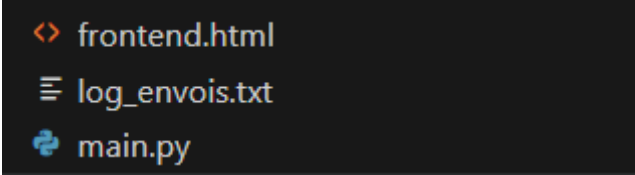


Rapport

## **Partie 01 :**

### **Exercice 1 – Création de l'API REST**

Fichier :



```
<> frontend.html
≡ log_envois.txt
+ main.py
```

**Objectif :** Afficher une liste statique de personnages fictifs.

**Tâches :**

- Création de main.py
- Initialisation de FastAPI
- Création d'un endpoint /personnages en méthode GET
- Retour d'une liste fixe (2+ personnages)

💡 **Code utilisé :**

**Dans fichier :** `main.py`

**Test :**

```
uvicorn main:app --reload
```

**Accès à :**

➡ <http://127.0.0.1:8000/docs>

### **Exercice 2 – Sécuriser l'accès avec un token**

**Objectif :** Autoriser uniquement les requêtes avec un token valide dans l'en-tête HTTP.

**Tâches :**

- Ajouter un Header pour le token
- Vérifier qu'il est égal à "token\_secret\_123"
- Retourner une erreur 401 Unauthorized sinon

default

GET

/personnages Get Personnages

Cancel

Parameters

Name	Description
prenom	<input type="text" value="prenom"/>
<small>string   (string   null) (query)</small>	
x-auth-token <small>* required</small>	<input type="text" value="x-auth-token"/>
<small>string (header)</small>	

Execute

Token : token\_secret\_123

Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8080/personnages' \
  -H 'accept: application/json' \
  -H 'x-auth-token: token_secret_123'
```

Request URL

```
http://127.0.0.1:8080/personnages
```

Server response

Code

Details

200

Response body

```
{
  "id": 1,
  "nom": "Harry Potter",
  "profession": "Sorcier",
  "age": 17,
  "univers": "Harry Potter",
  "score": 85,
  "niveau": null
},
{
  "id": 2,
  "nom": "Ron Weasley",
  "profession": "Sorcier",
  "age": 17,
  "univers": "Harry Potter",
  "score": 75,
  "niveau": null
},
{
  "id": 3,
  "nom": "Hermione Granger",
  "profession": "Sorcière",
  "age": 17,
  "univers": "Harry Potter",
  "score": 90,
  "niveau": null
}
```

Download

127.0.0.1:5500/frontend.html

Afficher les personnages

Harry Potter

Univers : Harry Potter

Profession : Sorcier

Âge : 17 ans

ID : 1

Ron Weasley

Univers : Harry Potter

Profession : Sorcier

Âge : 17 ans

ID : 2

Hermione Granger

Univers : Harry Potter

Profession : Sorcière

Âge : 17 ans

ID : 3

Albus Dumbledore

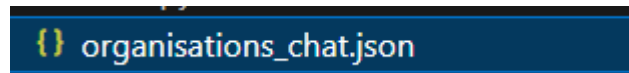
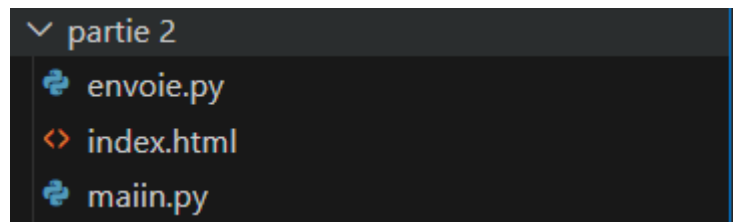
Univers : Harry Potter

Profession : Directeur de Poudlard

Âge : 115 ans

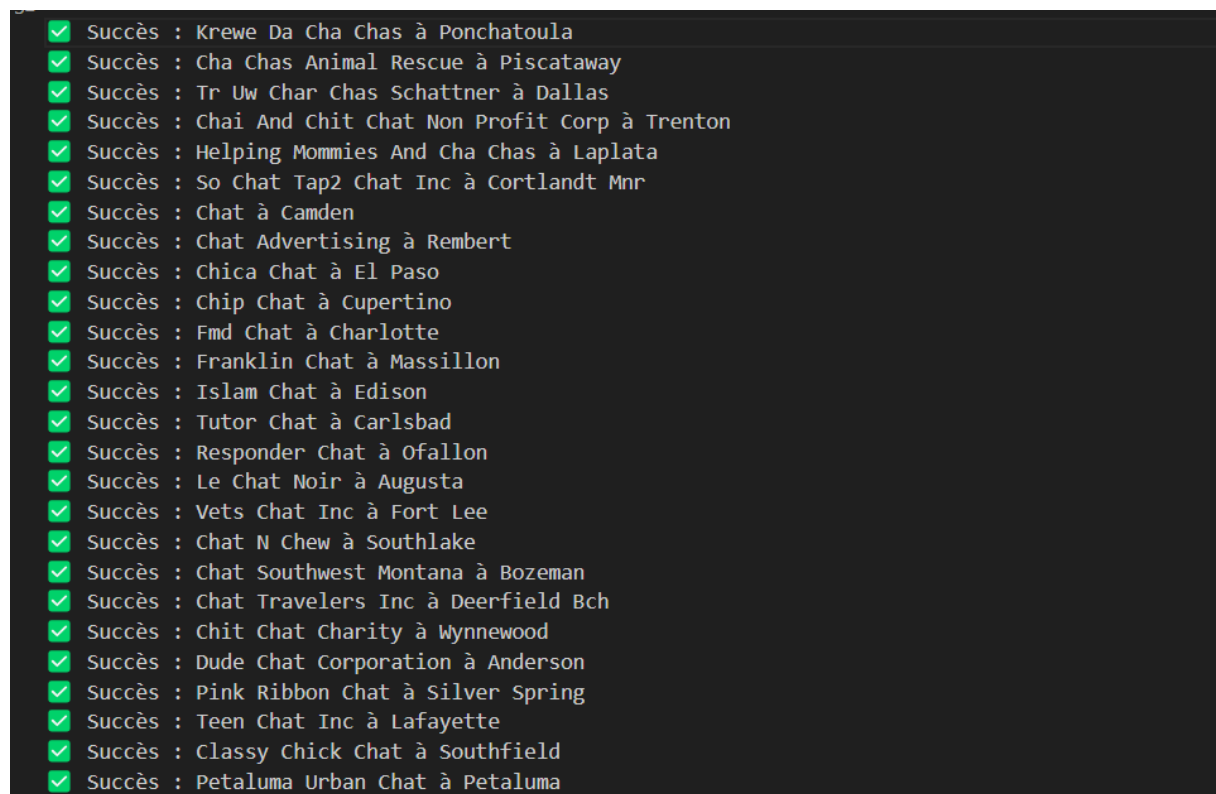
ID : 4

## Partie 02 :



Logs d'envoi :

