l'un des principaux cas d'utilisation de la technologie de conversion de la parole en texte. Les « mots du réveil » engagent des dispositifs pour traiter ce qu'ils entendent, comme « Hey Google » ou « Alexa », qui les envoient souvent dans le nuage si une connexion a été établie. Vous pouvez utiliser le service Watson « Speech to Text » de la même manière, selon la façon dont vous écrivez votre application client.

Bien que les bibliothèques et les méthodes puissent différer pour votre plate-forme cible ou votre langage de programmation, les étapes de ce tutoriel vous aident à comprendre comment créer une application qui est toujours à l'écoute, mais qui ne fonctionne qu'après le déclenchement d'un « mot de réveil ».

I-A - Objectifs d'apprentissage

Dans ce tutoriel, vous allez apprendre à créer un assistant virtuel dans Node.js qui s'engage avec l'utilisateur seulement après avoir entendu un « mot de réveil ». Il utilise plusieurs services Watson pour gérer le dialogue verbal entre l'utilisateur et l'assistant virtuel.

I-B - Conditions préalables

Pour mener à bien ce projet, il vous faudrait :

- Un compte **IBM Cloud** ;
- Une instance de service **Watson Speech to Text** ;
- Une instance de service de synthèse vocale **Watson Text to Speech** ;
- Une instance de service **Watson Assistant** ;
- Une compréhension de base de **Node.js** et **NPM**.

I-C - Temps estimé

Ce tutoriel devrait prendre environ 30 minutes. Ce temps suppose que vous avez un compte IBM Cloud et un environnement NPM installé sur votre ordinateur local (Mac OS X ou Linux).

Vous devrez créer plusieurs ressources pour compléter ce tutoriel, c'est donc une bonne idée de commencer par créer un nouveau sous-répertoire local pour les placer toutes.

II - Mise en œuvre du projet

II-A - Étape 1 : Créer des services Watson

Dans le tableau de bord **IBM Cloud**, cliquez sur Créer une ressource pour créer les versions « Lite » de chaque service.

