

## **SEMAINE 1**

### Lundi 15 mai

#### Matin

Présentation avec l'équipe et avec le directeur du service. Visite des locaux, découvertes des différents projets que la MSHS créer/ étudie. Passage au service informatique pour récupérer un pc portable et un écran.

#### Après Midi

Début de l'installation, création du cahier des charges. Recherche sur ce qu'est Nakala, comment ça fonctionne de même pour Quarto. (Nakala accessible seulement avec des autorisations et une demande d'accès rattacher à l'établissement). Nakala est une base de données de recherche permettant aux chercheurs/ professeurs de déposer leurs projets. Le service Nakala offre deux niveaux de préservation :

- Un niveau par défaut mis en pratique dès lors qu'une donnée est enregistrée dans Nakala. La donnée est décrite, contextualisée et stockée de manière sécurisée.
- Un niveau avancé qui s'inscrit dans un partenariat avec le CINES. Dans ce circuit de dépôt avancé, la préservation à long terme est assurée par le CINES. Le CINES apporte des garanties supplémentaires telles que la valeur juridique du dépôt, l'intégrité du fichier, la migration des formats pour leur lisibilité, etc.

L'équipe aimerait que je crée un prototype de génération de site web statique à partir de Nakala pour permettre aux professeurs de déposer leurs projets

Quarto est un outil permettant de coder simplement le problème et qu'il n'y a quasiment pas de vidéo ou de cours en ligne pour sinicier.

## Mardi 16 mai

### Matin

Reprise de la recherche sur Quarto et Nakala. Installation de l'extension Quarto sur Vscode mais beaucoup d'erreurs, je n'arrive pas à exécuter mon code en .qmd aucune possibilité de débogage car il n'y a pas d'extension de débogage pour Quarto sur Vs code. Voici le message affiché quand j'essaie d'ouvrir mon fichier de code :

Cannot GET /venv/lib/test.qmd

### Fin de matinée

Réunion avec l'équipe pour parler des zones d'ombres sur le projet et planification du début du cahier des charges.

Solution à mon problème ci-dessus.

### Après midi

Benchmarking sous forme de tableau des différents générateurs de site web statique disponible sur le marché.

	Quarto	Jekyll	Hugo	Eleventy
Avantages	- Syntaxe simple et intuitive	- Grande communauté de développeurs	- Performance élevée	- Flexibilité et liberté de personnalisation
	- Intégration aisée avec Jupyter	- Nombreux plugins et thèmes disponibles	- Structure de fichiers flexible	- Performances élevées
	- Génération de rapports PDF et HTML	- Facilité d'utilisation pour les blogs	- Prise en charge multilingue	- Prise en charge de plusieurs langages
	- Prise en charge des langages de programmation	- Mise en page flexible avec Liquid	- Grande communauté de développeurs	- Documentation détaillée

Avancement sur le cahier des charges avec mon tuteur et hypothèse de choix final pour l'outil à choisir.

## Mercredi 17 mai

### Matin

Modification du tableau de benchmarking car pas assez clair. Modifications du cahier des charges de quelques éléments avec mon tuteur.

### Après midi

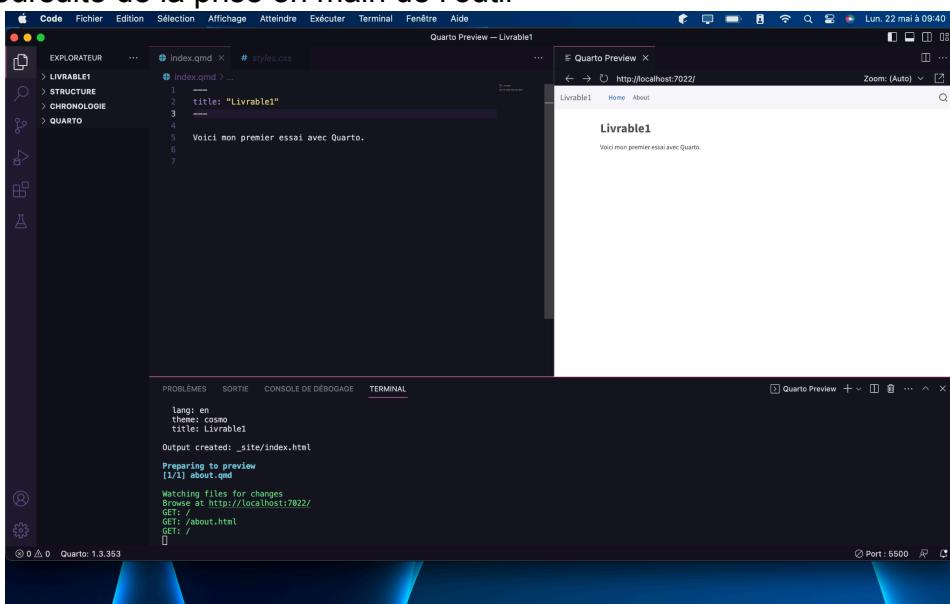
Premier contact avec l'outil Quarto. Visualisations de tuto en ligne et sur le site (quarto.org). J'ai commencé par essayer de voir ce qu'était le Markdown et comment l'outil l'utilise. j'ai ensuite observé les différents exemples de langages utilisables. Python est à mon avis le plus logique à utiliser. Pour finir j'ai observé différents exemples de projets réalisables avec Quarto. En fin d'après-midi j'ai complété le premier livrable de mon cahier des charges.

## Semaine 2

### Lundi 22 Mai

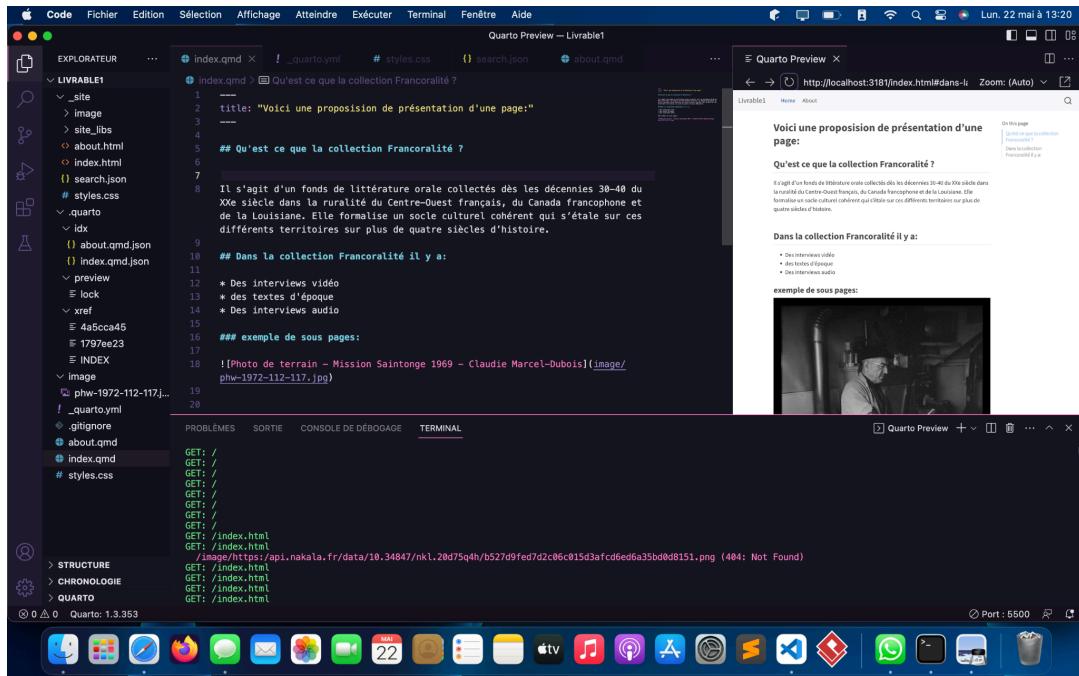
#### Matin

Début du livrable 2 consistant à créer une simple page en utilisant l'outil Quarto et poursuite de la prise en main de l'outil



## Après midi

Confection d'une page dite "model" pour la suite.

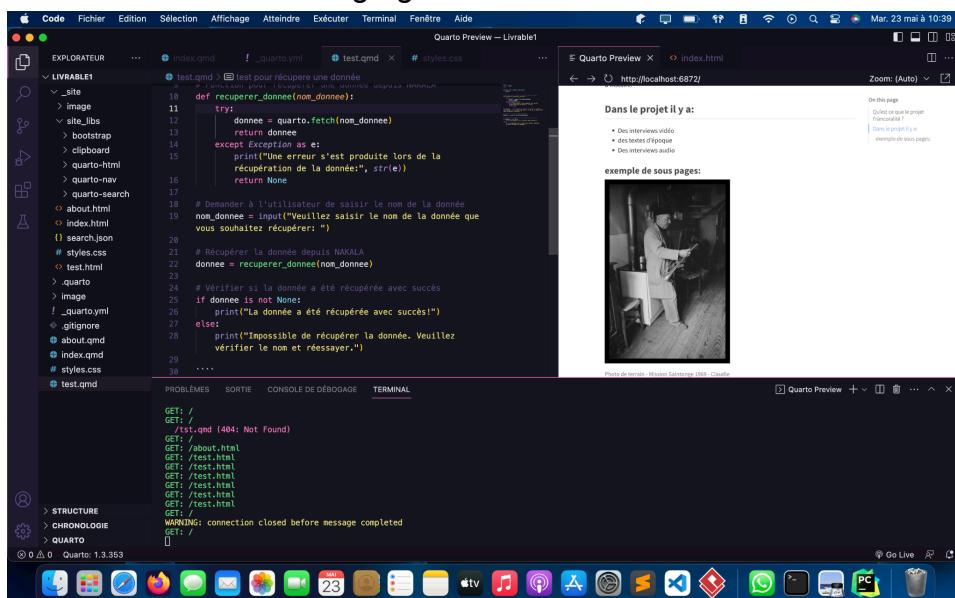


```
index.qmd
1 ---  
2 title: "Voici une proposition de présentation d'une page:"  
3 ---  
4  
5 ## Qu'est ce que la collection Francoralité ?  
6  
7 Il s'agit d'un fonds de littérature orale collectés dès les décennies 30-40 du XXe siècle dans la ruralité du Centre-Ouest français, du Canada francophone et de la Louisiane. Elle formalise un socle culturel cohérent qui s'étale sur ces différents territoires sur plus de quatre siècles d'histoire.  
8  
9 ## Dans la collection Francoralité il y a:  
10  
11 * Des interviews vidéo  
12 * des textes d'époque  
13 * Des interviews audio  
14  
15  
16 ### exemple de sous pages:  
17  
18 ! [Photo de terrain - Mission Saintonge 1969 - Claudio Marcel-Dubois] (image/phw-1972-112-117.jpg)  
19  
20  
PROBLÈMES SORTIE CONSOLE DE DÉBOGAGE TERMINAL
```

## Mardi 23 mai

### Matin et apres midi

Conception d'une fonction python permettant de récupérer sur l'entrepôt de données nakala une image grâce à son DOI.



```
test.qmd
10 def recuperer_donnee(nom_donnee):
11     try:
12         donnee = quarto.fetch(nom_donnee)
13         return donnee
14     except Exception as e:
15         print("Une erreur s'est produite lors de la récupération de la donnée.", str(e))
16         return None
17
18 # Demander à l'utilisateur de saisir le nom de la donnée
19 non_donnee = input("Veuillez saisir le nom de la donnée que vous souhaitez récupérer: ")
20
21 # Récupérer la donnée depuis NAKALA
22 donnee = recuperer_donnee(non_donnee)
23
24 # Vérifier si la donnée a été récupérée avec succès
25 if donnee is not None:
26     print("La donnée a été récupérée avec succès!")
27 else:
28     print("Impossible de récupérer la donnée. Veuillez vérifier le nom et réessayer.")
29
30
GET: /  
GET: /_quarto.yml (404: Not Found)  
GET: /about.html  
GET: /about.html  
GET: /test.html  
GET: /test.html  
GET: /test.html  
GET: /test.html  
GET: /test.html  
WARNING: connection closed before message completed  
GET: /
```

J'ai eu plein de soucis liés à l'installation de Python. Comme ci je ne l'avais pas installer alors que si.

