

Les étapes de la création de l'application avec le framework Django.

-> Créer un dépôt sur GitHub : « Send_messages_on_social_networks »

-> Initialiser un nouveau projet et le lier au **dépôt GitHub**.

-> Installer l'**environnement virtuel**.

1. pip3 install virtualenv (initialiser)
2. virtualenv -p python3 env (créer le dossier dans le projet)
3. source env/bin/activate (activer l'environnement virtuel)
4. deactivate (désactiver l'environnement virtuel)

-> **Installation de django :**

commande => \$ **pip3 install django**

-> **Créer un nouveau projet Django :**

commande => \$ **django-admin startproject send_messages_on_social_networks**

-> **Tester le bon fonctionnement du serveur :**

commande => \$ **python3 manage.py runserver**

-> **Créer une première application dans le projet :**

commande => \$ **python3 manage.py startapp chat**

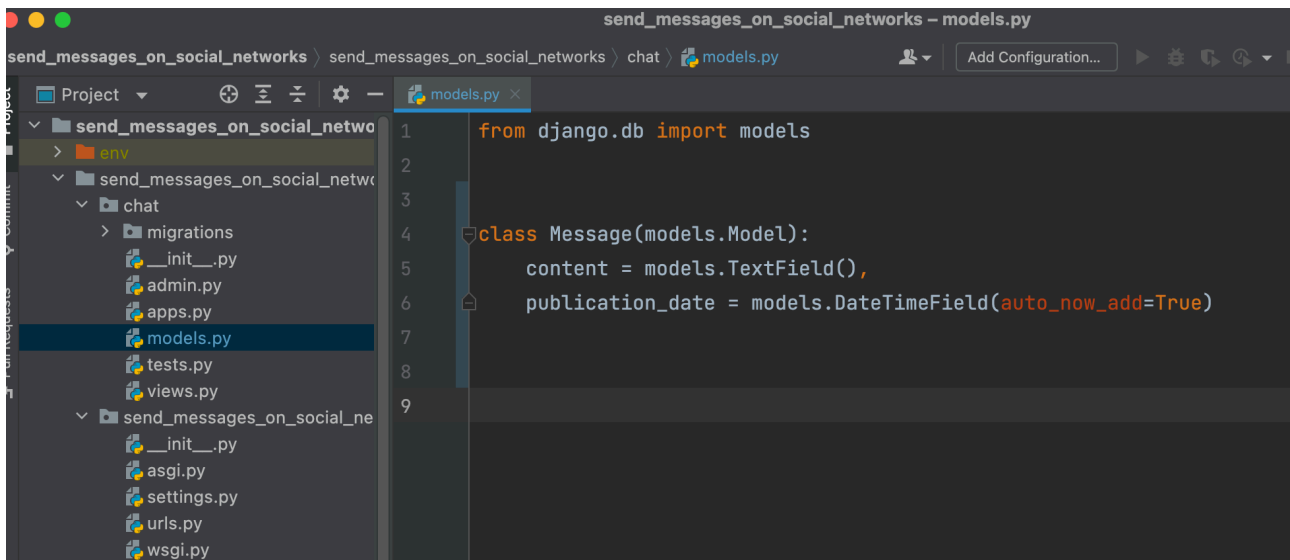
-> **Configuration de l'application (fichier settings.py) :**

- Ajout de l'application « chat » récemment créée aux « INSTALLED_APPS »

- Cacher la clé secrète de l'application pour quelle ne soit pas accessible pour tout le monde sur Github. commande => \$ **pip3 install python-dotenv**

- Configuration de la base de données, je ne vais pas utiliser la base de données par défaut qui est SQLite mais configurer le fichier settings.py pour lier mon application à une BBDR PostgreSQL.

-> Création du premier modèle, et test de la migration dans la base de données :

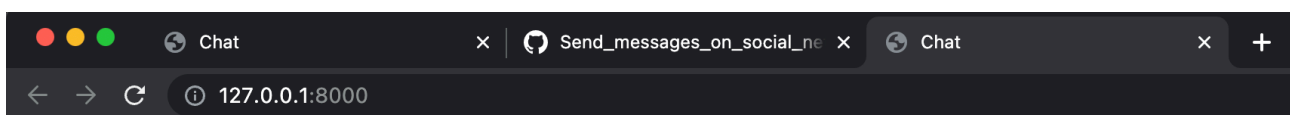


```
1 from django.db import models
2
3
4 class Message(models.Model):
5     content = models.TextField(),
6     publication_date = models.DateTimeField(auto_now_add=True)
7
8
9
```

commande => \$ **python3 manage.py makemigrations**

-> Création des premiers templates qui affichent les données provenant depuis la base de données Postgre SQL.