1 **Configurer les informations d'identification**

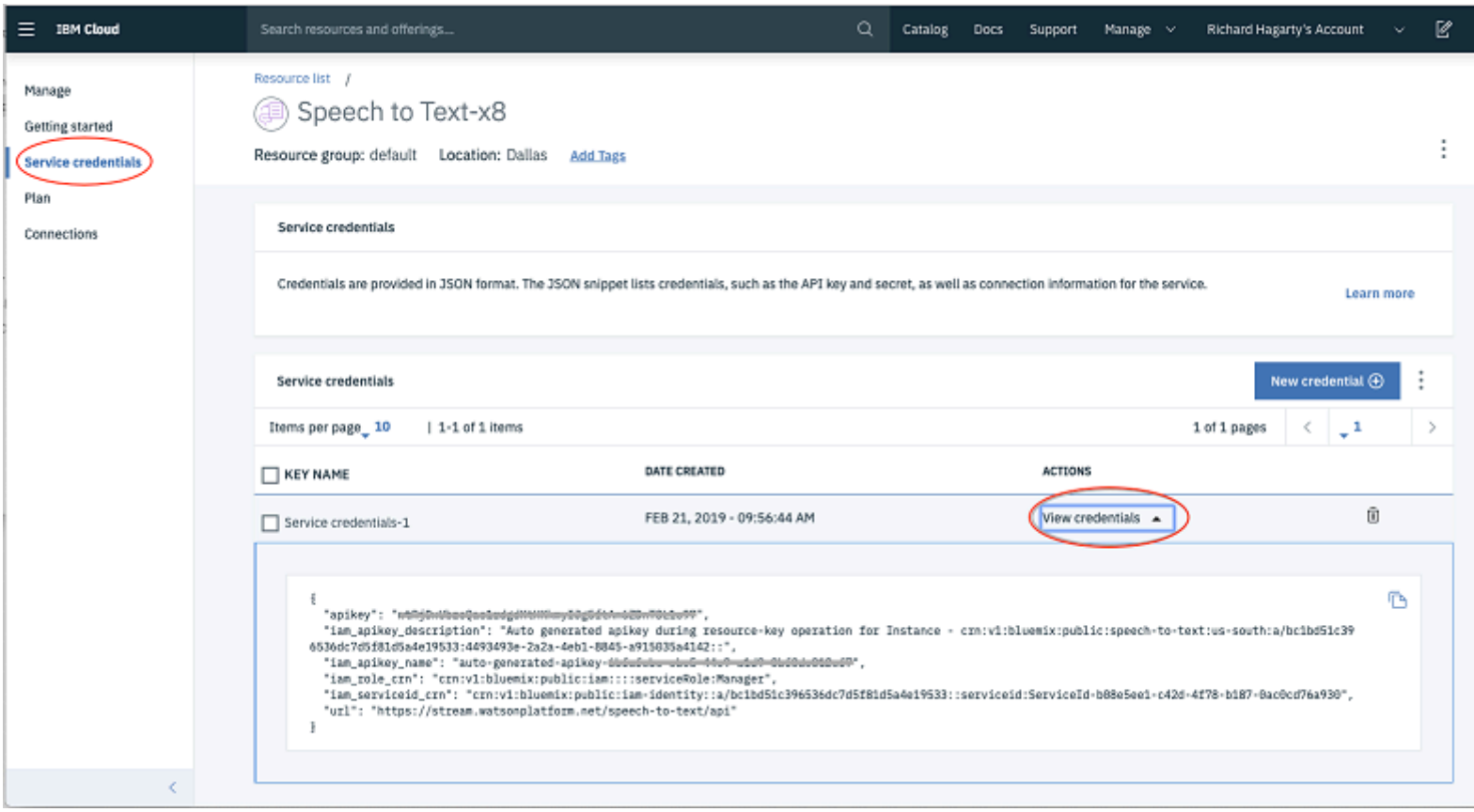
2 Copiez et collez le code suivant dans un fichier local et nommez-le .env.

```
# Watson Assistant
 ASSISTANT_WORKSPACE_ID=<add_assistant_workspace_id>
 ASSISTANT_IAM_APIKEY=<add_assistant_iam_apikey>
 # Watson Speech to Text
 SPEECH_TO_TEXT_URL=<url>
 SPEECH_TO_TEXT_IAM_APIKEY=<add_speech_to_text_iam_apikey>
 # Watson Text to Speech
 TEXT_TO_SPEECH_URL=<url>
```


Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

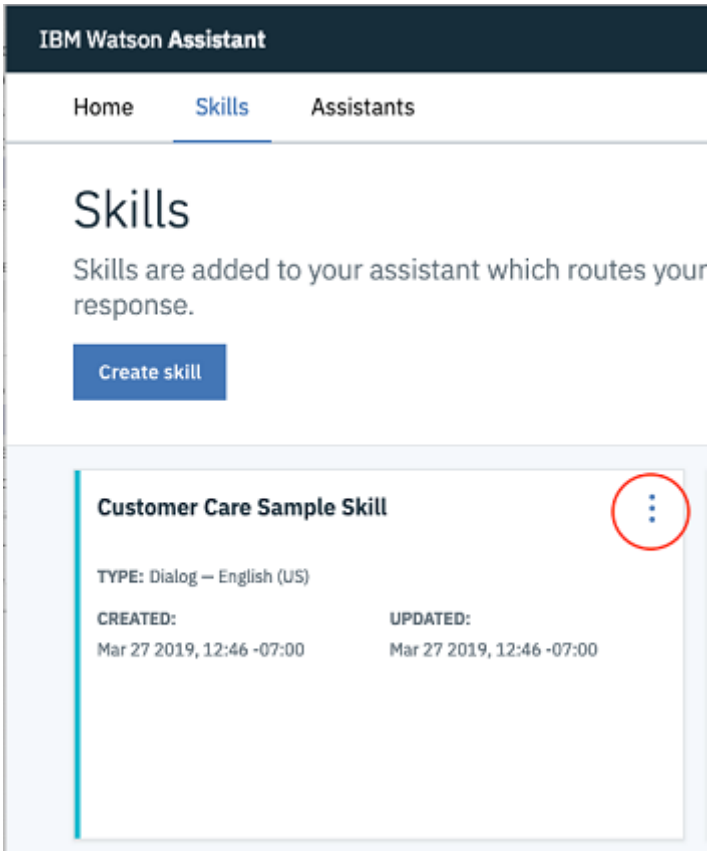
```
TEXT_TO_SPEECH_IAM_APIKEY=<add_text_to_speech_iam_apikey>
```