### **Mercredi 24 mai**

#### **Matin**

J'ai poursuivi dans la création de ma fonction python permettant de récupérer une image sur l'entrepôt de données. J'ai quelques soucis avec python sur ma machine. Cela m'a pris toute la matinée mais j'ai trouvé le problème:  
il fallait changer le chemin d'accès à python dans le bashrc.

#### **Après-midi**

Poursuite sur la fonction **download images from doi**.

En faisant une recherche d'exemple de script pour récupérer le contenu d'une page web en HTML je suis tombé sur un article de scraping avec python très intéressant qui m'a permis de bien comprendre comment je devais mettre en place ma fonction pour qu'elle soit le plus simple et efficace possible.

En fin d'après midi j'ai recherché les classes à récupérer dans nakala pour pouvoir les transférer dans mon site fait avec quarto. Je pense avoir récupérer les éléments les plus importants pour ne pas surcharger la page et le transfert donc j'ai noté:

- titre
- citer
- collection
- image
- dépositaire

Jeudi 25 mai

Matin

J'ai continué à regarder le scraping et à créer une page allant chercher tout ce qui est sur nakala grâce aux librairies request, beautifulsoup et panda.

The screenshot shows a Mac desktop with a dark-themed interface. In the center, there are two Geany code editors. The left editor contains Python code named 'testscarp.py' which imports urllib.request, bs4, and pandas, then sends a GET request to a Nakala URL and prints the BeautifulSoup object. The right editor shows the output of a shell script named 'geany\_run\_script\_E00541.sh', which is a JSON-like document with fields like 'Titre', 'Auteur', 'Date de création', and 'Licence'. A terminal window titled 'Photo de terrain - Mission S' is also visible, displaying the same JSON data. The desktop dock at the bottom has icons for various applications like Mail, Safari, and Finder.

```
import urllib.request
from bs4 import BeautifulSoup
import pandas as pd

url = 'https://www.nakala.fr/10.34847/nkl.20d75q4h'
page = urllib.request.urlopen(url, timeout=5)
soup = BeautifulSoup(page)
print(soup)
```

```
aïntonge 1969 - Enquête UPOI_ATP_0001_0012
    Titre
    François
    Auteure
    Claudio Marcel-Dubois
    Date de création
    1969-09-13
    Licence
    Etalab Open License 2.0 <a href="https://spdx.org/licenses/etalab-2.0.html#licenseText" target="_blank">(etalab-2.0)</a>
```

### Après midi

Dans le suite de ce que j'ai fait ce matin je vais "filtrer" les éléments que je veux récupérer qui sont:

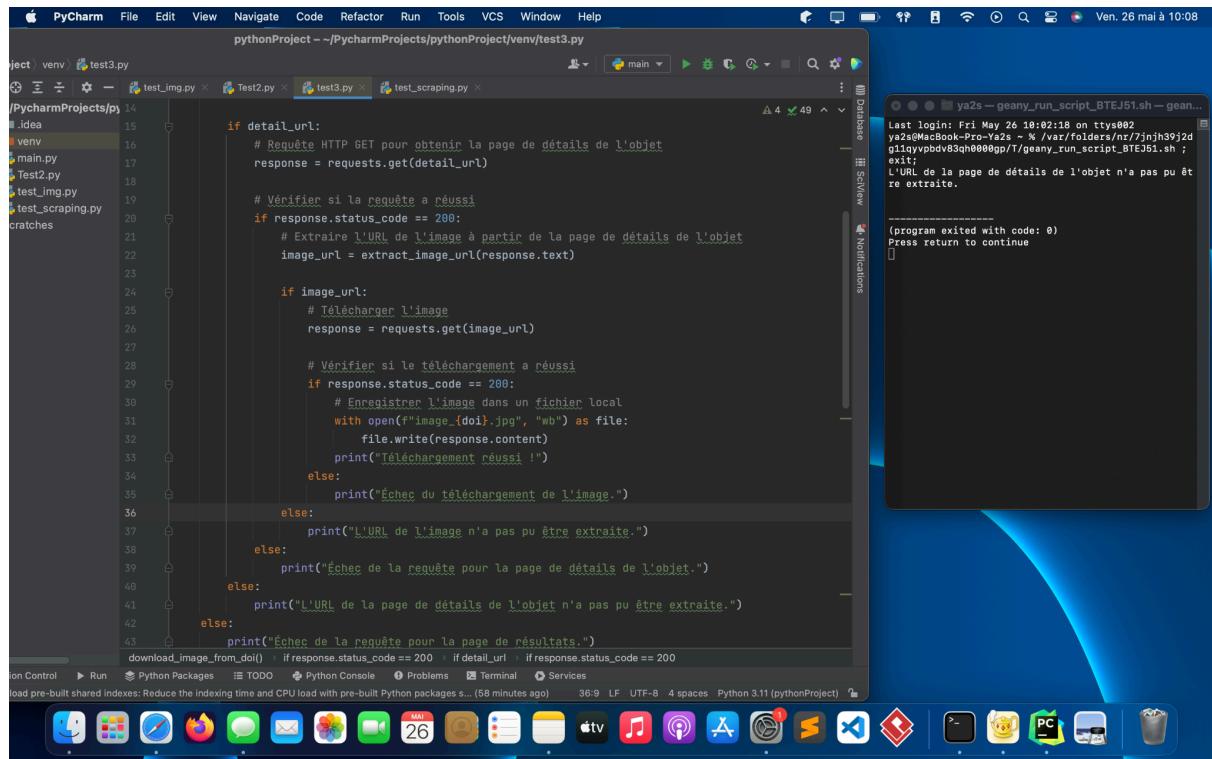
- le titre
- la description
- les images

Dans la fin d'après-midi mon tuteur m'a donné de la documentation pour l'API de NAKALA. Il m'a dit que c'était une bonne initiative de s'orienter vers le scraping mais que dans le projet il est plus logique et simple d'utiliser des scripts et des bibliothèques créés par l'équipe. Je repars donc de "zéro" sur le python alors que j'avais bien commencer.

## Vendredi 26 mai

### Matin

Nouvelle fonction python permettant de récupérer les informations d'une page de Nakala grâce à son url et son DOI. Je bloque car cela me sors un message d'erreur qui est : L'url de la page de détails de l'objet n'a pas pu être extraite.



The screenshot shows the PyCharm IDE interface. The left sidebar displays a project structure with files like main.py, Test2.py, test\_scraping.py, and test3.py. The main editor window contains Python code for scraping Nakala details. The terminal window at the bottom right shows the output of a command, indicating that the URL of the object details page could not be extracted. The status bar at the bottom shows the date as Vend. 26 mai à 10:08.

```
if detail_url:
    # Requête HTTP GET pour obtenir la page de détails de l'objet
    response = requests.get(detail_url)

    # Vérifier si la requête a réussi
    if response.status_code == 200:
        # Extraire l'URL de l'image à partir de la page de détails de l'objet
        image_url = extract_image_url(response.text)

        if image_url:
            # Télécharger l'image
            response = requests.get(image_url)

            # Vérifier si le téléchargement a réussi
            if response.status_code == 200:
                # Enregistrer l'image dans un fichier local
                with open(f'image_{doi}.jpg', "wb") as file:
                    file.write(response.content)
                print("Téléchargement réussi !")
            else:
                print("Echec du téléchargement de l'image..")
        else:
            print("L'URL de l'image n'a pas pu être extraite..")
    else:
        print("Echec de la requête pour la page de détails de l'objet..")
else:
    print("L'URL de la page de détails de l'objet n'a pas pu être extraite..")
else:
    print("Echec de la requête pour la page de résultats..")
```

En fin de matinée j'ai fait une réunion avec le reste de l'équipe pour voir où est ce que j'en été sur le projet. Ils m'ont demandé de finir le prototype de site avec Quarto . Je l'ai déposé en début d'après-midi.

## Après midi

J'ai fini le prototype de site avec Quarto et j'ai commencé à faire du python. À l'aide d'une librairie créée par un membre de l'équipe j'ai pu me mettre en "contact" avec L'API de NAKALA et faire des requêtes de récupération.

The screenshot shows a terminal window with the following details:

- Terminal Title:** testcmd.py - /Users/ys2s/Desktop - Geany
- Code Content:** The code is a Python script named `testcmd.py`. It imports `nakalapycon`, `nkt`, `nktAPI`, `nktData`, `nktProd`, and `nktUtil`. It defines a `dataIdentifier` variable and a `targetProd` variable. The script then loops through `rs[0]` and `rs[1]` to print `listecollect` for each item. It also prints `listefich` and `listefich` for each item in `listefich`. Finally, it loops through `rs[2]` to print `dictVals` for each item.
- Output:** The output shows the execution of the script. It lists several files (e.g., `ph-1972-112-117.jpg`, `ph-1972-112-118.jpg`) and their details (e.g., Albert Maissen, informateur musicien). It also shows the command `geany_run_script_DPWK61.sh` being run.
- Bottom Status Bar:** The status bar shows the terminal number (sel:0), column (col:158), line (line:22/53), and mode (mode:LF).

## Semaine 3

**Mardi 30 mai**

## Matin

Dans la continuité de ce que j'ai fait vendredi, je poursuis sur la récupération de données. L'objectif est de récupérer l'url de chaque image.

## Après midi

Réflexion sur la liaison entre le script python, la récupération sur l'API de NAKALA et mon site créer grâce à quarto.

Voici la première version de mon programme qui permet grâce à l'importation de la bibliothèque markdown de transformer les informations récupérées en markdown puis de les transformer en html.

