

---

# Manipulations Audio pour ISN: Speech-to-text

Christophe Barès

3 février 2014

## 1 Comment incorporer des fonctions Speech-To-Text dans un programme ?

### 1.1 Solution 1 : Utiliser une bibliothèque

Par exemple : CMU Sphinx <http://cmusphinx.sourceforge.net/>

Bon courage... si vous arrivez à la faire marcher, je suis preneur :-)

### 1.2 Solution 2 : Une usine à gaz autour d'Android

L'OS de google intègre des fonctionnalités Speech-To-Text, peut-on les utiliser dans nos projets ?

Pareil, bon courage... si vous arrivez à la faire marcher, je **ne suis pas** preneur ;-)

### 1.3 Solution 3 : Google est ton ami... littéralement !

L'application [Google Traduction](#) dispose à la fois du **Text-To-Speech** et du **Speech-To-Text**

En tout cas avec les navigateurs compatibles.

Démonstration !

#### 1. Navigateur

Firefox -> KO

Chromium / Chrome -> OK

#### 2. Requetes ?

Chromium/Inspector l'élément -> KO

Wireshark -> KO (en fait pas vraiment... mais cela revient à monter un "Trusted Man-In-The-Middle", qui est un peu trop proche de la véritable attaque "Man-In-The-Middle" à mon goût)

Dernière solution, utiliser les outils dédiés de chrome/chromium :

Se rendre à l'adresse [chrome://chrome-urls](#) -> [chrome://net-internals](#) puis sélectionner "Capture". À partir de là, chrome/chromium se comporte comme un petit Wireshark, mais limité au flux réseau qui passe par le navigateur.

Après avoir stoppé l'acquisition, un petit coup d'oeil sur le "waterfall" pour trouver la transmission qui nous interesse, et on en déduit l'url, la méthode (POST), les entêtes, et le type de fichier audio à envoyer (FLAC).

En utilisant la bibliothèque "requests" de python, le reste est un jeu d'enfant :

```
In [1]: import requests
```

```
In [2]: base = 'https://www.google.com'
url = "/speech-api/v1/recognize?client=chromium&"\
"lang=fr&maxresults=6"
base + url
```

```
Out [2]: 'https://www.google.com/speech-
api/v1/recognize?client=chromium&lang=fr&maxresults=6'
```

```
In [44]: headers = {'content-type': 'audio/x-flac; rate=44100',
                    'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/5
                    'Ubuntu Chromium/32.0.1700.102 Chrome/32.0.170
data= open('../speech/mouton.flac', 'rb').read()
```

```
In [45]: proxies = { "https": "http://172.16.0.1:3128", }
```

```
In [46]: resp=requests.post(base + url, data, headers=headers, proxies=proxies)
```

```
In [47]: resp.request.headers
```

```
Out [47]: CaseInsensitiveDict({'Content-Length': '136470', 'content-type':
'audio/x-flac; rate=44100', 'Accept-Encoding': 'gzip, deflate,
compress', 'Accept': '*/*', 'User-Agent': 'Mozilla/5.0 (X11; Linux
x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu
Chromium/32.0.1700.102 Chrome/32.0.1700.102 Safari/537.36'})
```

```
In [48]: resp.content
```

```
Out [48]: '{"status":0,"id":"","hypotheses":[{"utterance":"dessine-moi un
mouton","confidence":0.885144}]]\n'
```

```
In [18]: import json
speech = json.loads(resp.content)
speech
```

```
Out [18]: {'hypotheses': [{'confidence': 0.885144,
                           u'utterance': u'dessine-moi un mouton'}],
           u'id': u'',
           u'status': 0}
```