3 Remplacez les balises <*****> par les valeurs réelles créées pour vos services.
Vous pouvez trouver les informations d'identification en cliquant sur Service credentials, puis affichez les informations d'identification dans la fenêtre du service Watson que vous avez créé, comme le montre l'image suivante.

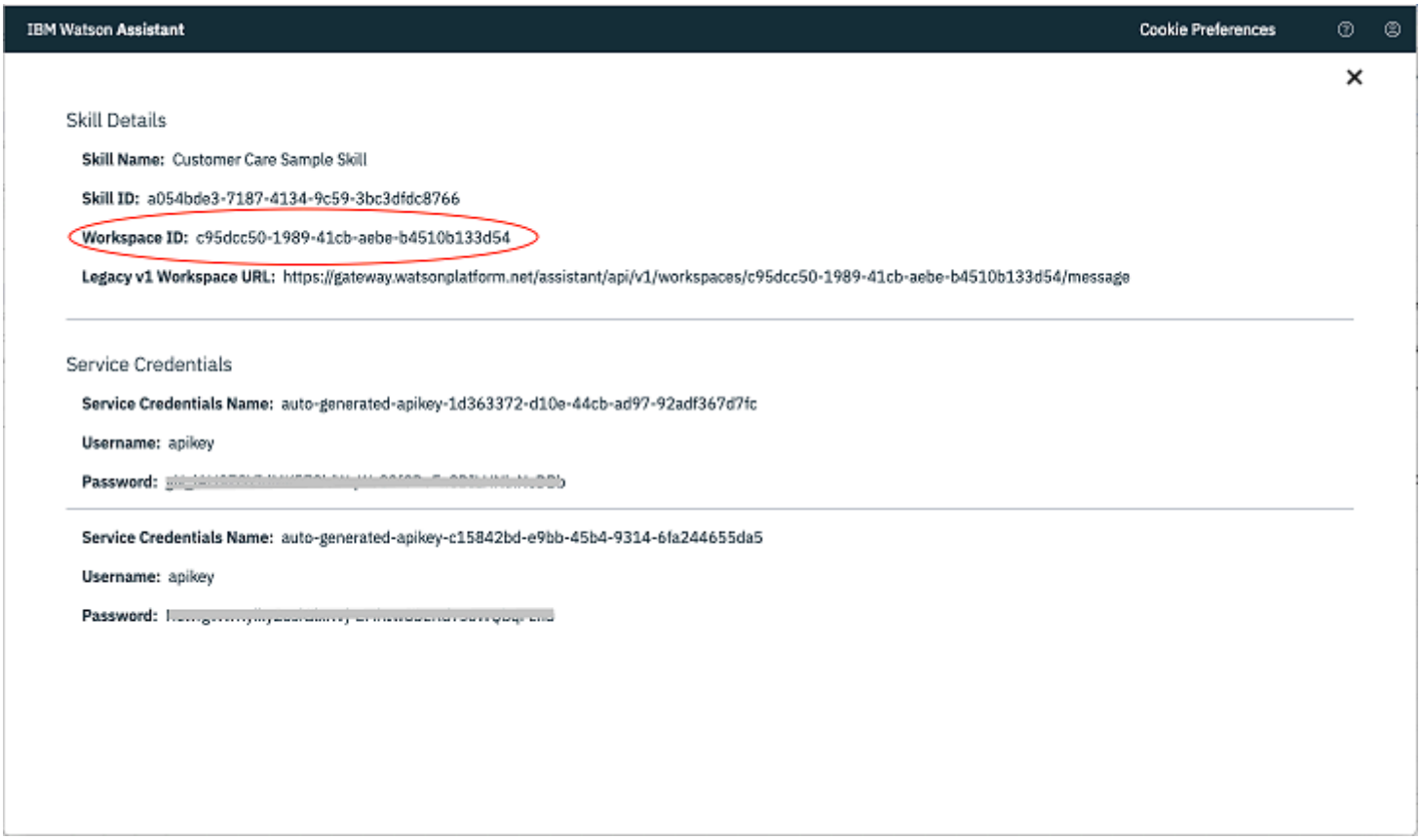


Une valeur WORKSPACE_ID supplémentaire est nécessaire pour accéder au service Watson Assistant. Pour obtenir cette valeur, sélectionnez Gérer, puis lancez l'outil depuis la fenêtre de votre service Watson Assistant. Dans la fenêtre de l'instance de service, sélectionnez Compétences pour afficher les compétences qui existent pour votre service. Pour ce tutoriel, nous utilisons l'exemple de compétence du service à la clientèle qui est fourni avec le service.

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.



4 Cliquez sur le bouton d'option (surligné dans l'image précédente) pour afficher tous les détails de vos compétences et vos références de service.



Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

II-B - Étape 2 : Créer votre environnement d'exécution

Maintenant, vous devez télécharger les paquets NPM pour exécuter votre application. Le code suivant montre un exemple de fichier package.json que vous pouvez placer dans votre répertoire local.

```
{
  "name": "chatbot-with-voice-activation-wake-word",
  "version": "1.0.0",
