

Documentation du projet

Introduction

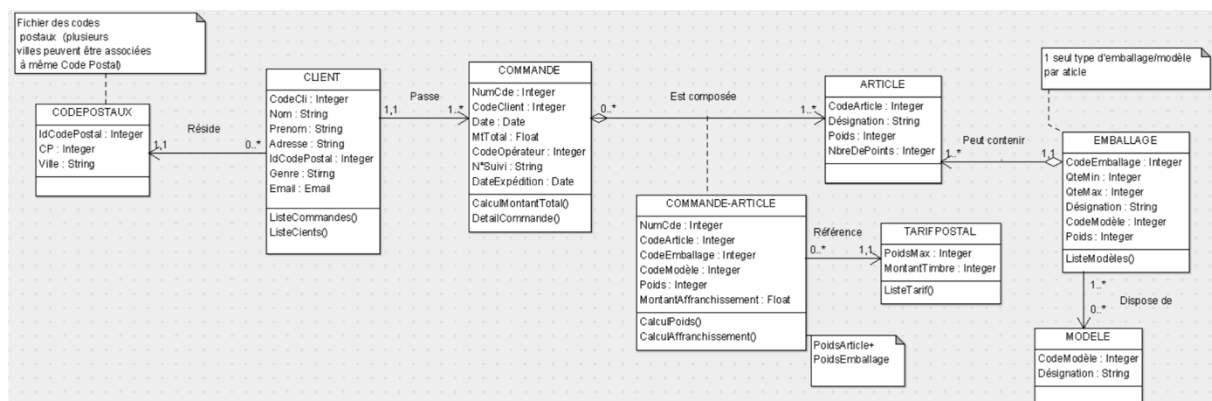
Bienvenue dans la documentation de notre projet, qui a pour objectif de fournir une compréhension détaillée de l'architecture, des modifications apportées en fonction du cahier des charges.

Objectif du Projet

L'objectif du projet est le développement d'une API Backend. Les principales fonctionnalités incluent la gestion des entités, l'interaction avec une base de données MySQL, et la documentation automatique de l'API à l'aide de Swagger.

Technologies Utilisées

- **Python** : Langage de programmation principal.
- **Flask** : Framework web léger pour la construction de l'application.
- **Mysql** : Langage de base de données
- **MariaDB** : Système de gestion de base de données relationnelle.
- **DBeaver** : Logiciel utilisé comme outil de base de données
- **Swagger** : Outil pour la documentation et la conception de l'API.
- **Katalon Recorder** : Extension utilisé pour effectuer les tests



Nous nous sommes appuyés sur la correction du diagramme UML pour concevoir notre base de données, ce qui nous a permis de représenter le fonctionnement attendu.

Notre travail s'est concentré sur quatre entités spécifiques, donc toutes les tables disponibles n'ont pas été utilisées.

La table Conditionnement, correspondant à la table Emballage, a été incluse.

De plus, une clé étrangère dans la table Commande a été ajoutée pour créer un lien direct avec la table Objet.

```
Last login: Wed Jun  5 10:03:51 on ttys004
(base) hetiaandro@airdehetiaandro-1 ~ % ssh-keygen
```


La commande `ssh-keygen` qu'on entre dans le terminal pour générer la clé SSH

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Authentication keys

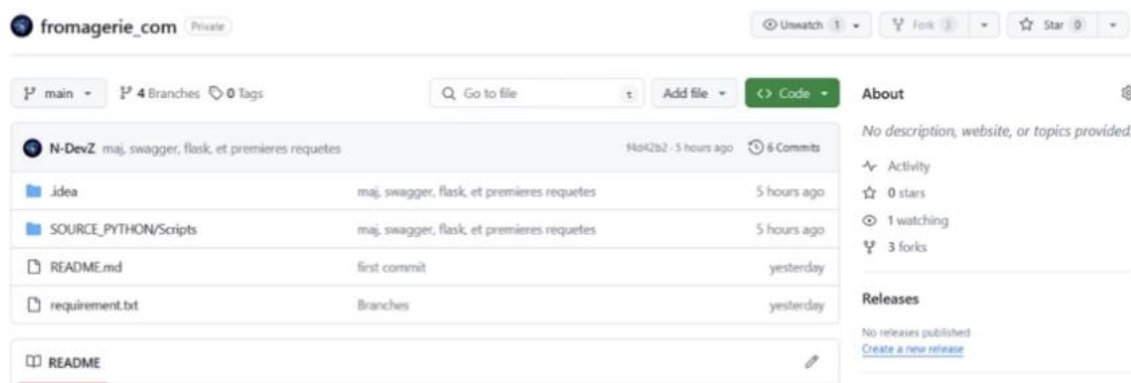


nat
SHA256:LJIDf0uuFQghCdbB1t741b1HzvEa8Py4G5r2D6vDzr0
Added on May 22, 2024
Last used within the last week — Read/write

Delete

Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

La clé SSH générée et entrée dans le terminal afin de permettre la connexion à GitHub sans avoir à indiquer le nom d'utilisateur et le mot de passe à chaque fois.