Data output				
Messages				
Notifications				
	id [PK] bigint	publication_date timestamp with time zone	content text	
1	3	2022-06-27 09:31:51.842...	Contenu message 2	
2	2	2022-06-27 09:28:47.668...	Contenu du message 1	



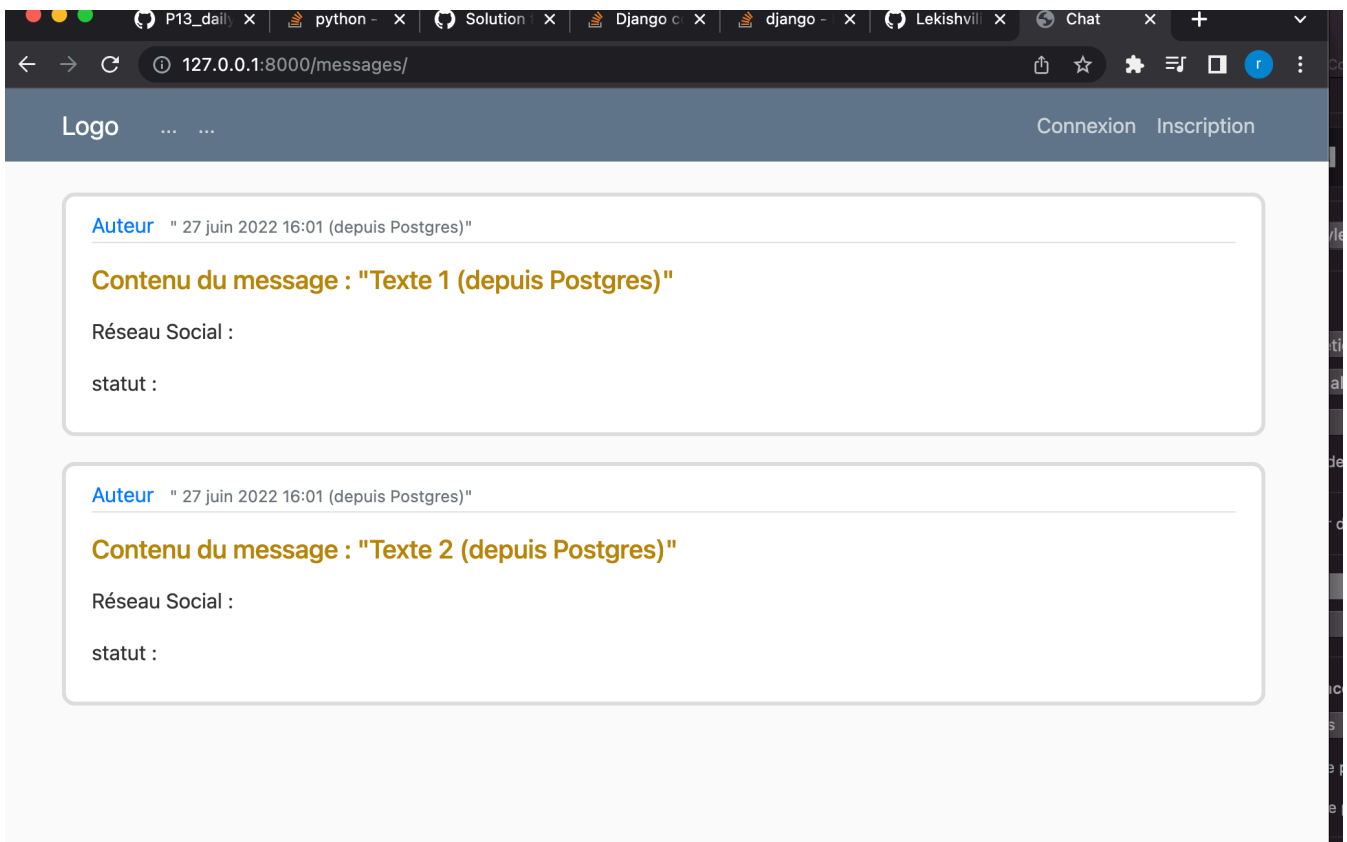
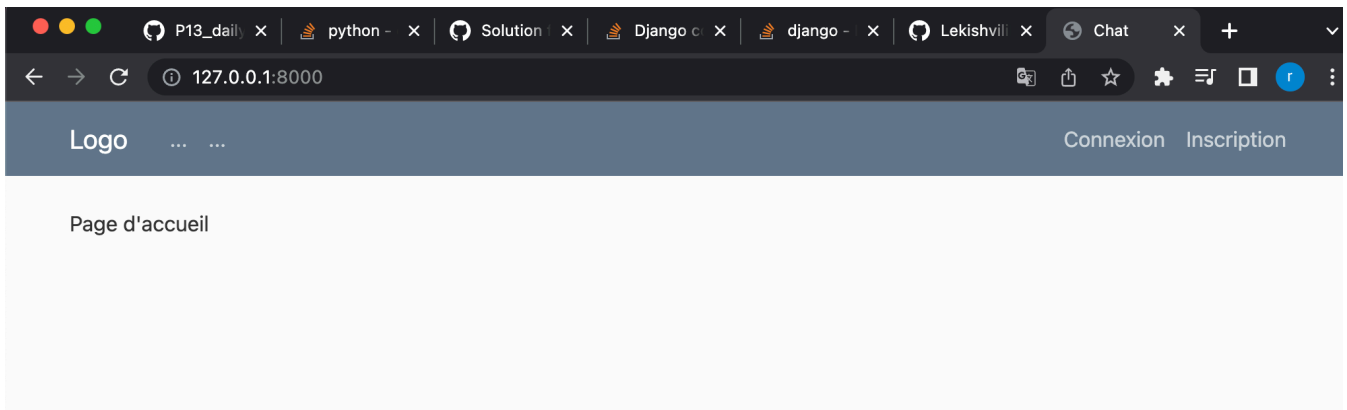
Messages