  "description": "Converse with a virtual assistant.",
  "main": "run.js",
  "scripts": {
    "start": "node run.js"
  },
  "dependencies": {
    "dotenv": "^6.0.0",
    "ibm-watson": "^4.0.1",
    "jsonfile": "^4.0.0",
    "mic": "^2.1.1",
    "node-ffprobe": "^1.2.2",
    "play-sound": "^1.1.1",
    "prompt": "^1.0.0"
  }
}
```

Après avoir créé le fichier, vous pouvez simplement exécuter la commande d'installation pour télécharger les paquets requis.

`npm install`

1 **Installer les dépendances liées à l'audio**
Vous devrez peut-être installer quelques dépendances audio si elles n'existent pas sur votre système.

Sous OS X

Utilisez la commande `brew`pour installer le lecteur `mplayer`, `sox` et `ffmpeg` :

`brew install sox mplayer ffmpeg`

Sur Ubuntu

Utilisez la commande `apt-get` pour installer `ffmpeg`

`sudo apt-get install ffmpeg`

II-C - Étape 3 : Charger le code de l'assistant virtuel

L'extrait de code suivant est une simple application Node.js qui utilise les services Watson que vous venez de créer.

Le code remplit les fonctions primaires suivantes :

- Créer et initialiser les instances des services Watson ;
- Créer et configurer l'objet microphone ;
- Convertir l'audio du microphone en texte ;
- Convertir le texte en audio qui est ensuite lu par le haut-parleur ;
- Mener un dialogue en répondant aux questions de l'utilisateur ;
- Gardez une minuterie pour déterminer si l'assistant virtuel est éveillé ou endormi.

