

CRUD NOSQL AVEC RAVENDB

# Stations services en fonction des régions

Yannis Bouttier  
Fabrice Robert  
Costes Duncan

# SOMMAIRE

1

CONTEXTE

2

PROBLÉMATIQUE

3

TECHNOS UTILISÉES

4

GESTION DE PROJET

5

PRESENTATION DB

6

ROUTES API

7

DÉMO

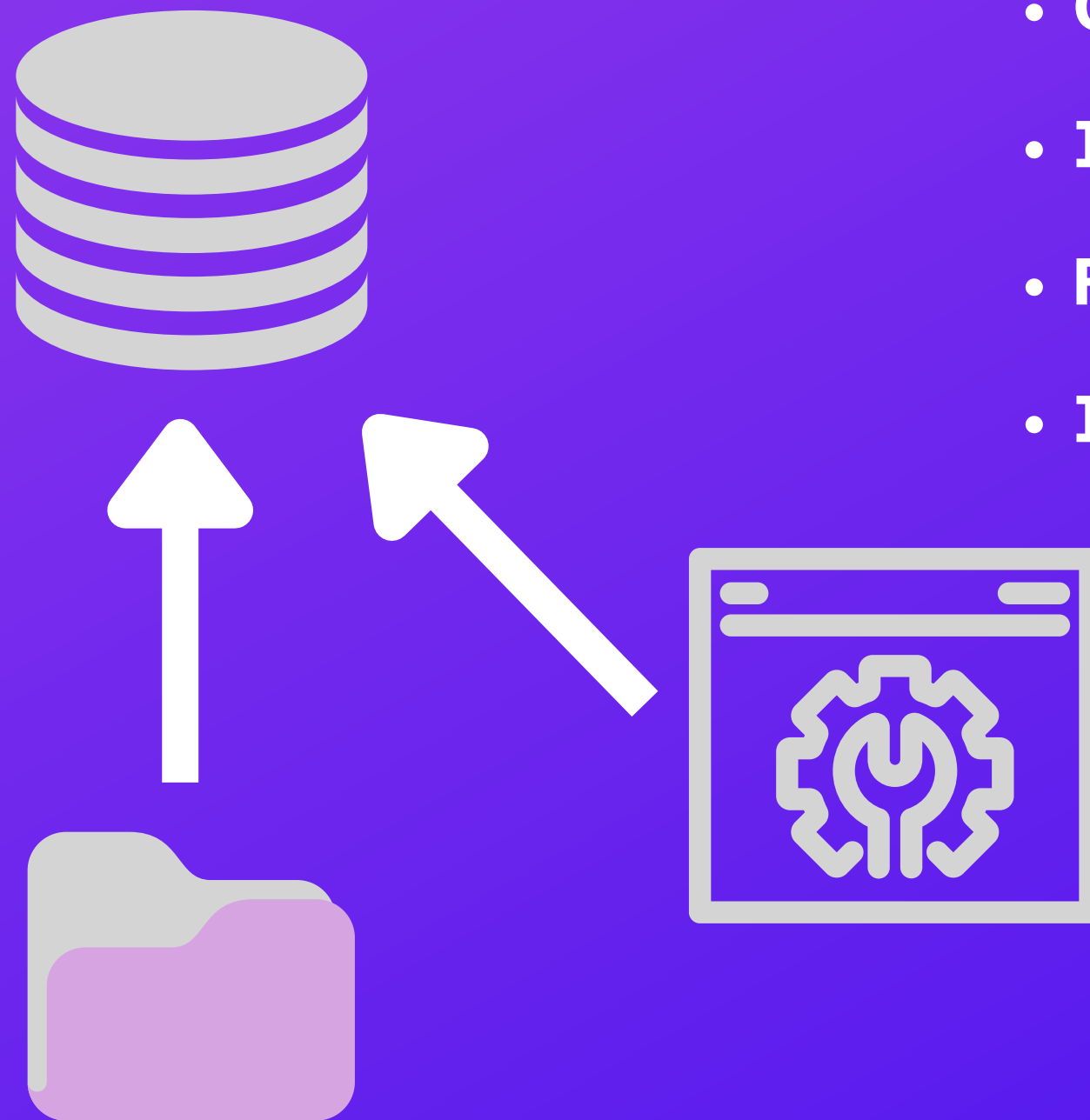
8

EXPÉRIENCES / AMÉLIORATIONS

9

CONCLUSION

# Contexte



- Concevoir une base de données NoSQL
- Injecter au moins 1 jeu de données
- Faire un CRUD
- Implémenter des requêtes d'agrégations



# PROBLEMATIQUE : PRIX ET SERVICES

Comment sont organiser les stations  
services dans la France ?