The screenshot shows the Geany IDE interface with the following details:

- Title Bar:** testlien.py - /Users/ya2s/Desktop - Geany
- File List:** Test2.py, test.py, testcarp.py, testenv.py, testmd.py, testlien.py, testlien.html
- Code Editor:** The code is a Python script named testlien.py. It uses the markdown library to parse multiple Markdown files and generate a structured output. The code includes imports for os, re, json, requests, and nklTtarget. It defines a function rs.md\_to\_markdown that takes a targetProd and dataIdentifier as parameters. The main part of the script iterates through collections, files, and images, extracting titles, descriptions, and URLs, and then concatenating them into a single Markdown string. Finally, it converts the entire string to HTML using markdown.markdown and prints the result.
- Status Bar:** Ligne : 34 / 47, col : 0, sel : 0, INS TAB mode : LF codage : UTF-8 type de fichier : Python portée : inconnu

voilà ce que cela renvoie :

Geany Fichier Éditer Rechercher Affichage Document Projet Construire Outils Aide

testLien.html - /Users/ya2s/Desktop - Geany

```
< Symbols > Test.py x test3.py x testscrap.py x testvend.py x testmod.py x testlien.py x testLien.html x
```

▼ Tress H2

```
<2>-> liste collections:</2>
1 <1>-> liste collections:</1>
2 <1>-> 18_34847/nkl.fabdg38</1>
3 <1>-> 18_34847/nkl.fabdg38</1>
4 <1>-> 18_34847/nkl.fef2bb2b</1>
5 <1>-> liste fichiers:</1>
6 <2>-> liste fichiers:</2>
7 <ul>
8   <li><strong>Nom</strong>: phw-1972-112-117.jpg
9   <strong>Description</strong>: Albert Maisan, informateur musicien
10  <strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
11  <li><strong>Nom</strong>: phw-1972-112-118.jpg
12  <strong>Description</strong>: Albert Maisan jouant de la trompette d'harmonie
13  <strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
14  <li><strong>Nom</strong>: phw-1972-112-119.jpg
15  <strong>Description</strong>: Albert Maisan jouant de la trompette d'harmonie
16  <strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
17  <li><strong>Nom</strong>: phw-1972-112-120.jpg
18  <strong>Description</strong>: Albert Maisan jouant du clairon
19  <strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
20 </ul>
21 <ul>
22   <li><strong>liste images:</strong></li>
23 <li><strong>Citation</strong>: Marcel-Dubois, Claude (2021) «photo de terrain - Mission Saintonge 1969 - Enquête UPOI_ATP_0001_0012» [Image] NAKALA. https://doi.org/10.34847/nkl.28d75qh
24 <li><strong>Citation</strong>: Marcel-Dubois, Claude (2021) «photo de terrain - Mission Saintonge 1969 - Enquête UPOI_ATP_0001_0012» [Image] NAKALA. https://doi.org/10.34847/nkl.28d75qh
25 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
26 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
27 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
28 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
29 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
30 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
31 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
32 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
33 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
34 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
35 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
36 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
37 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
38 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
39 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
40 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
41 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
42 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
43 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
44 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
45 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
46 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
47 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
48 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
49 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
50 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
51 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
52 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
53 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
54 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
55 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
56 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
57 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
58 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
59 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
60 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
61 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
62 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
63 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
64 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
65 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
66 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
67 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
68 <li><strong>URI</strong>: https://doi.org/10.34847/nkl.28d75qh</li>
```

Il faut que je puisse mettre en relation mes scripts python et mon site quarto pour faire de l'auto génération de page.

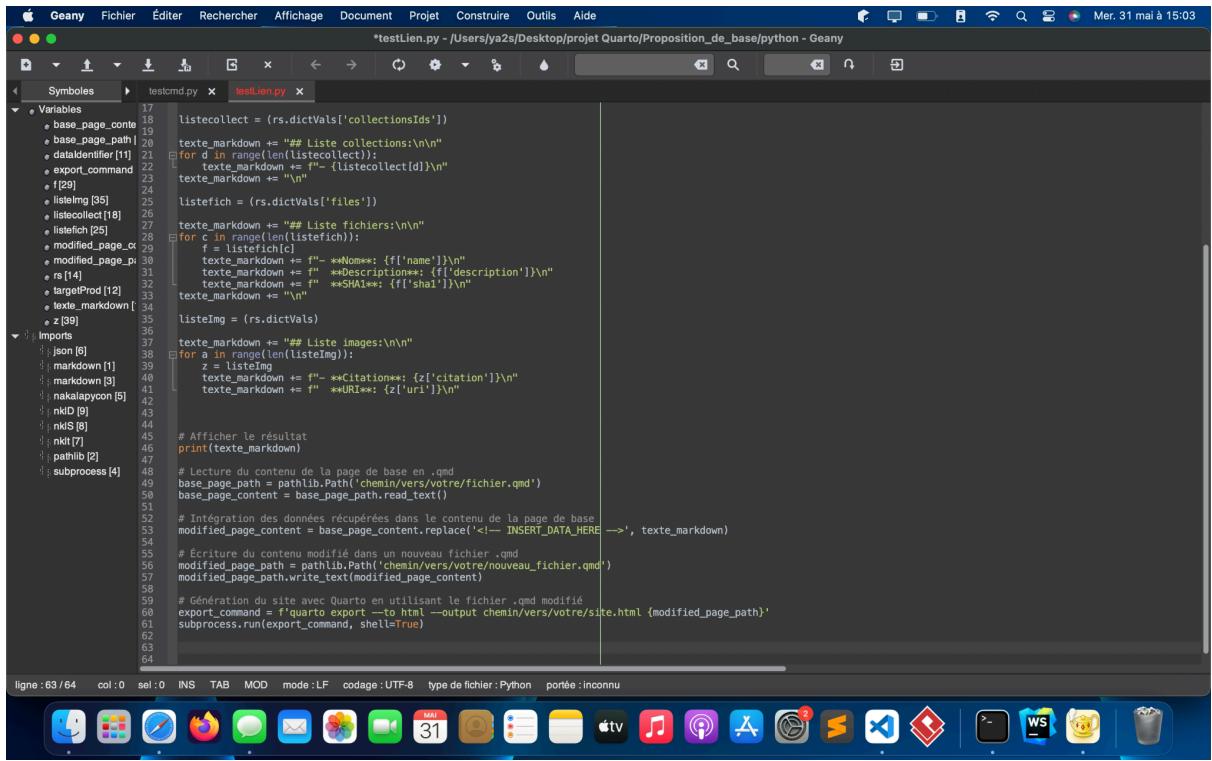
## Mercredi 31 Mai

### Matin

Dans la poursuite de ce que j'ai fais hier je vais faire en sorte d'automatiser le processus pour que mon script python soit directement injecté dans mon .qmd une fois transformé en markdown.

### Après midi

Après avoir regardé des exemples sur internet et comment faire pour automatiser tout cela j'essaie d'optimiser mon code pour que tout cela soit fonctionnel.



The screenshot shows the Geany IDE interface with the file `testLien.py` open. The code is a Python script that reads a QMD file, extracts collections and files, and generates a corresponding Markdown document. It uses the `pathlib` and `subprocess` modules to handle file paths and execute commands like `quarto export --to html`. The code is well-structured with comments explaining its purpose.

```
testcmd.py x testLien.py x
Variables
    base_page_content [1]
    base_page_path [1]
    dataIdentifier [1]
    export_command [1]
    f [29]
    listeing [35]
    listecollect [18]
    listefich [25]
    modified_page_content [1]
    modified_page_file [1]
    rs [14]
    targetProd [12]
    texte_markdown [1]
    z [39]
Imports
    json [6]
    markdown [1]
    markdown [3]
    nakalapyoncon [5]
    nikD [9]
    nikS [8]
    nkit [7]
    pathlib [2]
    subprocess [4]
17
18     listecollect = (rs.dictVals['collectionsIds'])
19
20     texte_markdown += "# Liste collections:\n\n"
21     for d in range(len(listecollect)):
22         texte_markdown += f"- {listecollect[d]}\n"
23     texte_markdown += "\n"
24
25     listefich = (rs.dictVals['files'])
26
27     texte_markdown += "# Liste fichiers:\n\n"
28     for c in range(len(listefich)):
29         f = listefich[c]
30         texte_markdown += f"- **Nom**: {f['name']}\n"
31         texte_markdown += f"- **Description**: {f['description']}\n"
32         texte_markdown += f"- **SHA1**: {f['sha1']}\n"
33     texte_markdown += "\n"
34
35     listeImg = (rs.dictVals)
36
37     texte_markdown += "# Liste images:\n\n"
38     for a in range(len(listeImg)):
39         z = listeImg[a]
40         texte_markdown += f"- **Citation**: {z['citation']}\n"
41         texte_markdown += f"- **URI**: {z['uri']}\n"
42
43     # Afficher le résultat
44     print(texte_markdown)
45
46     # Lecture du contenu de la page de base en .qmd
47     base_page_path = pathlib.Path('chemin/vers/votre/fichier.qmd')
48     base_page_content = base_page_path.read_text()
49
50     # Intégration des données récupérées dans le contenu de la page de base
51     modified_page_content = base_page_content.replace('<!-- INSERT_DATA_HERE -->', texte_markdown)
52
53     # Écriture du contenu modifié dans un nouveau fichier .qmd
54     modified_page_path = pathlib.Path('chemin/vers/votre/nouveau_fichier.qmd')
55     modified_page_path.write_text(modified_page_content)
56
57     # Génération du site avec Quarto en utilisant le fichier .qmd modifié
58     export_command = f'quarto export --to html --output chemin/vers/votre/site.html {modified_page_path}'
59     subprocess.run(export_command, shell=True)
60
61
62
63
64
```

De ce que j'ai vu un utilisant la librairie `pathlib` et `subprocess` cela pourrait être fonctionnel. Je vais faire les tests nécessaires.

Dans la fin d'après midi j'ai vu avec mon tuteur et il m'a aidé à faire le remplacement et a consolidé ma base.

Jeudi 1 Juin

# Matin

je reprends là où je me suis arrêter hier c'est-à -dire le processus de remplacement de la page de base.

J'ai fait le code pour le remplacement de l'auteur, du DOI, de la date de création, du dépositaire.

## Après-midi

Je poursuis sur ce que j'ai fait durant la matinée dans l'objectif de pouvoir intégrer les photos des pages.

Vendredi 2 Juin

Matin

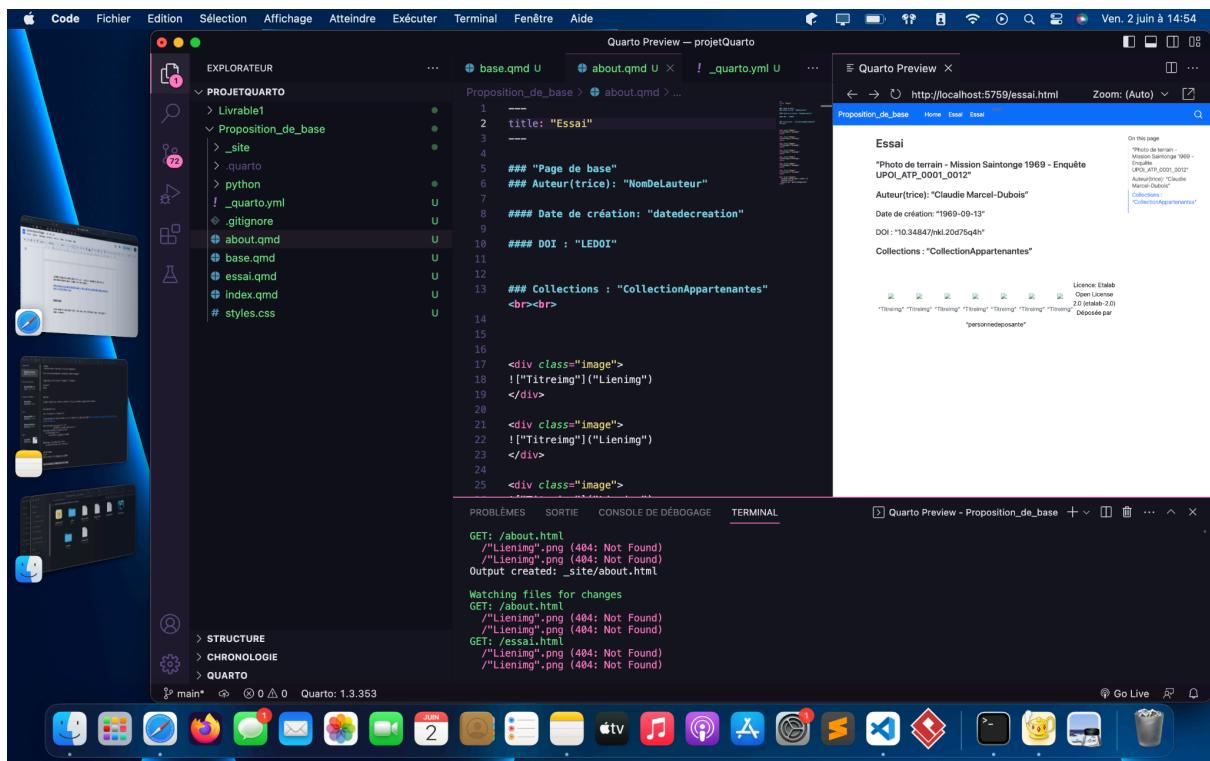
Je continue sur la récupération et l'insertion dans ma page de base. j'ai vu avec mon tuteur et il m'a expliqué comment fonctionnait l'irl d'une image sur nakala :  
<https://api.nakala.fr/iiif/doi/sha1/default.jpg>