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

Copiez et collez le code suivant dans un fichier local et nommez-le run.js.

```
require('dotenv').config({ silent: true });
const AssistantV1 = require('ibm-watson/assistant/v1');
const TextToSpeechV1 = require('ibm-watson/text-to-speech/v1');
const SpeechToTextV1 = require('ibm-watson/speech-to-text/v1');
const fs = require('fs');
const mic = require('mic');
const speaker = require('play-sound')(opts = {});
const ffprobe = require('node-ffprobe');
var context = {};
var debug = false;
var botIsActive = false;
var startTime = new Date();
const wakeWord = "hey watson";      // if asleep, phrases that will wake us up
const SLEEP_TIME = 10 * 1000;       // number of secs to wait before falling asleep
/**
 * Configuration and setup
 */
/* Create Watson Services. */
const conversation = new AssistantV1({
  version: '2019-02-28'
});
const speechToText = new SpeechToTextV1({
});
const textToSpeech = new TextToSpeechV1({
});
/* Create and configure the microphone */
const micParams = {
  rate: 44100,
  channels: 2,
  debug: false,
  exitOnSilence: 6
};
const microphone = mic(micParams);
const micInputStream = microphone.getAudioStream();
let pauseDuration = 0;
micInputStream.on('pauseComplete', ()=> {
  console.log('Microphone paused for', pauseDuration, 'seconds.');
  // Stop listening when Watson is talking.
  setTimeout(function() {
    microphone.resume();
      console.log('Microphone resumed.');
  }, Math.round(pauseDuration * 1000));
});
/**
 * Functions and main app
 */
/* Convert speech to text. */
const textStream = micInputStream.pipe(
  speechToText.recognizeUsingWebSocket({
    content_type: 'audio/l16; rate=44100; channels=2',
    interim_results: true,
    inactivity_timeout: -1
  })).setEncoding('utf8');
/* Convert text to speech. */
const speakResponse = (text) => {
  var params = {
    text: text,
    accept: 'audio/wav',
    voice: 'en-US_AllisonVoice'
    // voices to choose from:
    // en-US_AllisonVoice
    // en-US_LisaVoice
    // en-US_MichaelVoice
  };
  var writeStream = fs.createWriteStream('output.wav');
  textToSpeech.synthesize(params)
  .then(audio => {
    // write the audio version of the text to the wav file
    audio.pipe(writeStream);
```


Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

```
  })
  .catch(err => {
    console.log('error:', err);
  });
  writeStream.on('finish', function() {
    // determine length of response to user
    ffprobe('output.wav', function(err, probeData) {
      if (probeData) {
        pauseDuration = probeData.format.duration;
        // pause microphone until response is delivered to user
        microphone.pause();
        // play message to user
        speaker.play('output.wav');
        // restart timer
        startTime = new Date();
      }
    });
  });
  writeStream.on('error', function(err) {
    console.log('Text-to-speech streaming error: ' + err);
  });
};
/* Log Watson Assistant context values, so we can follow along with its logic. */
function printContext(header) {
  if (debug) {
    console.log(header);
    if (context.system) {
      if (context.system.dialog_stack) {
        const util = require('util');
        console.log("     dialog_stack: ['" +
                    util.inspect(context.system.dialog_stack, false, null) + "']");
      }
    }
  }
}
/* Log significant responses from Watson to the console. */
function watsonSays(response) {
  if (typeof(response) !== 'undefined') {
    console.log('Watson says:', response);
  }
}
/* Determine if we are ready to talk, or need a wake up command */
function isActive(text) {
  var elapsedTime = new Date() - startTime;
  if (elapsedTime > SLEEP_TIME) {
    // go to sleep
    startTime = new Date();
    botIsActive = false;
  }
  if (botIsActive) {
    // in active conversation, so stay awake
    startTime = new Date();
    return true;
  } else {
    // we are asleep - did we get a wake up call?
    if (text.toLowerCase().indexOf(wakeWord) > -1) {
      // time to wake up
      console.log("App just woke up");
      botIsActive = true;
    } else {
      // false alarm, go back to sleep
      console.log("App needs the wake up command");
    }
    return botIsActive;
  }
}
/* Keep conversation with user alive until it breaks */
function performConversation() {
  console.log('App is listening, you may speak now.');
  textStream.on('data', (user_speech_text) => {
    userSpeechText = user_speech_text.toLowerCase();
    console.log('\n\nApp hears: ', user_speech_text);
```


Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

```
    if (isActive(user_speech_text)) {
      conversation.message({
        workspace_id: process.env.CONVERSATION_WORKSPACE_ID,
        input: {'text': user_speech_text},
        context: context
      }, (err, response) => {
        context = response.context;
        watson_response =  response.output.text[0];
        if (watson_response) {
          speakResponse(watson_response);
        }
        watsonSays(watson_response);
      });
    }
  });
}
/* Start the app */
microphone.start();
performConversation();
```