The screenshot shows a GitHub repository page for 'fromagerie_com'. The repository is private and has 4 branches and 0 tags. The main branch is 'main'. The repository contains several files and folders: '.idea', 'SOURCE_PYTHON/Scripts', 'README.md', and 'requirements.txt'. The 'README.md' file is selected, showing its content. The repository has 6 commits, with the most recent one from 5 hours ago. The repository is watched by 1 person, has 0 stars, and 3 forks. There are no releases published.

Le dépôt GitHub sépare en quatre branches correspondant au nombre de personnes dans le groupe de sorte à ce que chacun puisse avoir la sienne.

Dans ce dépôt il y aussi les différents livrables attendu (source python, readme, etc...)

```
C:\Program Files\MariaDB 11.3\bin>mysql -u new_user -p
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 194
Server version: 11.3.2-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database fromagerie_com;
```

Création de la database « fromagerie_com » dans l'environnement MariaDB

	Nom de la colonne	#	Type de donnée	Non Null	Auto-Incrémentation	Clef	Défaut	Extra	Expression	Commentaire
Colonnes	code_utilisateur	1	int(11)	[v]	[v]	PRI		auto_incre...		
Contraintes										
Clefs étrangères	nom_utilisateur	2	varchar(50)	[]	[]		NULL			
	prenom_utilisat...	3	varchar(50)	[]	[]		NULL			
Références	username	4	varchar(50)	[]	[]		NULL			
Triggers	couleur_fond_ut...	5	int(11)	[]	[]		NULL			
Indexes	date_insc_utilisa...	6	date	[]	[]		NULL			

Un exemple d'une des tables présentes dans la database (ici la table utilisateur)

```
class Utilisateur(Base):
    __tablename__ = "t_utilisateur"
    code_utilisateur = Column(Integer, primary_key=True)
    nom_utilisateur = Column(String(50), default=None)
    prenom_utilisateur = Column(String(50), default=None)
    username = Column(String(50), default=None)
    couleur_fond_utilisateur = Column(Integer, default=0)
    date_insc_utilisateur = Column(Date)
```

Et ici nous avons les lignes de codes qui ont permis la création de cette table avec le « code_utilisateur » en clé primaire.

```
app = Flask(__name__)
app.config["DEBUG"] = True
Base.metadata.create_all(bind=engine)

# Configuration de Swagger
SWAGGER_URL = '/swagger'
API_URL = '/static/swagger.json'
swaggerui_blueprint = get_swaggerui_blueprint(SWAGGER_URL, API_URL, config={'app_name': "API des utilisateurs"})
app.register_blueprint(swaggerui_blueprint, url_prefix=SWAGGER_URL)
```

Ici on configure Swagger en déclarant les variables qui comportent l'URL où le fichier est situé (au format json) et le chemin permettant de visualiser et de tester l'API

POST	/commandes	Ajouter une nouvelle commande	▼
GET	/clients/{codcli}/commandes	Obtenir les commandes par client	▼

Ci-dessus, l'interface Swagger qui nous permet d'effectuer des tests fonctionnels.

Katalon Recorder 7.0.0

Workspace

- Test Suites
 - Passed: 1
 - Failed: 0
 - Exo3 *
 - fromagerie *
 - users ***
- Dynamic Test Suites
- Test Data
- Extension script
- Profiles

users

+ Add new tag

Command	Target	Value
open	http://127.0.0.1:5000/swagger/	
click	xpath=//div[@id=operations-default-getUtilisateurs]/div[2]/div/div/div[2]/button	
click	xpath=//div[@id=operations-default-getUtilisateurs]/div[2]/div/div[2]/button	

+ Add new row (Cmd/Ctrl + I)

Log | Screenshots | Variables | Reference | Self-healing | Dashboard | All

Test Case	Status	Broken Locator	Proposed	Approve
users	Passed	xpath=//div[@id=operations-default-getUtilisateurs]/div[2]/div/div/div[2]/button	xpath=//*[normalize-space(text()) and normalize-space(.)='Obtenir la liste des utilisateurs'] [1]/following-button[1]	<input type="checkbox"/>
users	Passed	xpath=//div[@id=operations-default-getUtilisateurs]/div[2]/div/div[2]/button	xpath=//*[normalize-space(text()) and normalize-space(.)='Parameters'] [1]/following-button[2]	<input type="checkbox"/>

Approve

Afficher le Bureau

Sign in to enable automatic backup. Refresh

Pour finir nous avons effectué des tests dynamiques avec katalon recorder