# TECHNOS UTILISÉES

## BACKEND



**Flask**

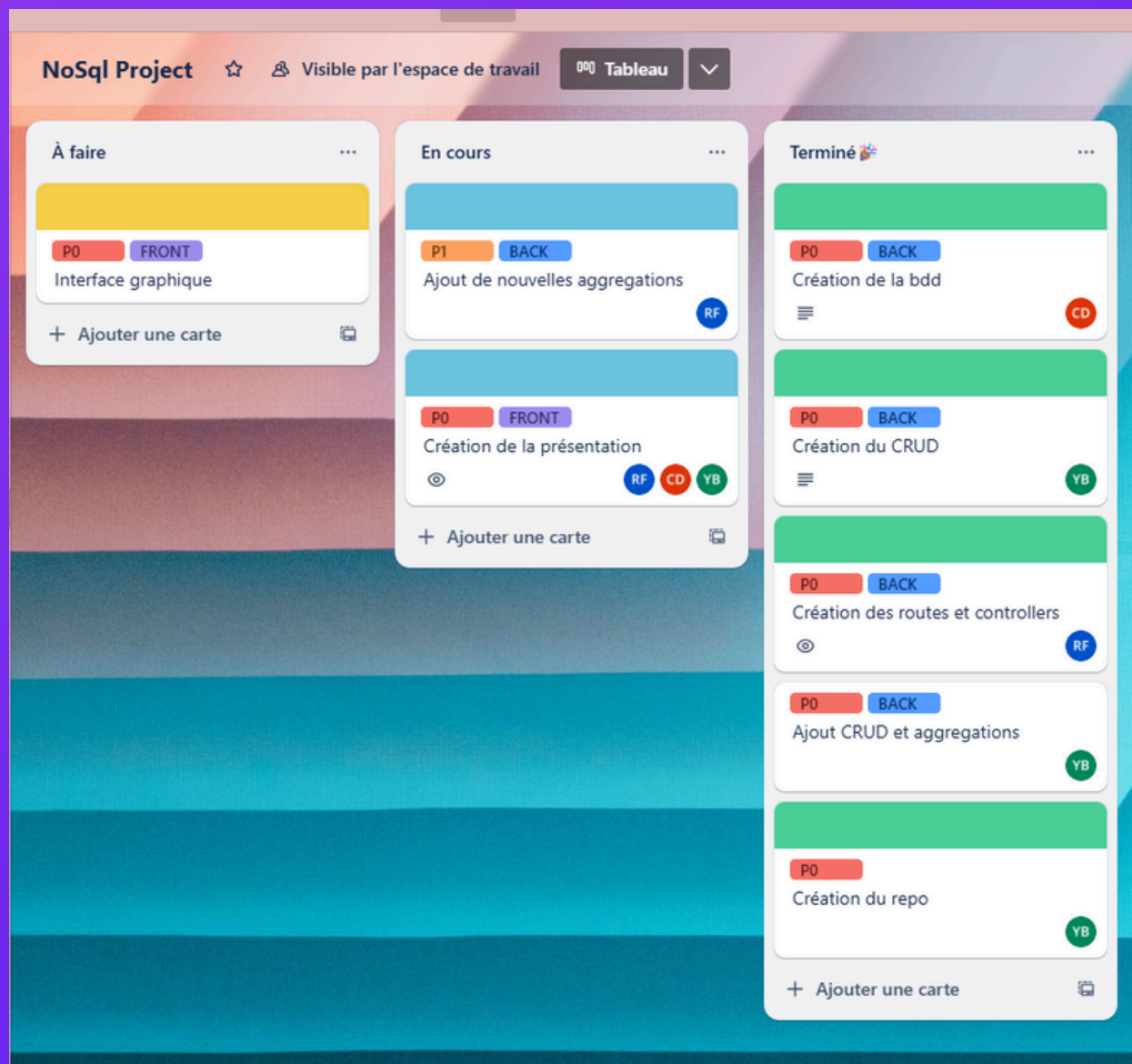
## BASE DE DONNÉES

**RavenDB**

# GESTION