REPORT
MUSIC CHATBOT

CHATBOT AND RECOMMENDATION

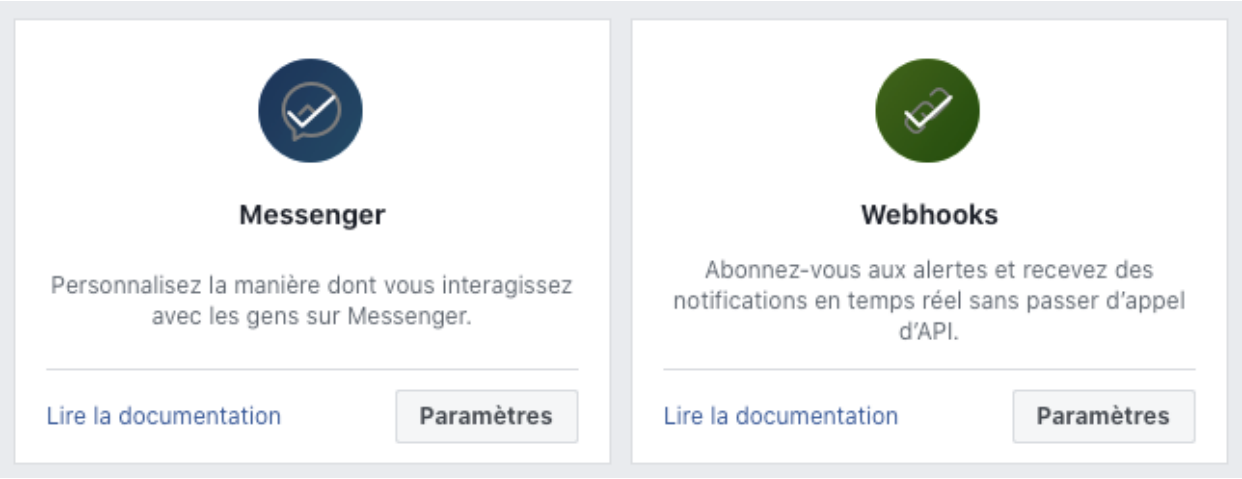
Nader Helali
Ulrich Mama
DIA4

https://github.com/NaderHELALI/Spotify-Chatbot

The main goal of this project is to create a Chatbot connected with a Facebook page. With Messenger, we can talk to the chatbot and he can answer immediately. Our concept was to create a Music Chatbot. In fact, sometimes we want to listen some music from artist we like, or we want to try new music playlist to find other music we do not listen as usual. Nowadays, most part of people who listen to music chose the streaming platform like Spotify, Deezer, Apple Music…
So, we make the choice to use platform music for our Chatbot to have a direct access to music's data, and we chose Spotify because it is the most used music platform.
Spotify suggest an API to have access to their Database (music and playlist).

Connection with Facebook Messenger

We have registered on Facebook for developer, created a Facebook page and chose the parameters for messenger access, to generate a Token some rights to use messenger for our application.



On messenger's parameters we can generate our security token to make a connection with our API

In our Node.js project, we need to create a way to make the connection work between our page's Messenger and our API.
We create a file development.json in a directory config. This json file is only made for store our security data like token and Id.

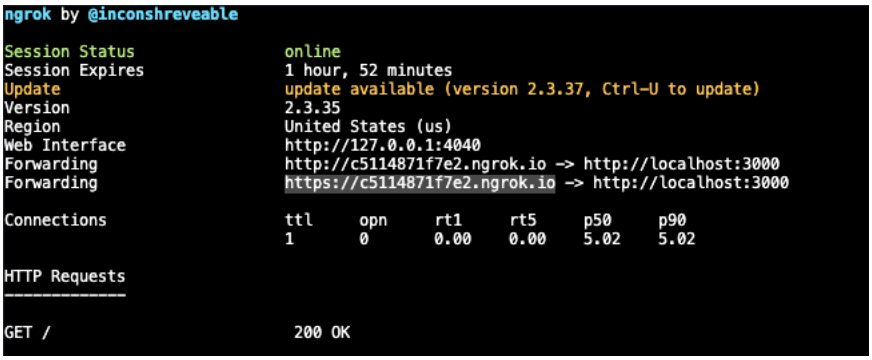
```
{
    "FB": {
        "pageAccessToken":"[redacted]
        "verifyToken": [redacted]
```