II-D - Étape 4 : Exécuter l'application

Utilisez la commande suivante pour exécuter l'application :

npm start

Quelques remarques importantes concernant l'exécution de l'assistant virtuel :

- L'application démarre en mode veille et ne répond que lorsqu'elle entend « Hey Watson » ;
- L'application revient en mode veille si elle n'entend rien pendant 10 secondes, bien que vous puissiez modifier cette valeur dans l'application ;
- Une fois engagée, l'application répond conformément à la boîte de dialogue Compétence de l'échantillon de service client définie dans le service Watson Assistant. Vous pouvez explorer le dialogue à partir de l'outil de service pour obtenir de l'aide afin de déterminer les phrases auxquelles il va répondre ;
- Chaque fois que l'assistant virtuel parle, il met le microphone en pause pendant la phrase afin qu'il ne s'entende pas par inadvertance et ne s'embrouille pas ;
- L'audio de l'application est transmis en continu à un fichier local nommé output.wav, qui est ensuite lu par le haut-parleur ;
- L'application génère une sortie console qui devrait vous aider à suivre et déboguer tous les problèmes que vous pourriez rencontrer.

L'image suivante montre un exemple de sortie de l'assistant virtuel.

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

```
rhagarty@Richs-MBP-15 voice-bot (dev3) $ npm start

> chatbot-with-voice@1.0.0 start /Users/rhagarty/journeys/voice-bot
> node run.js

App is listening, you may speak now.


App hears:  hello
App needs the wake up command


App hears:  Hey Watson
App just woke up
Watson says: Hello. Good evening
Microphone paused for 2.142404 seconds.
Microphone resumed.


App hears:  how are you
Watson says: Hi. Good evening
Microphone paused for 2.116961 seconds.
Microphone resumed.


App hears:  can I make an appointment
Watson says: What day would you like to come in?
Microphone paused for 2.066576 seconds.
Microphone resumed.


App hears:  Sunday
App needs the wake up command


App hears:  Hey Watson
App just woke up
Watson says: I didn't understand can you try again
Microphone paused for 2.811701 seconds.
Microphone resumed.


App hears:  Sunday
Watson says: What time on Sunday do you want to come in?
Microphone paused for 3.226395 seconds.
Microphone resumed.
```

Note : Comme vous pouvez le voir, la boîte de dialogue n'est pas optimale. Lorsque l'application se remet en veille, elle devrait effacer le dialogue et repartir à zéro. Ceci peut être corrigé en modifiant la boîte de dialogue d'exemple fournie dans Watson Assistant, mais la personnalisation de la boîte de dialogue n'entre pas dans le cadre de ce tutoriel.

III - Dépannage

Si vous exécutez l'application sous Mac OS X et que le microphone ne semble capter aucun son, assurez-vous que la fonction de base du microphone fonctionne.

```
sox -d test.wav        // speak into mic, then ctrl-c to exit
sox test.wav -d        // playback
```

IV - Conclusion

Ce tutoriel a montré comment vous pouvez utiliser un mot de réveil pour initier un dialogue avec un assistant virtuel qui est construit en utilisant les services IBM Watson.

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.

Si vous voulez une solution similaire qui fonctionne dans un navigateur, regardez « **Capturing Audio in the Browser for Wake Words** ». Vous cherchez d'autres solutions pour le streaming ? Regardez cet **exemple de SDK Node.js**.

Remerciements Developpez.com

Developpez.com remercie IBMpour l'autorisation de publication de ce tutoriel. Tous les remerciements aussi à Guillaume SIGUI pour la mise au gabarit et **Claude Leloup** pour la relecture orthographique.

Les sources présentées sur cette page sont libres de droits et vous pouvez les utiliser à votre convenance. Par contre, la page de présentation constitue une œuvre intellectuelle protégée par les droits d'auteur. Copyright ® 2019 IBM. Aucune reproduction, même partielle, ne peut être faite de ce site et de l'ensemble de son contenu : textes, documents, images, etc. sans l'autorisation expresse de l'auteur. Sinon vous encourez selon la loi jusqu'à trois ans de prison et jusqu'à 300 000 € de dommages et intérêts.