DE PROJET



Lien du trello



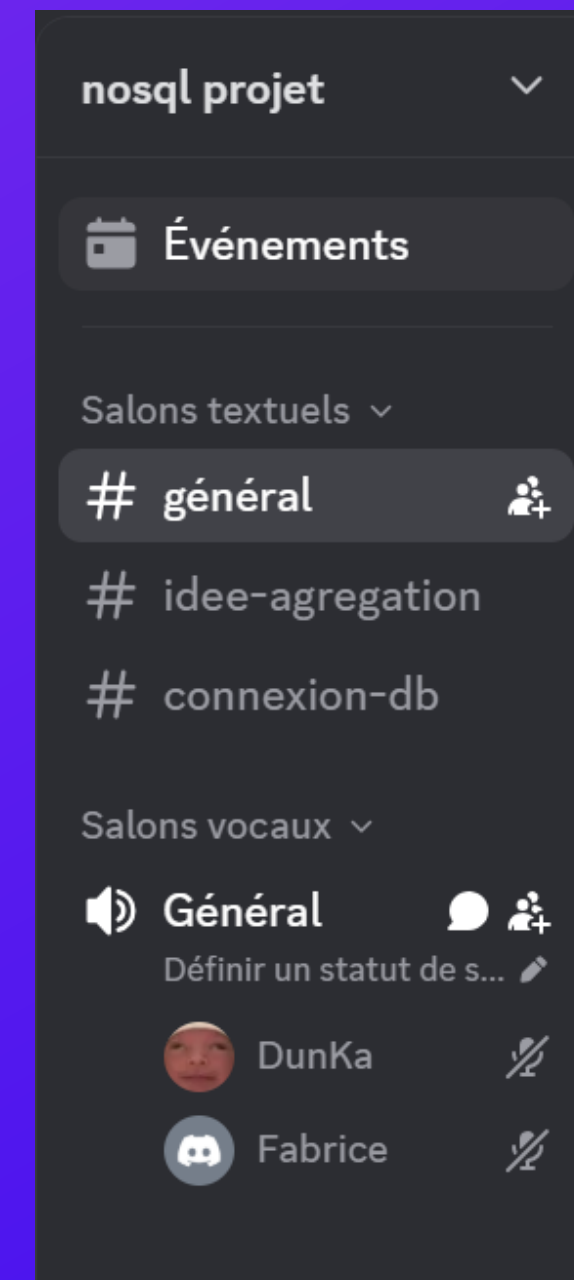
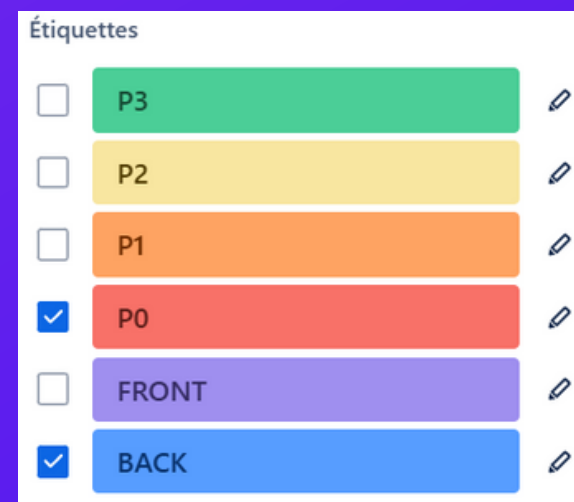
Priorité :