Le DOI doit être récupérés et le sha1 aussi il est aussi possible de donner des paramètres d'affichage à la suite du SHA1 exemple :

<https://api.nakala.fr/iiif/11280/111643de/b11ddf74a574b13ab04d790ebfd6d643db96eaed/full/max/45/default.jpg>

## Après-midi

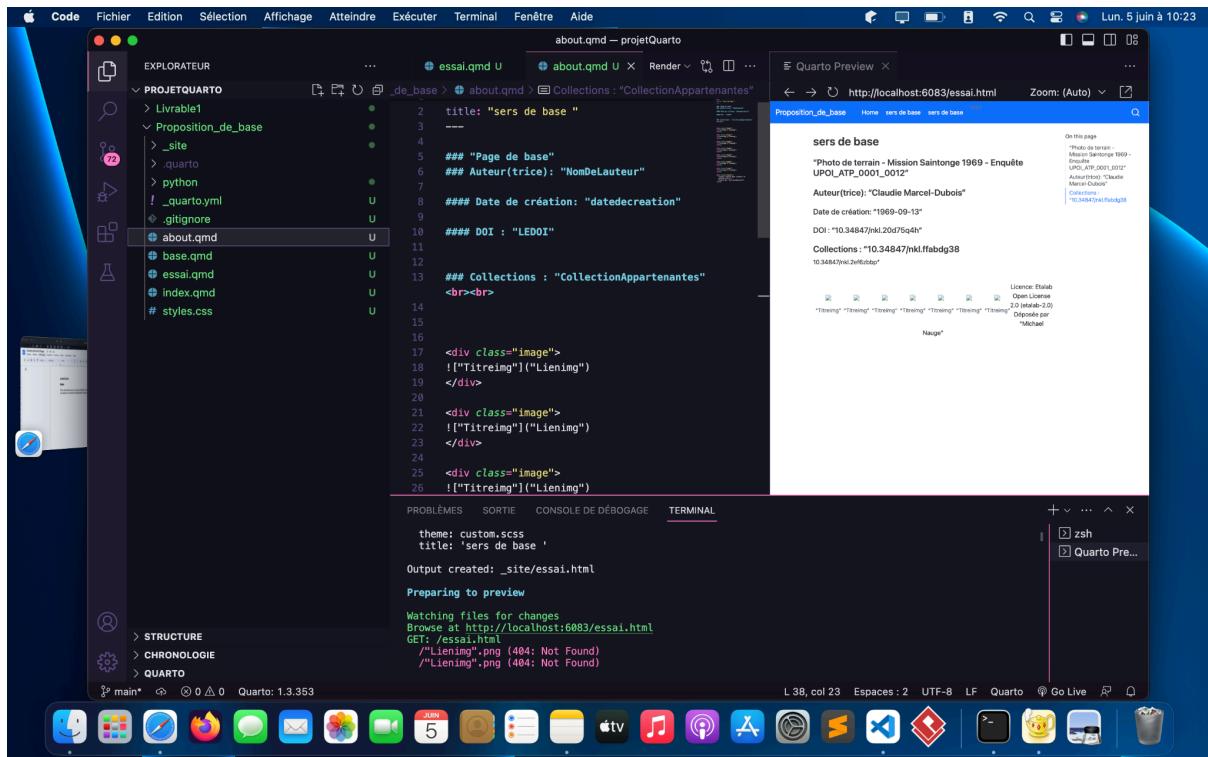
Je poursuis sur la recherche de la récupération et de l'insertion dans ma page de base. Je bloque :



## Lundi 5 juin

### Matin

Pour poursuivre sur ce que je faisais vendredi,  
j'ai repris et j'ai fait la récupération des collections appartenantes.



The screenshot shows the Geany Python IDE with a script named 'testLiens.py' open. The code uses the requests library to interact with an API endpoint to get data about a collection and then prints it to the console. The code includes imports for requests, json, and os, and defines functions for getting data and printing results.

```
import requests
import json
from os import path

# Variables
base_page_contenu = "http://localhost:6083/_site"
base_page_identi = "http://localhost:6083/_de_base"
collection_string = "CollectionAppartenantes"
dataIdentifier = "10.34847/nkl.20d75q4h"
modified_page_cr = "http://localhost:6083/_de_base/_site/essai.html"
nkAPI_Search = "http://localhost:6083/_de_base/_site/essai.html"
nkAPI_Data = "http://localhost:6083/_de_base/_site/essai.html"

# fonction pour faire une requête à l'API
def targetProd(isNakalaProd=True, apiKey = ""):
    r = nkAPI_Search(dataIdentifier)
    rs = nkAPI_Data(targetProd, dataIdentifier)

# fonction pour extraire les données
def extractData(r):
    rs = targetProd()
    rs = rs.json()
    listeImg = []
    listeCollect = []
    listeFich = []

    # Extraction des images
    for i in range(len(rs['files'])):
        listeImg.append(rs['files'][i])

    # Extraction des collections
    for i in range(len(rs['collections'])):
        listeCollect.append(rs['collections'][i])

    # Extraction des fichiers
    for i in range(len(rs['files'])):
        listeFich.append(rs['files'][i])

    return listeImg, listeCollect, listeFich
```

Je vais essayer de reprendre sur l'insertion des images.

J'ai réussi à récupérer et construire l'url de récupération d'image:

```
36     listefich = (rs.dictVals['files'])
37
38     for c in range (len(listefich)):
39         f = listefich[c]
40         print("https://api.nakala.fr/iiif/", dataIdentifier, "/", (f["sha1"]), "/default.jpg")
41
42
43
44
45     #url_img= "hhttps://api.nakala.fr/iiif/", dataIdentifier, "/", sha1, "/default.jpg"
46
47     #print("_____ -", url_img)
48     """
```

Voilà ce que cela renvoie:

```
Last login: Mon Jun  5 10:46:00 on ttys011
ya2s@MacBook-Pro-Ya2s ~ % /var/folders/nr/7jnjh39j2dg11qvpbdv83qh0000gp/T/geany_run_script_VJQT51.sh ; exit;
https://api.nakala.fr/datas/10.34847/nkl.20d75q4h/metadata
https://api.nakala.fr/iiif/ 10.34847/nkl.20d75q4h / b527d9fed7d2c06c015d3afcd6ed6a35bd0d8151 /default.jpg
https://api.nakala.fr/iiif/ 10.34847/nkl.20d75q4h / 384c5038e9ec76c17cbd5b0b49e1e71ac0189d26 /default.jpg
https://api.nakala.fr/iiif/ 10.34847/nkl.20d75q4h / ea6eca0c44f080864f7fd57b49d0644bd143013c /default.jpg
https://api.nakala.fr/iiif/ 10.34847/nkl.20d75q4h / c4ddeb39bb099b35c15f9c10735658a68e57cb9 /default.jpg

-----
(program exited with code: 0)
Press return to continue
```

## Après midi

Mon tuteur est venu voir comment est-ce que j'avançais. En vérifiant ce que j'avais fait, on s'est rendu compte que la façon que nous avons opté de procéder dans notre code nous limitait. En effet, en faisant de la récupération et du remplacement obligatoire le programme demande des arguments entrer en dur et donc il n'est pas automatisé. Voici comment nous avons commencé à changer le code :

```
# Lecture du contenu de la page de base en .qmd
base_page_path = pathlib.Path('../about.qmd')
base_page_content = ""
modified_page_content = ""

with open(base_page_path) as f:
    for line in f:
        if(line.find("Page de base")):
            modified_page_content += line.replace('Page de base', texte_markdown_titre)
        elif(line.find("NomDeAuteur")):
            print("a \n")
            modified_page_content += line.replace('NomDeAuteur', texte_markdown_NomDeAuteur)

        elif(line.find("datedecreation")):
            modified_page_content += line.replace('datedecreation', texte_markdown_date_creation)

        elif(line.find("LEDOI")):
            modified_page_content += line.replace('LEDOI', dataIdentifier)

        elif(line.find("CollectionAppartenantes")):
            for z in len(rs.dictVals['collectionIds']):
                modified_page_content += line.replace('CollectionAppartenantes', rs.dictVals['collectionIds'][z])

        else:
            modified_page_content += line
```

De cette façon si jamais le champs à remplacer n'est pas trouver dans la recherche il sortira le champs dit "de base"

## Mardi 6 Juin

### Matin

Problème sur le code, le changement de mon titre s'opère mais la suite ne fonctionne pas . Il ne rentre pas dans la suite (elif).

```
with open(base_page_path) as f:
    for line in f:
        if(line.find("Page de base")):
            modified_page_content += line.replace('Page de base', texte_markdown_titre)
        elif(line.find("NomDeAuteur")):
            print("a \n")
            modified_page_content += line.replace('NomDeAuteur', texte_markdown_NomDeAuteur)

        elif(line.find("datedecreation")):
            modified_page_content += line.replace('datedecreation', texte_markdown_date_creation)

        elif(line.find("LEDOI")):
            modified_page_content += line.replace('LEDOI', dataIdentifier)

        elif(line.find("CollectionAppartenantes")):
            for z in len(rs.dictVals['collectionIds']):
                modified_page_content += line.replace('CollectionAppartenantes', rs.dictVals['collectionIds'][z])

        else:
            modified_page_content += line
```

Voici la nouvelle version qui devrait mieux fonctionner :

```
with open(base_page_path) as f:
    for line in f:
        if "Page de base" in line:
            modified_page_content += line.replace('Page de base', texte_markdown_titre)
        elif "NomDeAuteur" in line:
            print("a \n")
            modified_page_content += line.replace('NomDeAuteur', texte_markdown_NomDeAuteur)
        elif "datedecreation" in line:
            modified_page_content += line.replace('datedecreation', texte_markdown_date_creation)
        elif "LEDOI" in line:
            modified_page_content += line.replace('LEDOI', dataIdentifier)
        elif "CollectionAppartenantes" in line:
            for z in range(len(rs.dictVals['collectionIds'])):
                modified_page_content += line.replace('CollectionAppartenantes', rs.dictVals['collectionIds'][z])
        else:
            modified_page_content += line
```

J'ai souvent des erreurs de ce type :

```
Last login: Tue Jun  6 10:59:17 on ttys001
ya2s@MacBook-Pro-Ya2s ~ % /var/folders/nr/7jnjh39j2dg11qyvpbdv83qh0000gp/T/geany_run_script_EWHN51.sh ; exit;
https://api.nakala.fr/datas/10.34847/nkl.20d75q4h/metadatas
a

Traceback (most recent call last):
  File "/Users/ya2s/Desktop/projetQuarto/Proposition_de_base/python/testLien.py", line 62, in <module>
    for z in range(len(rs.dictVals['collectionIds'])):
KeyError: 'collectionIds'
```

Pourtant l'orthographe de collectionIds est respecté

## Après midi

Rendez Vous médical extérieur

## Mercredi 7 juin

### Matin

Débogage de la fin de ma fonction pour permettre le bon fonctionnement de mon programme en réglant le soucis de collectionIds.