- Date de publication : 27 juin 2022 09:28
- Contenu : Contenu du message 1
- Date de publication : 27 juin 2022 09:31
- Contenu : Contenu message 2

-> Gestion personnalisée des utilisateurs :

=> Ajout de la classe User dans l'application « users » ce qui va nous permettre de gérer les utilisateurs de manière personnalisée. On utilise pas le modèle proposé par défaut par django pour pouvoir stocker des informations supplémentaires sur chaque utilisateur.

-> Premiers templates et routes de l'application



-> **Création d'un super utilisateur pour accéder à la page admin de l'application.**

commande => \$ **python3 manage.py createsuperuser**

-> **Ajout de la fonctionnalité Inscription / Connexion**

- Nouvelles routes
- Nouveaux templates
- Modification de la bar de navigation une fois que l'utilisateur est connecté.
- Accès à la page messages une fois connecté

NON CONNECTÉ

The screenshot shows the login page of an application. At the top, there is a purple navigation bar with a 'Logo' on the left and 'Connexion' and 'Inscription' on the right. The 'Connexion' and 'Inscription' links are circled in red. Below the navigation bar, the page title is 'Page de connexion'. The main content area contains a 'Connexion' form with two input fields: 'Nom d'utilisateur*' and 'Mot de passe*'. Below these fields is a 'Connexion' button. At the bottom of the form, there is a link 'Pas encore de compte ? Créer un compte'.

CONNECTÉ

The screenshot shows the messages page of an application. At the top, there is a purple navigation bar with a 'Logo' on the left and 'Mes messages' and 'Déconnexion' on the right. The 'Mes messages' and 'Déconnexion' links are circled in red. Below the navigation bar, the page displays two message cards. Each card has a header with the name 'rati' and the timestamp '27 juin 2022 16:01 (depuis Postgres)'. The main content of each card is 'Contenu du message : "Texte 1 (depuis Postgres)"' and 'Contenu du message : "Texte 2 (depuis Postgres)"' respectively. Below the message content, there are labels for 'Réseau Social' and 'statut'.

-> Ajout de la fonctionnalité Déconnexion

- Création du fichier logout.html
- Une fois déconnecté l'utilisateur n'est plus redirigé vers une page admin (par défaut)

-> Création d'un compte test Discord

- Id : lekishvili.rati@gmail.com
- psw : Koesio_project@7400

-> 1er message (manuel) réussi depuis un fichier python de l'application vers un groupe test de Discord.