P0: Urgent

P1: Important/ Majeur

P2: Utile/ Moyenne

P3: Faible/Bonus



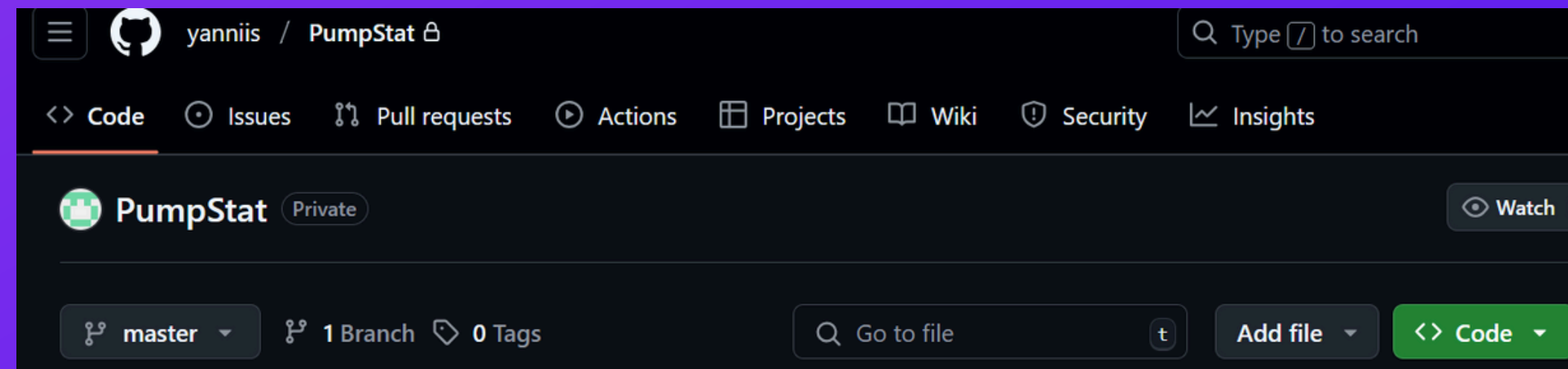


# STRUCTURE DU PROJET

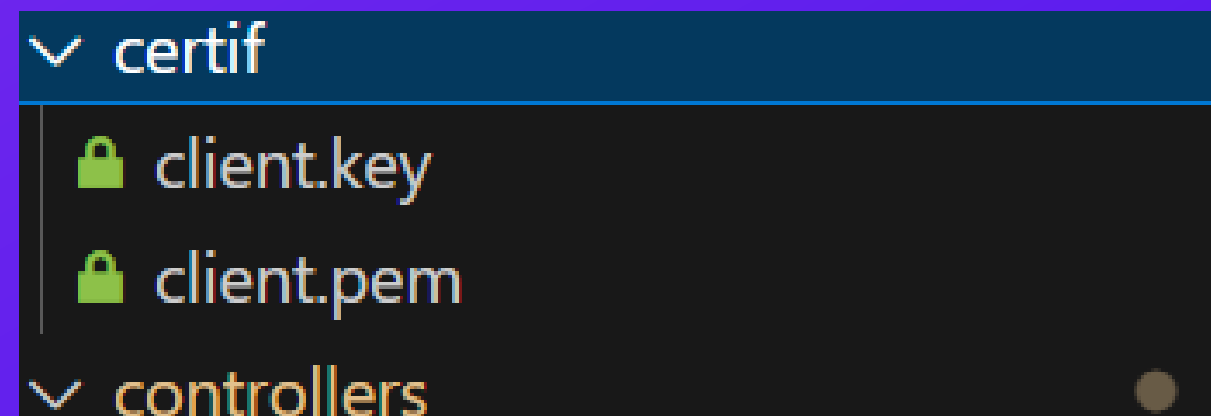
## Markdown

```
1  ...
2  └─ NoSql project
3     └─ .venv
4         .gitignore
5         pyvenv.cfg
6     └─ certif
7         client.key
8         client.pem
9     └─ controllers
10
11         carburants_controller.cpython-313.pyc
12         regions_controller.cpython-313.pyc
13         stations_controller.cpython-313.pyc
14         stats_controller.cpython-313.pyc
15     └─ __init__.py
16     carburants_controller.py
17     regions_controller.py
18     stations_controller.py
19     stats_controller.py
20 └─ routes
21
22     carburants.cpython-313.pyc
23     regions.cpython-313.pyc
24     stations.cpython-313.pyc
25     stats.cpython-313.pyc
26     __init__.py
27     carburants.py
28     regions.py
29     stations.py
30     stats.py
31 └─ templates
32     home.html
33 └─ .env
34 └─ .gitignore
35 └─ app.py
36 └─ responses.log
37
```

<https://github.com/yanniis/PumpStat>



Propre à RavenDB, génération d'un certificat :



# PRÉSENTATION DE LA DB





# RAVENDB



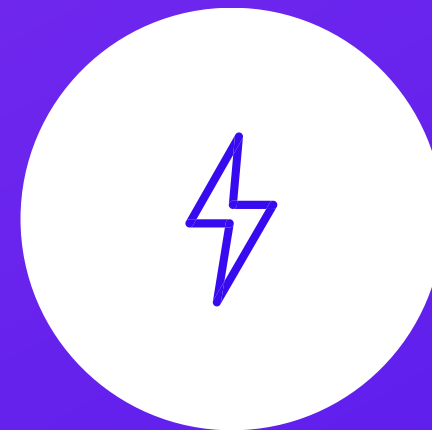
## ACID

Pensee pour la durabilite et la fiabilite



## MOYEN DE CONNEXION

Parfois compliquer car besoin d'un fichier pfx qui peut avoir des conflits



## OPEN SOURCE

Code opensource en c# .net



## DOCUMENTATION IMCOMPLETE

Documentation incomplete trop axee sur .net

# Routes API

## ROUTES BASIC CRUD

- Create
- Read
- Update
- Delete

## ROUTES AGREGATIONS

- Services le plus utilisé
- Prix le moins cher de chaque carburant
- Région avec le plus de services

# DÉMO DE L'API

# RETOUR D'EXPÉRIENCES

1

DÉCOUVERTE D'UNE  
NOUVELLE BDD

2

PRISE EN MAIN DES  
AGGRÉGATIONS

3

UTILISATION DES CERTIFS  
CLIENT KEY/PEM

# POINT D'AMÉLIORATION

1

**IMPLÉMENTATION  
INTERFACE**

2

**AJOUT DE REQUÊTE  
D'AGRÉGATION**

3

**MEILLEURE  
COMMUNICATION ET  
REPARTITION DES TÂCHES**



# CONCLUSION



**MERCI D'AVOIR ÉTÉ À L'ÉCOUTE**