We create a class FBeamer to execute each function we need for understand Messenger's data, like recover the message or send a response to the user, make a connection by verified the token etc…

```
class FBeamer {
  constructor({ pageAccessToken, verifyToken }) {
    try {
      if (pageAccessToken && verifyToken) {
        this.pageAccessToken = pageAccessToken;
        this.verifyToken = verifyToken;

      }
      else {
        throw "One or more tokens/credentials are missing!";
      }       Find related code in Spotify-Chatbot
    }
    catch (error) {
      console.log(error);
    }
  }
```

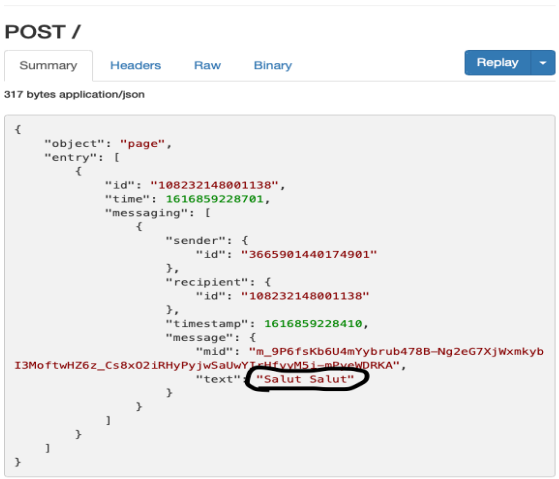
Then, we need to make the connection between our Node.js application and Messenger. To do it, we have to create a Webhooks

For the Url we have chosen the module Ngrok with Node.js, we select the same port that we chose for our app, this server will create the connection between them



When we have executed all the steps, we can verify if the Chatbot is working by using the Web Interface proposed by Ngrok http://127.0.0.1:4040.

We send a simple message on messenger and we check on the web Interface



Now we can see that our connection is working.

Use Spotify API

To use spotify's API we need to register with a Spotify account here https://developer.spotify.com

Then, Spotify give us a Client ID and Token to use the API, we store the IDs in the development.json file and we install the Spotify's module on Node.js.

```
var SpotifyWebApi = require('spotify-web-api-node');

// credentials are optional
var spotifyApi = new SpotifyWebApi({
  clientId: 'fcecfc72172e4cd267473117a17cbd4d',
  clientSecret: 'a6338157c9bb5ac9c71924cb2940e1a7',
  redirectUri: 'http://www.example.com/callback'
});
```

All documentation and information about Spotify's API are available on this link
https://developer.spotify.com
With Spotify's API, with have access to all artist's album and playlist on the platform, with some function made by Spotify's developer.
Some functions can search the album of an artist, a single. Other function can search a playlist with a key word or an ID.

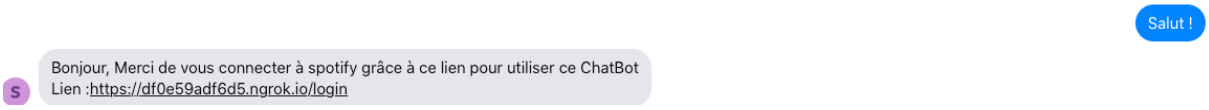
Most part of people, are looking for new music Our chatbot will be able to help them find other music or playlists of all kinds according to their desires. Music is an important part of people's lives. Sometimes we listen to music to take our mind off things, to think about something else, to party, to play sports....
The chatbot is able to suggest playlists with a simple keyword. Using a category or an artist's name, it can suggest a complete playlist or a complete album according to the user's choices
To allow the chatbot to communicate the answers adapted to the user's needs, we designed regex models, with an entity for each model. This allows us to send the right answers to the chatbot user

```
{
    pattern: 'playlist de \\b(?<genre>[a-z]+[ a-z])',
    intent: 'Genre'
}, {
    pattern: 'musique de \\b(?<artist>[a-z]+[ a-z])\\b',
    intent: 'Artist'
}
```

In both models, we turn the keyword into a variable so that it can be used directly in the functions offered by Spotify

```
pattern: '\\b(?<greeting>Hi|Hello|Hey|Salut|Bonjour)\\b',
intent: 'Hello'
```