Soucis résolu l'erreur était dans l'orthographe collectionIds s'est écrit collectionsIds. La nouvelle fonction que j'avais entrepris hier fonctionne. Je continue dans la poursuite du mon cahier des charges, la prochaine étape est de récupérer une donnée audio.

Voici tous les livrables que j'ai à réaliser sur le stage:

**Livrable 1: benchmarking des outils et technologies disponibles**

**Livrable 2: premier pas avec l'outil (obtenir une chaîne de production minimale fonctionnelle)**

**Livrable 3: croquis d'une page de “base” complète (saisie manuelle d'une donnée stockée dans Nakala)**

**Livrable 4 A: publier une Data (de type image) en provenance de Nakala grâce à un script qui génère le fichier de base**

**Livrable 4 B: publier une Data (de type audio) en provenance de Nakala grâce à un script qui génère le fichier de base**

**Livrable 5: publier une Data complète (multi-types) dans une page (métadonnées, miniatures, liens)**

**Livrable 6: créer une page présentant une collection: métadonnées (éventuelles) et liste des données. Pour chaque donnée, faire une page présentant son contenu.**

**Livrable 7 : graphique analysant les données des collections**

Pour l'instant je suis au livrable 4 A, je vais bientôt passer au 4B dans l'attente de la validation par mon tuteur.

Jeudi 8 Juin

# Matin

## Début de la recherche sur la récupération d'un fichier audio

## Après-midi

Consultation de la documentation de quarto dans la continuité de ma recherche de ce matin.

Pour la récupération je dois utiliser mon code de base en changeant simplement le DOI (DataIdentifier) et de refaire une sélection comme fait pour le code de l'image.

Voici le code de récupération:

```
import nakalapycon
import json
import NklTarget as nklt
import nklAPI_Search as nkls
import nklAPI_Datas as nkld

dataIdentifier = "10.34847/nkl.be5b029k"
targetProd = nklt.NklTarget(isNakalaProd=True, apiKey = "")

rs = nkld.get_datas(targetProd, dataIdentifier)
print(rs)

print("-----")

r = nkld.get_datas_metadata(targetProd, dataIdentifier)

print(r)
```

Voilà ce que cela renvoie:

## Amélioration du code de récupération :

```
import nakalapycon
import json
import NklTarget as nklt
import nklAPI_Search as nkls
import nklAPI_Datas as nkld

dataIdentifier = "10.34847/nkl.be5b029k"
targetProd = nklt.NklTarget(isNakalaProd=True, apiKey = "")

rs = nkld.get_datas(targetProd, dataIdentifier)
#print(rs)

listecollect = (rs.dictVals['collectionsIds'])

print("----- Liste collections:")

for d in range (len(listecollect)):
    print(listecollect[d])

print("----- Liste fichiers")
listefich = (rs.dictVals['files'])

for c in range (len(listefich)):
    f = listefich[c]
    print(f['name'], " ", f['description'], " ", f["sha1"])

print("-----")

r = nkld.get_datas_metadata(targetProd, dataIdentifier)
for i in range (len (r.dictVals)):
    print(r.dictVals[i])

#print(r)
```

Voilà ce qu'il renvoie :

Je pense qu'il faut récupérer l'URI et l'incruster dans mon autre code.

## Vendredi 9 Juin

### Matin

Dans la poursuite de ce que j'ai fais hier voici le code ajouter à ma fonction ce matin pour récupérer les données audio:

```
# On peut aussi traiter des fichiers son, si on en a
elif line.find("<div audio>") != -1:
    for i in range(len(sons)):
        Output_Text += line.replace("TitreSon", sons[i]["description"]).replace("URLAudio", baseURI + dataDOI + "/" + sons[i]["SHA1"])
```

qui remplacera dans mon .qmd les champs suivants:

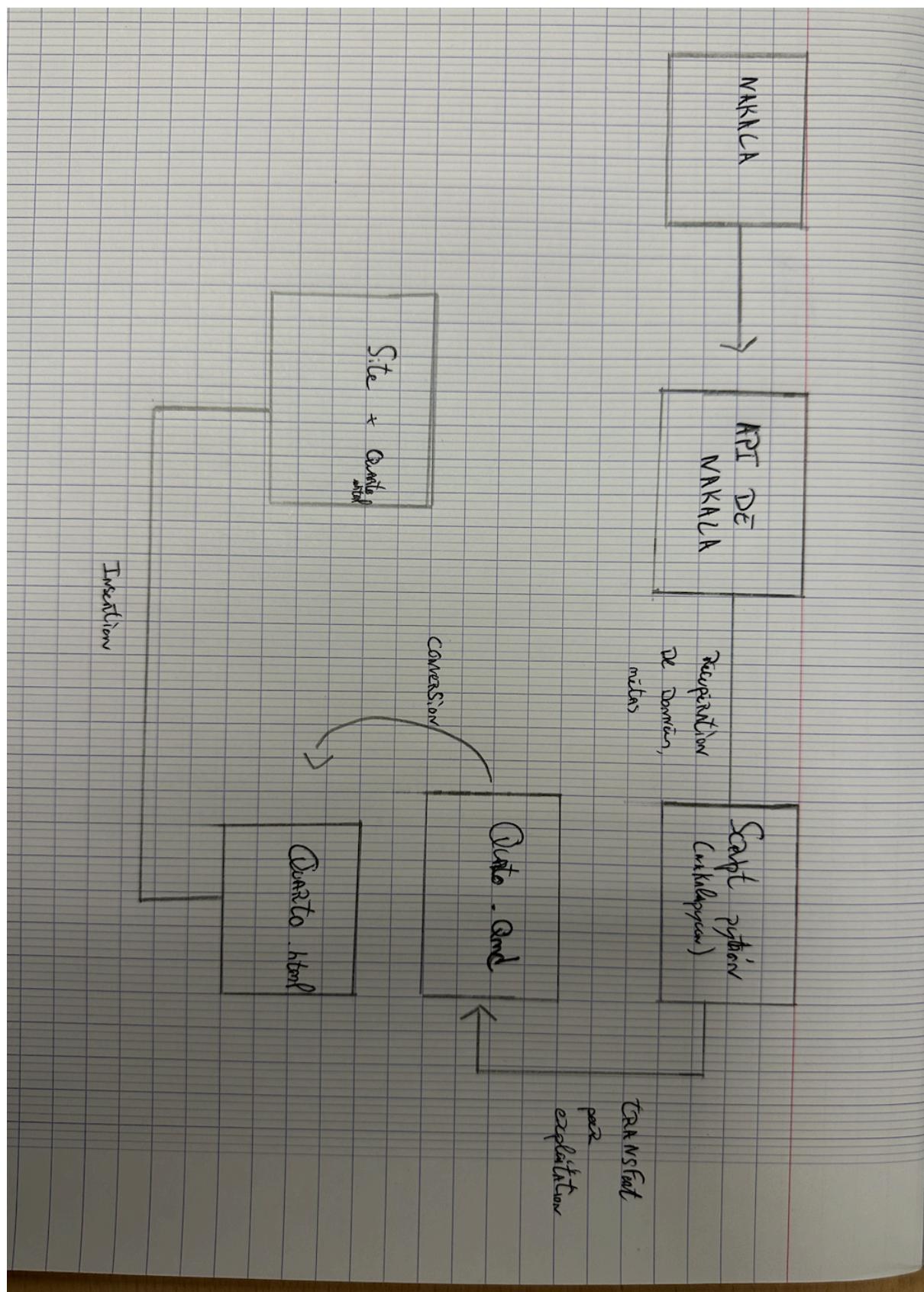
```
<div audio>! [TitreSon] (URLAudio)</div>
```

### Fin de matinée

Avec mon tuteur nous avons discuté des 4 semaines de stages et il m'a donné quelques conseils d'organisation. Pour bien tout saisir nous avons repris à la base , non pas car le code ne fonctionnait pas mais car il souhaité que je comprenne bien tout des réponses renvoyées par L'api de nakala.

### Après midi

À la demande de mon tuteur j'ai effectué un schéma pour représenter le projet que je réalise et une explication de mon code python en algo:



Script du client :

importation de différentes librairies.

création d'une requête de connexion à l'API de WhatsApp.

~~Ensuite~~

Initialisation des deux DOI (Image, Son)

Initialisation de l'API de base.

utilisation de la requête de connexion

Ensuite dans un if :

si la connexion a ~~pas~~ échoué

↳ message d'erreur

↳ déconnexion brisée

Affiliation :

dataFiles est égale au résultat de Files dans RL.DictVals

dataMetas est égale au résultat de Metas dans RL.DictVals

Initialisation de OutputText.

Fonction getmetasByProperty

À l'aide d'un paramètre rentrant pour l'utilisation (prop) et d'un champ lui aussi rentrant pour l'utilisation (field), ce programme va introduire une liste à l'aide de val.

Écrire à l'aide d'une structure itérative (for) :

↳ recherche de la validité du prop

↳ si Field ≠ ""

↳ recherche à l'aide du field →  $\rightarrow$  <sup>Affiliation</sup>  $\rightarrow$   $\rightarrow$  Generator

↳ affiliation  $\rightarrow$  VK

Quatrième du fichier About-And en lecture

Pas de ligne dans F:

Si une ligne ("le Titre") est trouvée :

↳ remplacement de ("le Titre") à l'aide de la fonction

Sinon Si une ligne ("Nom de l'artiste")

↳ remplacement de ("Nom de l'artiste") à l'aide de la fonction

↳ Beaucoup si il y a plusieurs artistes

Sinon Si une ligne ("Nom du Chanteur")

↳ Remplacement à l'aide de la fonction

↳ Beaucoup si il y en a plusieurs

Sinon Si une ligne ("le Dei")

↳ Remplacement ("le Dei") à l'aide de la fonction

Sinon Si une ligne ("Collection ID")

↳ Remplacement grâce à la fonction

↳ Beaucoup si il y en a plusieurs

Sinon Si une ligne ("<div class="img"></div>")

↳ Beaucoup si plusieurs images.