The screenshot displays a development environment with a code editor, a terminal, and a web browser. The code editor shows a Python script using the `requests` library to send a POST request to the Discord API. The payload is a JSON object with a `content` field set to "depuis fichier python". The terminal shows the command to run the script and the output "Process finished with exit code 0". The web browser shows the Discord interface with a message "depuis fichier python" sent to the "general" channel. The message is attributed to the user "rati".

```
import requests
1
2
3 # https://discord.com/api/v9/channels/991618332895752255
4
5 # Authorization: OTkxMjE1NjI0TEY0TA50DQ0.61G0V...IE...
6
7 payload = {
8     'content': 'depuis fichier python'
9 }
10
11 header = {
12     'Authorization': 'OTkxMjE1NjI0TEY0TA50DQ0.61G0V...'
13 }
14
15 r = requests.post('https://discord.com/api/v9/channels/991618332895752255/messages', headers=header, json=payload)
```

Run: discord_chat - /Users/ratikishvili/Desktop/send_messages_on_social_networks/env/bin/python3 discord_chat.py

Process finished with exit code 0

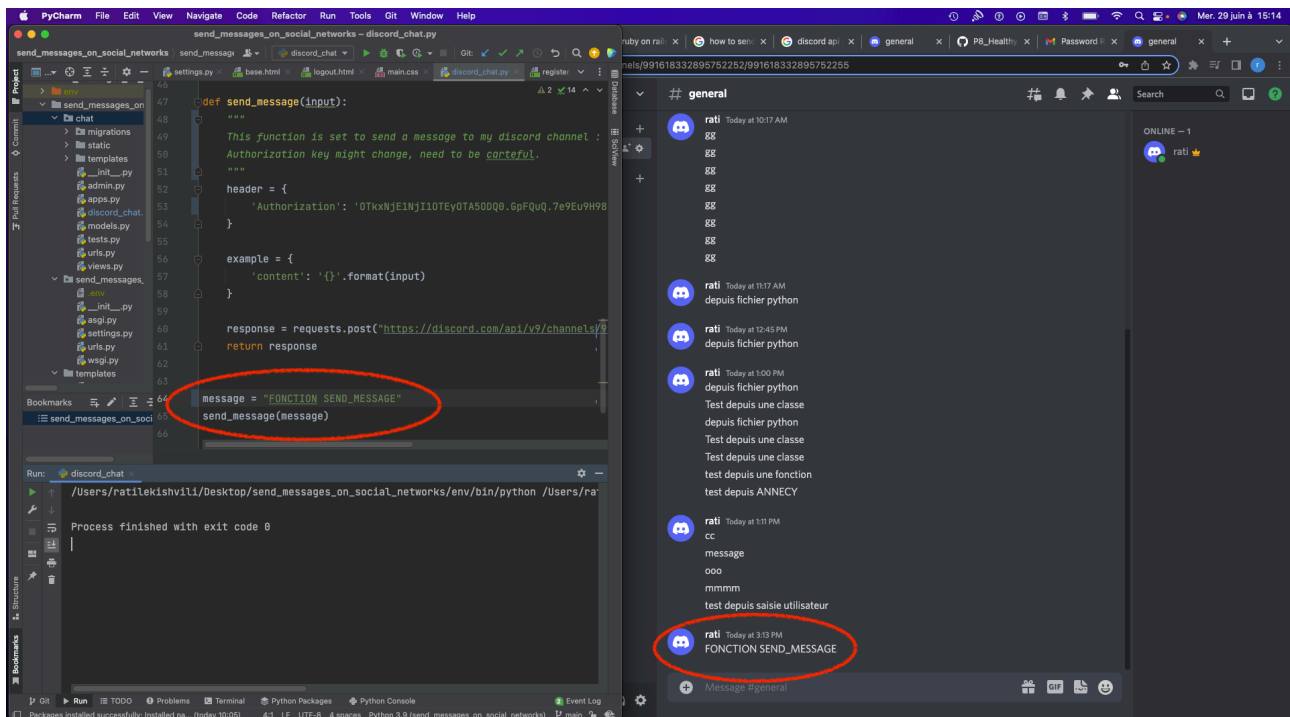
Discord Channel: # general

Message: depuis fichier python

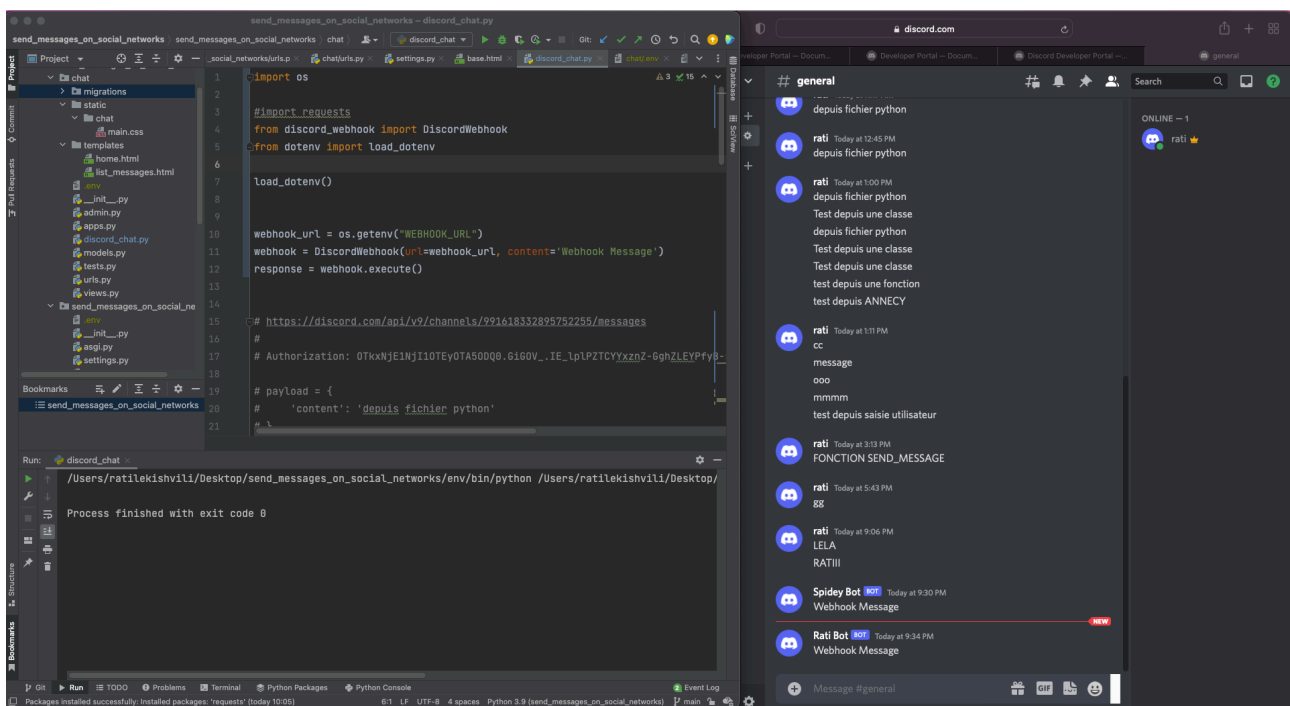
User: rati

Status	URL	Initiator	Total Bytes	Error
failure	https://discord.com/assets/9e5128488810e63b69d4.js.map	https://discord.co...		HTTP error: status code 404, net-ERR_HTTP_RES...
failure	https://discord.com/assets/905782068868524e052.js.map	https://discord.co...		HTTP error: status code 404, net-ERR_HTTP_RES...
failure	https://discord.com/assets/5c2b2d6d9c743e657.js.map	https://discord.co...		HTTP error: status code 404, net-ERR_HTTP_RES...
failure	https://discord.com/assets/49ee1481a096ca3e36.js.map	https://discord.co...		HTTP error: status code 404, net-ERR_HTTP_RES...
failure	https://discord.com/assets/4c03a60e1199768e682.js.map	https://discord.co...		HTTP error: status code 404, net-ERR_HTTP_RES...