```
PS C:\Users\debou\Desktop\TP Sol> & C:/Users/debou/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/debou/Desktop/TP Sol/partie 2/envoie.py"
Envoi des données à l'API...: 100% | 2481/2481 [20:59<00:00, 1.97it/s]
Tous les envois ont été traités. Voir log_envois.txt pour les détails.
PS C:\Users\debou\Desktop\TP Sol>
```



# Test API Personnages

Token d'authentification:

Filtrer par prénom:

Afficher tous les personnages

Filtrer par prénom

Cliquez sur un bouton pour charger les données

Filtrer par nom :

# Test API Personnages

Token d'authentification:

Filtrer par prénom:

Afficher tous les personnages

Filtrer par prénom

- Nom:** Albus Dumbledore  
**Profession:** Directeur de Poudlard  
**Âge:** 115  
**Univers:** Harry Potter

```
[
  {
    "ein": 831061850,
    "strein": "83-1061850",
    "name": "Krewe Da Cha Chas",
    "sub_name": "Krewe Da Cha Chas",
    "city": "Ponchatoula",
    "state": "LA",
    "ntee_code": "E12",
    "raw_ntee_code": "E12",
    "subseccd": 3,
    "has_subseccd": true,
    "have_filings": null,
    "have_extracts": null,
    "have_pdfs": null,
    "score": 75.61292
  },
  {
    "ein": 923163493,
    "strein": "92-3163493",
    "name": "Cha Chas Animal Rescue",
    "sub_name": "Cha Chas Animal Rescue",
    "city": "Piscataway",
    "state": "NJ",
    "ntee_code": "D99",
    "raw_ntee_code": "D99",
    "subseccd": 3,
    "has_subseccd": true,
    "have_filings": null,
    "have_extracts": null,
    "have_pdfs": null,
    "score": 75.61292
  }
]
```

Partie 3 :

```
[
  {
    "nom": "Voldemort",
    "score": 85,
    "niveau": "Fort"
  }
]
```



Nouveau personnage : Voldemort - Niveau Niveau.FORT

#### Partie 4 :

POST /traitement Traitement

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "id": 0,
  "nom": "string",
  "profession": "string",
  "age": 0,
  "univers": "string",
  "score": 0,
  "niveau": "string"
}
```

Un exemple

Avant

```
[
  {
    "id": 1,
    "nom": "Harry Potter",
    "profession": "Sorcier",
    "age": 17,
    "univers": "Harry Potter",
    "score": 85
  },
  {
    "id": 2,
    "nom": "Ron Weasley",
    "profession": "Sorcier",
    "age": 17,
    "univers": "Harry Potter",
    "score": 75
  }
]
```

Execute

Après :

Request URL

http://127.0.0.1:8080/traitement

Server response

Code Details

200

Response body

```
{
  "message": "Traitement réussi",
  "data": [
    {
      "id": 1,
      "nom": "Harry Potter",
      "profession": "Sorcier",
      "age": 17,
      "univers": "Harry Potter",
      "score": 85,
      "niveau": "expert"
    },
    {
      "id": 2,
      "nom": "Ron Weasley",
      "profession": "Sorcier",
      "age": 17,
      "univers": "Harry Potter",
      "score": 75,
      "niveau": "intermédiaire"
    }
  ]
}
```



Download

Response headers