↳ Si deux files ligne ("image")

↳ Remplacement de ("URLImage") par un autre URL  
avec des paramètres d'affichage.

Sinon Si une ligne ("<div class="video">")

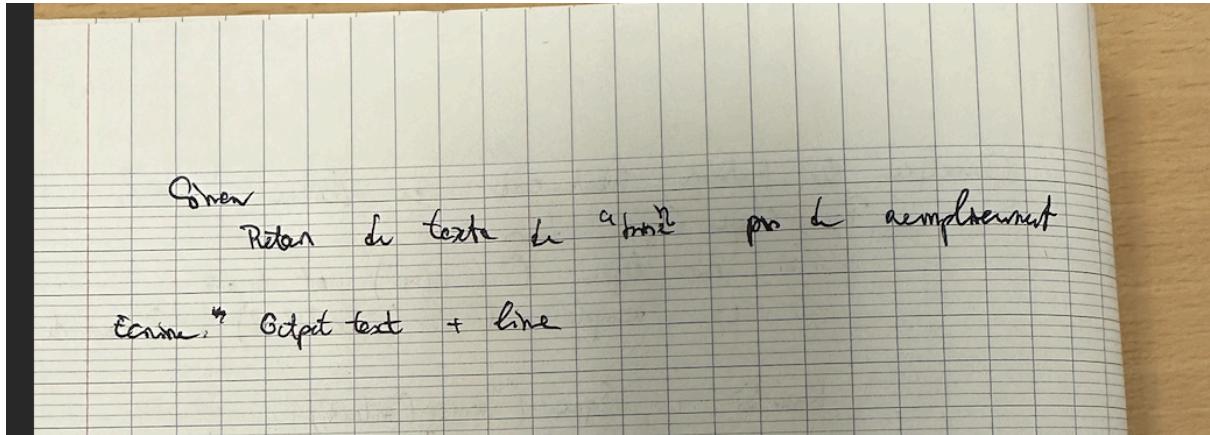
↳ Beaucoup si plusieurs vidéos

↳ Si deux files ligne ("video")

↳ Remplacement ("URLVideo") pour le vrai URL

Sinon Si une ligne ("<div class="photo">")

↳ Remplacement à l'aide de la fonction



## Lundi 12 Juin

### Matin

Reprise sur mon code python, amélioration pour qu'il soit fonctionnel dans toutes les situations donc changement dans le code et dans le .qmd :

```

def getMetasByProperty(prop, field):
    """
    Parameters
    prop : String
        PropertyURI
    field : String
        subfield we want the value.
    Returns
    List of String.
    """
    val = []
    for i in range(len(dataMetas)):
        if dataMetas[i]["propertyUri"] == prop:
            if field == "value":
                val.append(dataMetas[i]["value"][field])
            else:
                val.append(dataMetas[i]["value"])
    return val

with open("./about.qmd", "r", encoding="utf-8") as f:
    for line in f:
        if line.find("LeTitre") == -1:
            Output_Text += line.replace("LeTitre", getMetasByProperty('http://nakala.fr/terms#title', '')[0])
        elif line.find("NomDeAuteur") == -1:
            Auteurs = getMetasByProperty('http://nakala.fr/terms#creator', 'fullName')
            for i in range(len(Auteurs)):
                Output_Text += line.replace("NomDeAuteur", Auteurs[i])
        elif line.find("NomDuContributeur") == -1:
            Contrib = getMetasByProperty('http://purl.org/dc/terms/contributor', '')
            for i in range(len(Contrib)):
                Output_Text += line.replace("NomDuContributeur", Contrib[i])
        elif line.find("LEDOI") == -1:
            Output_Text += line.replace("LEDOI", dataDOI)
        elif line.find("CollectionId") == -1:
            for i in range(len(rc.dictVals["collectionIds"])):
                Output_Text += line.replace("CollectionId", rc.dictVals["collectionIds"][i])
        elif line.find('<div class="image">') == -1:
            for i in range(len(dataFiles)):
                if dataFiles[i]["mime_type"].find("image") != -1:
                    Output_Text += line.replace("TitreImage", dataFiles[i]["description"]).replace("URLImage", baseURI + "iiif/" + dataDOI + "/" + dataFiles[i]["sha1"] + "/full/200,200/o") # /default.jpg
        elif line.find('<div class="audio">') == -1:
            for i in range(len(dataFiles)):
                if dataFiles[i]["mime_type"].find("audio") != -1:
                    Output_Text += line.replace("TitreSon", dataFiles[i]["description"]).replace("URLAudio", baseURI + dataDOI + "/" + dataFiles[i]["sha1"] ) # + "+" + dataFiles[i]["extension"]
        elif line.find('<div classe="footer">') == -1:
            Output_Text += line.replace("TheDepositor", rc.dictVals["depositor"]["name"]).replace("DepositDate", rc.dictVals["creDate"]).replace("TheLicence", getMetasByProperty('http://nakala.fr/terms#license', "")[0])
        else:
            Output_Text += line
    
```

Le code fonctionne mais tous les champs ne sont pas remplacés, je dois faire des vérifications dans l'API search .

```
1 ---  
2 title: "LeTitre"  
3 ---  
4  
5 # Metadata  
6 datedecreation  
7  
8 ## Auteur(trice)  
9 NomDeL'auteur  
0  
1 ## Contributeur(trice)  
2 NomContributeur  
3  
4 ##### DOI : LEDOI  
5  
6  
7 ### Appartient aux collections:  
8 > CollectionId'|<br><br>...  
0  
1 # Files  
2 <div audio>! [TitreSon] (URLAudio)</div>  
3  
4 <div class="image">  
5 ! [TitreImage] (URLImage)  
6 </div>  
7  
8 <div footer>TheDepositor TheLicence</div>
```

## Après midi

Poursuite sur l'amélioration de mon code , erreur de data non récupérer régler.  
Début de la recherche sur la récupération de fichier audio.

## Mardi 13 juin

Maladie

## Mercredi 14 juin

Maladie

## Jeudi 15 juin

### Matin

Reprendre sur la récupération des fichiers audio : recherche dans l'api de nakala à l'ai toujours d'un DOI.

Voila ma fonction de récupération d'audio:

```
if line.find('<div class="audio">') != -1:
    for i in range(len(dataFiles)):
        if dataFiles[i]['mime_type'].find("audio") != -1:
            Output_Text += line.replace("TitreSon", dataFiles[i]["description"]).replace("URLAudio", baseURI + "/" + dataFiles[i]["sha1"]) # +"." + dataFiles[i]["extension"]
            #audio_element = f'{i}!{dataFiles[i]["description"]}{URLAudio}'.replace("URLAudio", f'{baseURI}{dataDOI}{dataFiles[i]["sha1"]}')
            #Output_Text += line.replace(f'{i}!{TitreSon}{URLAudio}', audio_element)
```

Et voici ce que cela renvoie dans un terminal :

```
<div class="audio">[On entend Hilaire Bonnin qui chante l'air, mais l'air est difficile à déterminer. Lorsque Marie Couillaud se joint à lui, l'air est clair. Lorsqu'elle a fini, Hilaire et Marie font comprendre que l'air se joue deux fois complètes. On entend la voix de Jacques Bonnin (membre des Gens de Cherves).](https://api.nakala.fr/10.34847/nkl.b7bd3k55/2f6af2ad54a46990e38aa76a2647d8ce7884c0b)</div>
<div class="audio">[Marie Couillaud chante l'air, elle précise à la fin qu'il faut le faire deux fois.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/7e03c963f0a5fabb432a9e731a0d2377a895ff) </div>
<div class="audio">[Les informateurs discutent à propos de la manière de danser les quadrilles.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/d32f2c1ed304c567ef0145bd32ebbfde9abc1a)</div>
<div class="audio">[Suite à la demande de l'enquêteuse, Marie Couillaud chante l'air de la berline, elle précise que ça se dansait avec un pas de polka avant de reprendre l'air une deuxième fois.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/42bba9e09c3c)</div>
<div class="audio">[Plusieurs témoins et des gens de Cherves dansent la danse à l'ours qu'ils chantent en même temps. Marie Couillaud confirme.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/b2ba9e09c3c)</div>
<div class="audio">[Marie Couillaud chante l'air de Jean Brisquet et avec les autres témoins, elle explique la danse. Les explications sont partielles à cause de coupe de la bande et du brouhaha par moment mais aussi car les témoins ont des souvenirs manquants.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/4881d2ef0d6fb5f0835abecabc7ce4eecc2dbb9703)</div>
<div class="audio">[Marie Couillaud et Simone Bonnin chantent l'air puis les témoins discutent de la danse par bribes. On devine que Simone danse.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/7dec96bf)</div>
<div class="audio">[L'enquêteur commence avec Martine Roy qui essaie de reproduire un air qu'elle a entendu. Marie Couillaud chante ensuite le galop du quadrille. Il n'y a que des fragments, il est difficile d'en comprendre la mélodie en entier.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/252435698c583c3e2f4bc4f58a1096782e9614c9)</div>
<div class="audio">[Marie Couillaud chante l'air de la gigouillette au tralala. La bande contient quelques bruits parasites mais la mélodie est intelligible.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/f5fa5315e56095ceeb9d3301a28edc68a0b6d46590f)</div>
<div class="audio">[Marie Couillaud et Simone Bonnin chantent l'air au tralala sur la première partie et avec des paroles sur la deuxième. A la fin, elles parlent d'une scottish qu'elles n'arrivent pas à reconnaître.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/590dd802d75ccb1bfecc79a6a819632065fa1dd)</div>
<div class="audio">[Marie Couillaud chante l'air d'une apparition des témoins la dansent en mazurka, alors elle précise à la fin qu'il s'agit d'une valse vienne.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/d5172eaca2df509fcdf192fd355cae2b0073b78e)</div>
<div class="audio">[Marie Couillaud chante au tralala l'air d'une mazurka mais la bande coupe, il manque la fin.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/dd572a7ac53bbc9acebec9f6178302ee5d4f)</div>
<div class="audio">[Marie Couillaud explique comment se danse la gatinelle et dans quel contexte elle était dansée.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/873af9b8dcccbeffd01a9b5a5ce83c1b8202116)</div>
```

et voici dans mon fichier en .qmd :

```
scottich qu'elles n'arrivent pas à retrouver.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/c500d802b75cb1b6fecc79a6a819632065fac1dd)</div>
<div>
<div class="audio">![Marie Couillaud chante l'air, apparemment des témoins la dansent en mazurka, alors elle précise à la fin qu'il s'agit d'une valse vienne.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/d5172eaca2df509fcdf192fd355cae2b0073b78e)</div>
<div>
<div class="audio">![Marie Couillaud chante au tralala l'air d'une mazurka mais la bande coupe, il manque la fin.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/dd572a7a6c53bbc9acebe6c9f6178302ee5d4f)</div>
<div>
<div class="audio">![Marie Couillaud explique comment se danse la gatinelle et dans quel contexte elle était dansée.](https://api.nakala.fr/10.34847/nkl.b7bd3k55/873af9b8dcccbeffd01a9b5a5ce83c1b8202116)</div>
```

Proposition_de...	Home	LeTitre	Veillée de danse autour de Marie Couillaud et Camille Bonnin - Maisonneuve 11.1971 [FR]
Roy, Martine			
DOI : 10.34847/nkl.b7bd3k55			
<b>Appartient aux collections:</b>	10.34847/nkl.2ef6zbpb		
	10.34847/nkl.1fa6pop7		
<b>Fichiers:</b>	 On this page <ul style="list-style-type: none"> <li>Metadata</li> <li>Auteur(trice)</li> <li>Contributeur(trice)</li> <li><b>Appartient aux collections:</b></li> <li>Fichiers:</li> </ul>		
	 Marie Couillaud chante l'air, elle précise à la fin qu'il faut le faire deux fois.		
	 Les informateurs discutent à propos de la manière de danser les quadrilles.		
	 Suite à la demande de l'enquêteuse, Marie Couillaud chante l'air de la berline, elle précise que ça se dansait avec un pas de polka avant de reprendre l'air une deuxième fois.		
	 Plusieurs témoins et des gens de Cherves dansent la danse à l'ours qu'ils chantent en même temps. Marie Couillaud confirme.		
	 Marie Couillaud chante l'air de Jean Brisquet et avec les autres témoins, elle explique la danse. Les explications sont partielles à cause de coupe de la bande et du brouhaha par moment mais aussi car les témoins ont des souvenirs manquants.		