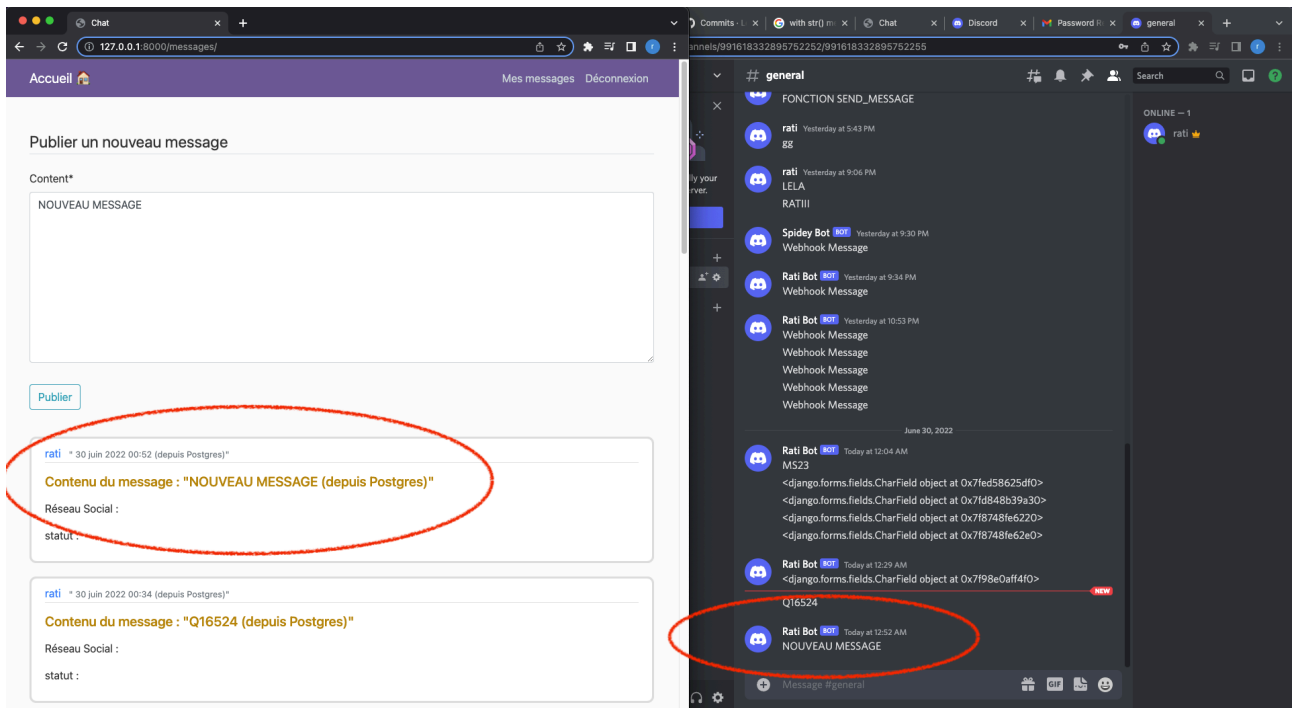
-> Envoyer un message à l'aide d'une fonction



-> En raison d'une activité anormale remarqué par Discord (m'obligeant à réinitialiser mon mot de passe plus de 4 fois en 6 heures), j'ai été obligé d'avoir recours aux Webhook de discord pour envoyer les messages depuis mon application.



-> Possible d'envoyer des messages sur Discord directement depuis l'application



-> Interdire l'accès à la page permettant d'envoyer des messages si l'utilisateur n'est pas connecté. Il sera redirigé vers la page de connexion ('login')

- Ajout de « LOGIN_URL = 'login' » dans le fichier settings.py
- Ajout de « login_required » dans le fichier chat/urls.py

-> Permettre à chaque utilisateur de consulter ses messages, et seulement ses messages non celui des autres utilisateurs

```
def list_message(request):
    if request.method == "GET":
        form = MessageRegisterForm()
        return render(request, 'list_messages.html', locals())
    elif request.method == "POST":
        form = MessageRegisterForm(request.POST)
        if form.is_valid():
            content = form.cleaned_data['content']
            send_content_to_discord = SendMessageToDiscord(content)
            form.instance.author = request.user
            form.save()
            form = MessageRegisterForm()
            user = request.user.id
            message = Message.objects.filter(author_id=user)
            return render(request, 'list_messages.html', {'messages': message, 'form': form})
```