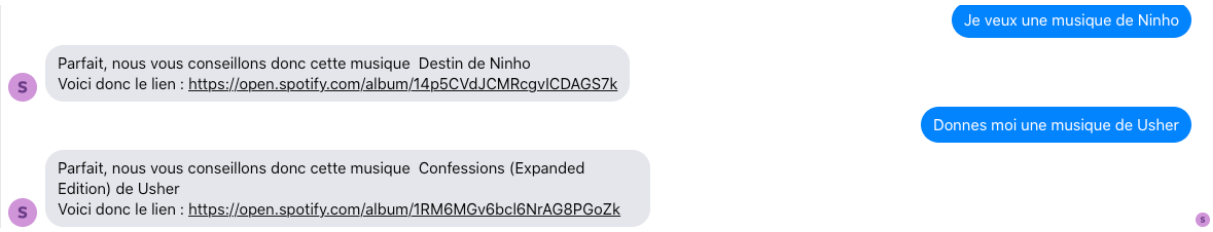
To begin with this chatbot, we need to send him a message like Salut/Bonjour… to start it and connect to our Spotify account



Once we are connected, we can use it.

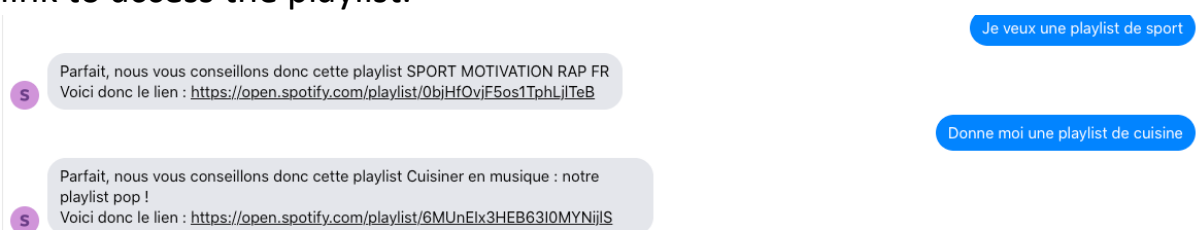
```
case 'Artist':
    {
        if (spotifyApi.getAccessToken() == undefined) {

            FB.sendMessage("RESPONSE", userData.sender, `Bonjour, Merci de vous connecter à spotify grâce à ce lien pour utiliser ce ChatBot\nLien :${link}/login`);
        } else {

            const artist = data.entities.groups.artist
            console.log(artist)
            spotifyApi.searchTracks(`artist:${artist}`)
                .then(function(data) {
                    var num = Math.floor(Math.random() * 20) + 1;
                    console.log('Found playlists are', data.body.tracks.items[num]);
                    const nametrack = data.body.tracks.items[num].album.name
                    const urltrack = data.body.tracks.items[num].album.external_urls.spotify
                    FB.sendMessage("RESPONSE", userData.sender, `Parfait, nous vous conseillons donc cette musique  ${nametrack} de ${artist}\nVoici donc le lien : ${urltrack}`)



                    console.log('Search tracks by "Love" in the artist name', data.body.tracks.items[0]);

                }, function(err) {
                    console.log('Something went wrong!', err);
                });
        }
```

For example, this function helps us to find the music of a particular artist directly and returns a link to the user with the music of the artist they requested.




```
case 'Genre':
    {
        if (spotifyApi.getAccessToken() == undefined) {

            FB.sendMessage("RESPONSE", userData.sender, `Bonjour, Merci de vous connecter à spotify grâce à ce lien pour utiliser ce ChatBot\nLien :${link}/login`);
        } else {
            const genre = data.entities.groups.genre
            console.log(genre)
            spotifyApi.searchPlaylists(genre)
                .then(function(data) {
                        var num = Math.floor(Math.random() * 20) + 1;
                        console.log('Found playlists are', data.body.playlists.items[num]);
                        const nameplaylist = data.body.playlists.items[num].name
                        const urlplaylist = data.body.playlists.items[num].external_urls.spotify
                        FB.sendMessage("RESPONSE", userData.sender, `Parfait, nous vous conseillons donc cette playlist ${nameplaylist}\nVoici donc le lien : ${urlplaylist}`)
                    },
                    function(err) {
                        console.log('Something went wrong!', err);
                    });
        }
        break;
    }
```

It is a similar logic for the playlist suggestion, the user enters a phrase, and the regex model will find the keyword itself and send it to the playlist search function. Then, the chatbot will answer the user by proposing a link to access the playlist.



To sum up, this project was very interesting because it allowed us to discover things we didn't know. Indeed, learning how to interact with any user without any contact using applications that we use every day seems extremely complicated. However, when we really focus on the problem and find the solution to our question, everything becomes clearer.
Now we know that it is possible to create chatbots with the theme of our choice without any limits.
We hope that it will be useful to most people who want to design music tools, it remains an original way to contribute to the music business.