Je pense que je dois ajouter quelque chose comme une extension pour que cela me renvoie un fichier audio avec une interface et un player .

En changeant le type de balise voila ce que cela donne :

```
<div class="audio"><audio controls><source src="https://api.nakala.fr/10.34847/nkl.b7bd3k55/fe0a73117f7a752ee95568e6b2ccb3df88e3e760" type="audio/mpeg">Your browser does not support the audio element.</audio></div><div class="audio"><audio controls><source src="https://api.nakala.fr/10.34847/nkl.b7bd3k55/bd6bd4654c6a8bcbfd17732244753cc540ee0c9f" type="audio/mpeg">Your browser does not support the audio element.</audio></div><div class="audio"><audio controls><source src="https://api.nakala.fr/10.34847/nkl.b7bd3k55/2133003a5e63c94a97b3aa15777af8144972bd97">
```

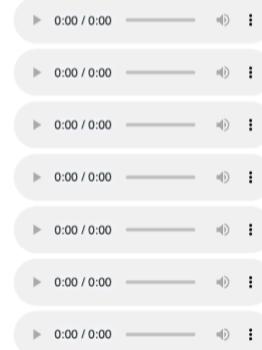


Appartient aux collections:

10.34847/nkl.2ef6zbbp

10.34847/nkl.3fa6pop7

Fichiers:



en revanche les audios ne sont pas accessibles.

Je repars de la balise de base car trop compliqué et cela ne fonctionne pas

## Après-midi

Petit débriefing avec un collègue de la plateforme, il m'a aidé avec mes soucis de récupération d'audio et m'a donné des conseils pour mes méthodes de travail et d'organisations. Voila mon code :

```
elif line.find('<div class="audio">') != -1:  
    for i in range(len(dataFiles)):  
        if dataFiles[i]['mime_type'].find("audio") != -1:  
            AudioBloc = ""  
            AudioBloc = line.replace("TitreSon", dataFiles[i]["description"])  
            AudioBloc = AudioBloc.replace("URLAudio", baseURI + "data/" + dataDOI + "/" + dataFiles[i]["sha1"])  
            Output_Text += AudioBloc
```

et voici ce que cela renvoie :

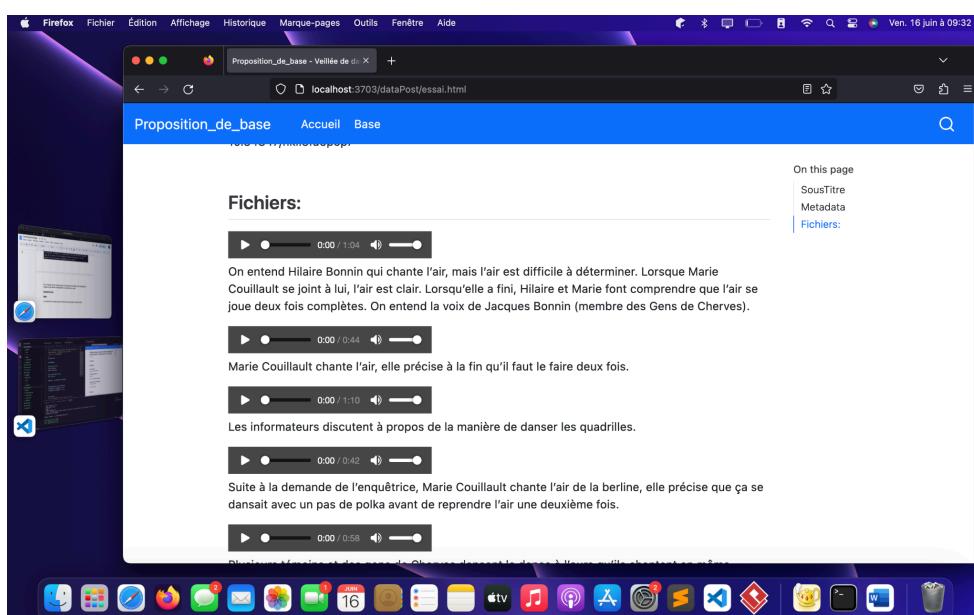
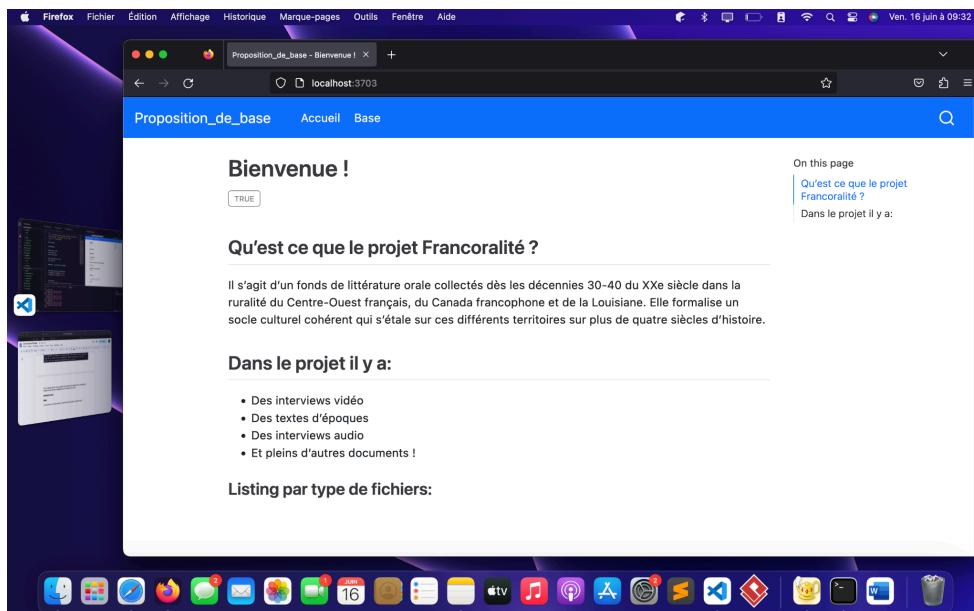
```
<div class="audio"> <audio controls> <source src="https://api.nakala.fr/data/10.34847/nkl.b7bd3k55/4881d2efd0b6f5b035abecabc7ce4eec2dbb9703" type="audio/mpeg"> <p>Marie Couillault chante l'air de Jean Brisquet et avec les autres témoins, elle explique la danse. Les explications sont partielles à cause de coupe de la bande et du brouhaha par moment mais aussi car les témoins ont des souvenirs manquants.</p> </audio>
```

En fin d'après midi j'ai mis au propre mes dossiers, donné des noms cohérents et simples à mes fichiers. Réorganiser mon index et mon .yaml

## **Vendredi 16 Juin**

### **Matin**

J'ai continué sur l'aspect visuel, rendu le site plus propre et ergonomique:



J'ai rajouter un type à la data qui peut être audio, photo, texte ou vidéo voici le code :

```
listeTypeFich = (rc.dictVals['files'])  
  
for c in range (len(listeTypeFich)):  
    TabType = []  
    f = listeTypeFich[c]  
    TabType = (f["mime_type"])  
  
  
    elif \line.find("TypeData") != -1:  
        Output_Text += line.replace("TypeData", TabType)
```

et voici ce que cela renvoie :

AUDIO/MPEG

Maintenant j'aimerais faire un listing des types de données disponibles sur mon index. De sorte que si un utilisateur veut consulter un type de données il puisse en choisissant simplement audio, vidéo, texte ou photo.

Voici ce que cela donne :

### Categories

- All (3)
- TypeData (1)
- audio/mpeg (1)
- image/jpeg (1)

et le listing de page d'accueil :

**Photo de terrain - Mission  
Saintonge 1969 - Enquête  
UPOI\_ATP\_0001\_0012**

IMAGE/JPEG

Claudie Marcel-Dubois



**Veillée de danse autour de  
Marie Couillault et Camille  
Bonnin - Maisonneuve 11.1971  
[FR]**

AUDIO/MPEG

Camille Bonnin Marie Couillault

La barre de recherche réactive :

The screenshot shows a search interface with a blue header bar. In the center, there is a search bar containing the text "veillé de danse". Below the search bar, a dropdown menu displays four matching documents. The first document is highlighted in blue and shows the title "Veillée de danse autour de Marie Couillault et Camille Bonnin - Maisonneuve 11.1971 [FR]" and the names "Camille Bonnin Marie Couillault Roy, Martine 10.34847/nkl.2ef6zbbp 10.34847/nkl.3fa6pop7". To the right of the search results, there is a sidebar with various filters and search options.