-> Déploiement sur Heroku

-> Télécharger et installer Heroku

Source : <https://devcenter.heroku.com/articles/heroku-cli#download-and-install>

-> Installation de Gunicorn

commande => \$ **pip3 install gunicorn**

-> Créer le fichier requirements.txt

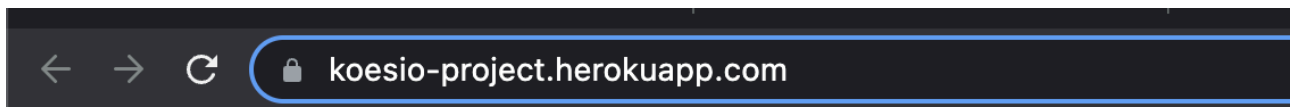
commande => \$ **pip3 freeze**

commande => \$ **pip3 freeze > requirements.txt**

-> Créer une application sur Heroku pour héberger l'application

commande => \$ **heroku create koesio-project**

```
(env) ratilekishvili@macbook-pro-de-rati send_messages_on_social_networks % heroku create koesio-project
Creating • koesio-project... done
https://koesio-project.herokuapp.com/ | https://git.heroku.com/koesio-project.git
(env) ratilekishvili@macbook-pro-de-rati send_messages_on_social_networks %
```

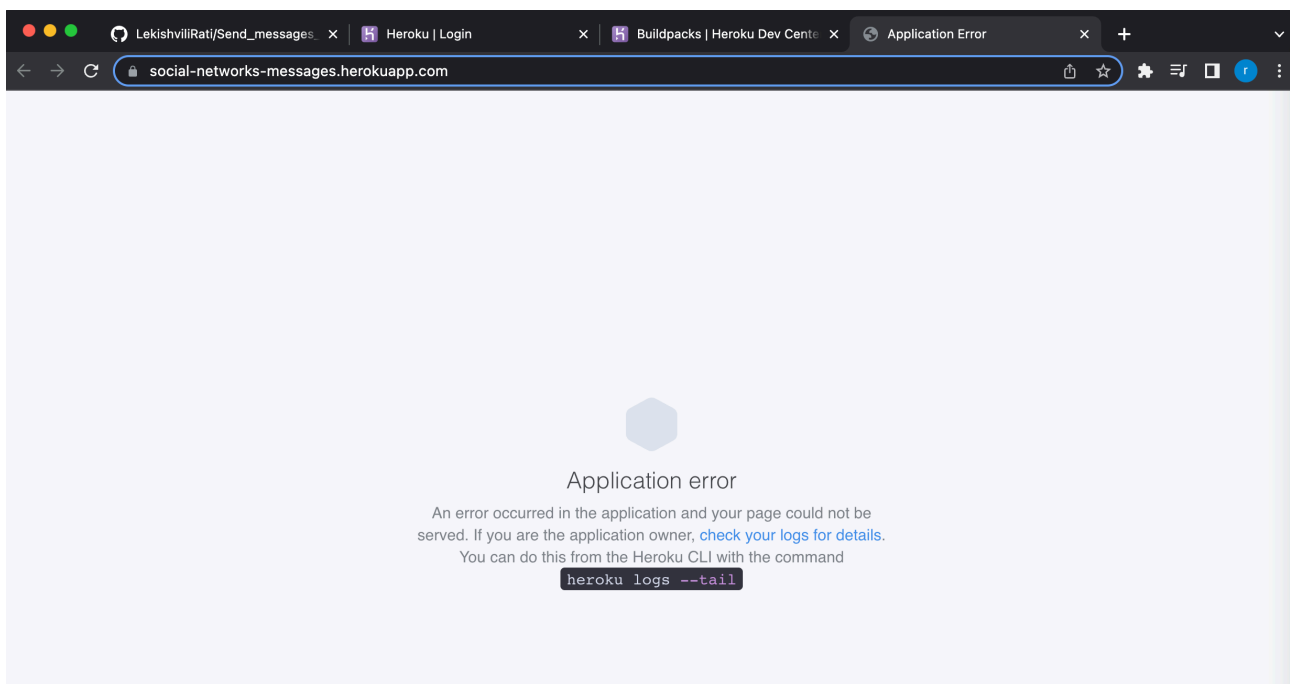


Heroku | Welcome to your new app!

Refer to the [documentation](#) if you need help deploying.

