

Musify

This is just the beginning of something big....



Team member and Roles

My name is Linson DOUMATE, a aspirant Software engineer from the republic of Benin.

I choosed to work solo on this project to train myself, working on different things we learnt through this program.

However, I have gotten sometime help from some peer when I was in difficulty to fix some bugs or handle a part of my application.

Different roles :

- **Project Manager**
- **Frontend Developer**
- **Backend Developer**



Story

Everything starts here, when I watched for the first time, the series the playlist, that describes how the music application was born.

It was increadable, and I wanted to build a more complex but small application about music, and getting some data to display them. All about the artists, the albums and their music.

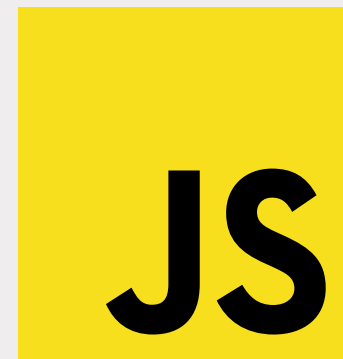
I wanted to know how to deal with data from artists, and how to play music online. I didn't really understood how to interact with an API, even after the last month projects. So...

When I saw the opportunity to build the research and approval project, I just decide to put myself to work, and try to get something from the Spotify API, to manage informations with my own backend.



Technologies and Architecture

Frontend technologies



Build the frontend structure, style and maintain consistent code base, and visual hierarchy.

Backend technologies



Flask

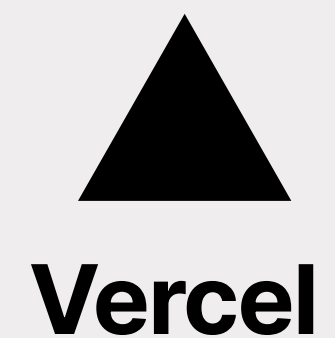


Spotify API



Handle routes, secured communications and interactions between the application and the API.

Deployment



Deploy the static landing page with vercel, and host all the project on pythonanywhere.

Algorithms and code snippets

```
14     __token = ""
15
16     Valeryio
17 > def __init__(self, app):...
18
19
20
21     1 usage Valeryio
22     @property
23 > def client_id(self):...
24
25     1 usage Valeryio
26     @property
27 > def client_secret(self):...
28
29     Valeryio
30     @property
```

OOP programming

When it comes to build a system, I think in oriented programming paradigm. So I built intern class like **class Spotify()** to handle different operations with the web api of the application.

```
*
te("/", strict_slashes=False, methods=["GET", "POST"])
te("/home", strict_slashes=False, methods=["GET", "POST"])
x():...

te("/artist/<string:search_data>", strict_slashes=False, methods=["GET", "POST"])
st_page(search_data):...

te("/songs/<artist_name>/<album_id>", strict_slashes=False, methods=["GET", "POST"])
s_page(artist_name, album_id):...
```

Routes management

I used flask and its flexibility to design and manage different routes for the application, to get the right informations and handle different **HTTP REQUESTS** as **GET** and **POST**.

Process collaboration and lifetime

Week 1

Figma design, and first draft of the idea I wanted to implement as project.

Simple design of the backend part.

Week 3

Frontend part of the project and integration of the different routes.

Building a simple MVP for the end of the project. Creation the landing page of my project.

Week 0

Ideation, and thinking about the project on its general aspect.

What should I build as project for the end of my SWE foundation program?

Week 2

Git repository for the project, and API testing to get the data to use for the backend.



Overcomed challenges

Technicals

Metric 1

I never used a CSS framework before this project. And I wasn't really used to use the node environment. It took me an entire day to figure out how to configure the right packages, create the right architecture and manage the frontend part easily.

Metric 2

Handling forms with Flask_wtf wasn't the most pleasant or easier part of my job. I took me two days to fix a bug about, how to send the name of an input, from a Jinja Template with Wtforms.

Non technicals

The major non-technical challenge I faced while building the project was the power cuts.

When you're not in a great context, you have to deal with this really stressful and situations that not depends on you.

Anyway, it justs reinforce your determination, and your resilience to get the job done!

Learning about technical interest

I really like to understand each part of the system, and make it so clean that it responds the most efficiently possible with the rest of an application. I'll probably choose the backend, and pursue my software engineer career with frontend skills to become a great **fullstack developer**.