Document Title	Names	DOI / URL
Veillée de danse autour de Marie Couillault et Camille Bonnin - Maisonneuve 11.1971 [FR]	Camille Bonnin Marie Couillault Roy, Martine	10.34847/nkl.2ef6zbbp 10.34847/nkl.3fa6pop7
Veillée de danse autour de Marie Couillault et Camille Bonnin - Maisonneuve 11.1971 [FR]	Camille Bonnin Marie Couillault Roy, Martine	10.34847/nkl.2ef6zbbp 10.34847/nkl.3fa6pop7
Veillée de danse autour de Marie Couillault et Camille Bonnin - Maisonneuve 11.1971 [FR]	Camille Bonnin Marie Couillault Roy, Martine	10.34847/nkl.2ef6zbbp 10.34847/nkl.3fa6pop7
Veillée de danse autour de Marie Couillault et Camille Bonnin - Maisonneuve 11.1971 [FR]	Camille Bonnin Marie Couillault Roy, Martine	10.34847/nkl.2ef6zbbp 10.34847/nkl.3fa6pop7

Après midi:

En fin de matinée j'ai vu mon tuteur qui m'a dit de revoir mon code pour qu'il soit exploitable sous forme de deux fonctions:

La première grâce à un DOI entrer en paramètre va faire les changements à l'aide d'un documents utilise comme template nommé "base.qmd"

```
def genDataPages(dataDOI):
    """
    Parameters
    dataDOI: String
    """
    baseURI = "https://api.nakala.fr/"

    if rc.isSuccess == True:
        Output_Text = ""
        dataFiles = rc.dictVals["files"]
        dataMetas = rc.dictVals["metas"]

        with open("./base.qmd", "r", encoding="utf-8") as f:
            for line in f:
                if line.find("LeTitre") != -1:
                    Output_Text += line.replace("LeTitre", getMetasByProperty('http://nakala.fr/terms#title', '')[0])
                    print(Output_Text, "1")

                elif line.find("TypeData") != -1:
                    listeTypeFich = [rc.dictVals['files']]
                    for c in range(len(listeTypeFich)):
                        TabType = []
                        f = (listeTypeFich[c])
                        TabType = [(f['mime_type'])]
                        Output_Text += line.replace("TypeData", TabType)
                        print(Output_Text, "2")

                elif line.find("NomDeL'auteur") != -1:
                    Auteurs = getMetasByProperty('http://nakala.fr/terms#creator', 'fullName')
                    for i in range(len(Auteurs)):
                        Output_Text += line.replace("NomDeL'auteur", Auteurs[i])
                    print(Output_Text, "3")

                elif line.find("NomContributeur") != -1:
                    Contrib = getMetasByProperty('http://purl.org/dc/terms/contributor', '')
                    for i in range(len(Contrib)):
                        Output_Text += line.replace("NomContributeur", "\n" + Contrib[i])
                    print(Output_Text, "4")

                elif line.find("LEDOI") != -1:
                    Output_Text += line.replace("LEDOI", dataDOI)
                    print(Output_Text, "5")

                elif line.find("CollectionId") != -1:
                    for i in range(len(rc.dictVals["collectionIds"])):
                        Output_Text += line.replace("CollectionId", rc.dictVals["collectionIds"][i])
                    print(Output_Text, "6")

                elif line.find('<div class="image">') != -1:
                    for i in range(len(dataFiles)):
                        if dataFiles[i]['mime_type'].find("image") != -1:
                            Output_Text += line.replace("TitreImage", dataFiles[i]['description']).replace("URLImage", baseURI + "iiif/" + dataDOI + "/" + dataFiles[i]['sha1'] + "/full/300/0/default.jpg")
                    print(Output_Text, "7")

                elif line.find('<div class="audio">') != -1:
                    for i in range(len(dataFiles)):
                        if dataFiles[i]['mime_type'].find("audio") != -1:
                            AudioBloc = ""
                            AudioBloc = line.replace("TitreSon", dataFiles[i]['description'])
                            AudioBloc = AudioBloc.replace("URLAudio", baseURI + "data/" + dataDOI + "/" + dataFiles[i]['sha1'])
                            Output_Text += AudioBloc
                    print(Output_Text, "8")

                elif line.find('<div class="footer">') != -1:
                    Output_Text += line.replace("TheDepositor", rc.dictVals["depositor"]["name"]).replace("DepositDate", rc.dictVals["creaDate"]).replace("TheLicence", getMetasByProperty('http://nakala.fr/terms#license', '')[0])
                    print(Output_Text, "9")

                else:
                    Output_Text += line

        #Écriture du contenu modifié dans un nouveau fichier .qmd
        modified_page_path = pathlib.Path('./env_quarto/dataPost/essai2.qmd')
        modified_page_path.write_text(Output_Text)

    else:
        print("Data non récupérée")
```

La deuxième fonction va chercher une métadonnée dans L'API search de Nakala à l'aide d'un paramètre en entrée

```
def getMetasByProperty(prop, field):
    """
    Parameters
    -----
    prop : String
        PropertyURI
    field : String
        subfield we want the value.

    Returns
    -----
    List of String.
    """
    dataFiles = rc.dictVals["files"]
    dataMetas = rc.dictVals["metas"]
    val = []

    for i in range(len(dataMetas)):
        if dataMetas[i]["propertyUri"] == prop:
            if field != "":
                val.append(dataMetas[i]["value"][field])
            else:
                val.append(dataMetas[i]["value"])
    return val
```

Maintenant il faut que j'effectue les changements pour que lorsque le fichier se créer le nom du fichier soit : DOI.qmd

```
NomFich = dataDOI.replace(".", "_").replace("/", "-")

#Écriture du contenu modifié dans un nouveau fichier .qmd
modified_page_path = pathlib.Path('..env_quarto/dataPost/', NomFich + '.qmd')
modified_page_path.write_text(Output_Text)
```

remplacement des . et des / sinon cela n'est pas fonctionnel.

le dernier objectif était de faire une fonction récupérant tous les DOIS présent sur nakala et présent dans une collection certifiée voici mon code qui est non fonctionnel :

```
import NklTarget as nklT
import nklAPI_Collections as nklC
import nklAPI_Datas as nklD
import pathlib

def get_collections_datas_test(collectionIdentifier):
    nklT_prodEmpty = nklT.NklTarget(isNakalaProd=True, apiKey="")

    dataIdentifier = collectionIdentifier
    page = 1
    allDOIs = []

    while True:
        print("Récupération des données de la page", page)
        response = nklC.get_collections_datas(nklT_prodEmpty, dataIdentifier, page=page)

        if not response.isSuccess:
            print("Une erreur s'est produite lors de la récupération des données.")
            break

        data = response.dictVals['data']
        for item in data:
            doi = item.get('doi')
            if doi:
                allDOIs.append(doi)

        lastPage = response.dictVals['lastPage']
        if page >= lastPage:
            break

        page += 1

    return allDOIs

collectionIdentifier = "10.34847/nkl.ffabdg38"
DOIS = get_collections_datas_test(collectionIdentifier)

print("Liste des DOIS de la collection :")
for doi in DOIS:
    print(doi)
```

## Mardi 20 juin

### Soir

j'ai créé une fonction nommée get\_all\_dois qui récupère tous les dois présents dans une collection.

Voici son code :

```
def get_all_dois(collection_identifier):
    nklT_prodEmpty = nklT.NklTarget(isNakalaProd=True, apiKey="")

    page = 1
    all_dois = []

    while True:
        print("Récupération des données de la page", page)
        response = nklC.get_collections_datas(nklT_prodEmpty, collection_identifier, page=page)

        if not response.isSuccess:
            print("Une erreur s'est produite lors de la récupération des données.")
            break

        data = response.dictVals['data']
        for item in data:
            doi = item.get('identifier')
            if doi:
                all_dois.append(doi)

        last_page = response.dictVals['lastPage']
        if page >= last_page:
            break

        page += 1

    return all_dois

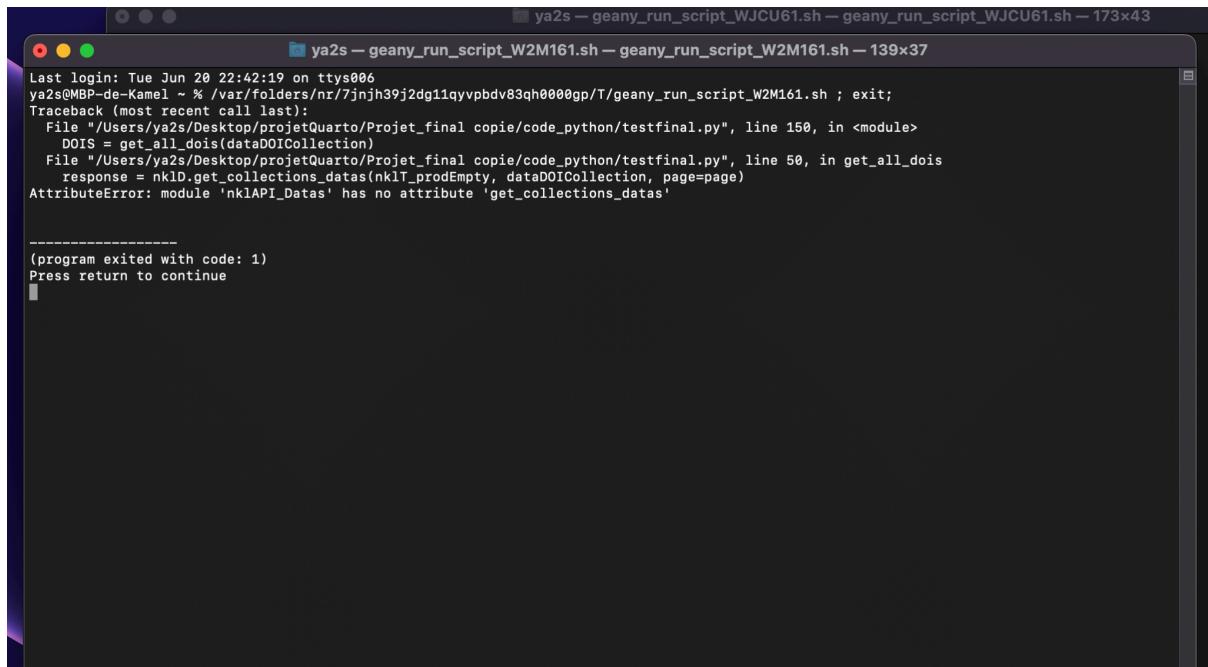
collection_identifier = "10.34847/nkl.3769u06s"
dois = get_all_dois(collection_identifier)

print("Liste des DOIs de la collection :")
for doi in dois:
    print(doi)
```

la fonction renvoie :

```
Liste des DOIs de la collection :
10.34847/nkl.abd8xc33
10.34847/nkl.bcd4021
10.34847/nkl.0fd4d61jx
10.34847/nkl.bcc09rhc
10.34847/nkl.0e4e5khz
10.34847/nkl.b02bb1wz
10.34847/nkl.d1f2es8d
10.34847/nkl.2c157jx0
10.34847/nkl.469d884m
10.34847/nkl.7c786wx
10.34847/nkl.a66e8nq3
10.34847/nkl.bea1i8dz
10.34847/nkl.0ceb2108
10.34847/nkl.947f98pv
10.34847/nkl.b7cb6ah
10.34847/nkl.a3647e42
10.34847/nkl.95581216
10.34847/nkl.93eb60hk
10.34847/nkl.e50fh84n
10.34847/nkl.c5ccu1i6
10.34847/nkl.d4430b01
10.34847/nkl.ec0333t1
10.34847/nkl.b32f41g8
10.34847/nkl.ce8abjk1
10.34847/nkl.6e67q36t
10.34847/nkl.bcf31m97
```

Le problème que je rencontre est lorsque je la rajoute dans mon autre fichier de code avec toute mes autres fonctions et voici l'erreur que cela me mets :



The screenshot shows a terminal window titled "ya2s — geany\_run\_script\_W2M161.sh — geany\_run\_script\_W2M161.sh — 173x43". The window contains the following text:

```
Last login: Tue Jun 20 22:42:19 on ttys006
ya2s@MBP-de-Kamel ~ % /var/folders/nr/7jnjh39j2dg11qyvpbdv83qh0000gp/T/geany_run_script_W2M161.sh ; exit;
Traceback (most recent call last):
  File "/Users/ya2s/Desktop/projetQuarto/Projet_final copie/code_python/testfinal.py", line 150, in <module>
    DOIS = get_all_dois(dataDOICollection)
  File "/Users/ya2s/Desktop/projetQuarto/Projet_final copie/code_python/testfinal.py", line 50, in get_all_dois
    response = nkID.get_collections_datas(nkIT_prodEmpty, dataDOICollection, page=page)
AttributeError: module 'nk1API_Datas' has no attribute 'get_collections_datas'

-----
(program exited with code: 1)
Press return to continue
```