-> Application déployé mais : Application error affiché

```
remote: -----> Discovering process types
remote:      Procfile declares types -> (none)
remote:
remote: -----> Compressing...
remote:      Done: 33.5M
remote: -----> Launching...
remote:      Released v5
remote:      https://social-networks-messages.herokuapp.com/ deployed to Heroku
remote:
remote: This app is using the Heroku-20 stack, however a newer stack is available.
remote: To upgrade to Heroku-22, see:
remote: https://devcenter.heroku.com/articles/upgrading-to-the-latest-stack
remote:
remote: Verifying deploy... done.
To https://git.heroku.com/social-networks-messages.git
 * [new branch]      main -> main
(env) ratilekishvili@MacBook-Pro-de-rati send_messages_on_social_networks % heroku open
```

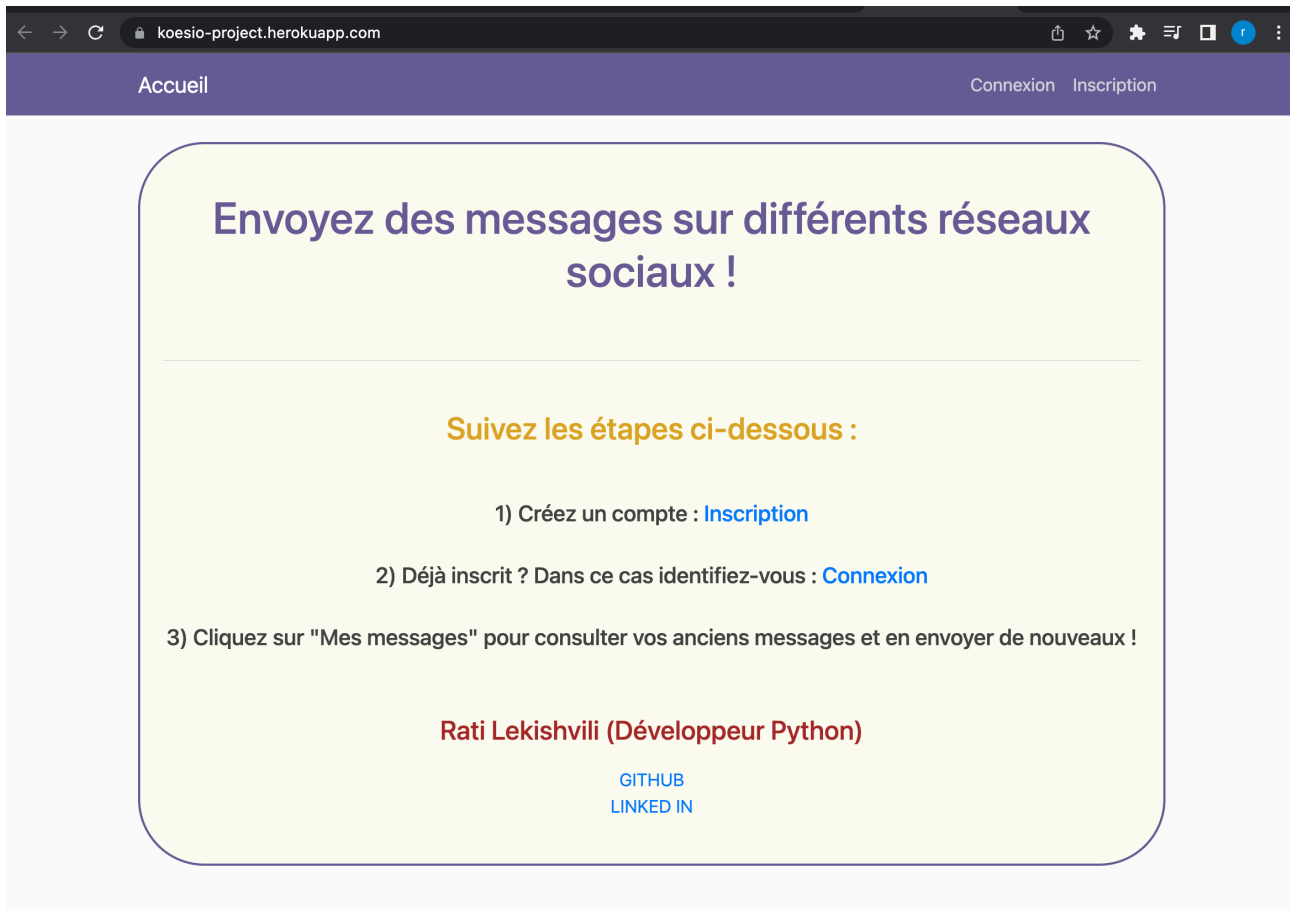


—> Solution :

Voir les erreurs : commande => \$ **heroku logs --tail**

-> Création « procfile » requis

—> Application déployé sur Heroku



-> Migrations sur la base de données Postgre SQL d'heroku.

commande => \$ **heroku run python3 manage.py migrate**