



Projet de stage de fin d'études

Recherche et développement démonstrateur web: le développement web full-stack d'un flux de travail boutique-logistique-entrepôt

Zijun PAN

Encadrement : Vincent Ricordel, Bruno Le Fellic

Mars - Août 2020

Sommaire

Sommaire

Abstract

1. Introduction

2. Génération des données fictives

3. Modifications des APIs Python

4. Angular pour l'Interface Homme-Machine

5. Déploiement en Docker

6. Conclusion

Abstract

Le thème de ce stage de fin d'études est sur le développement démonstrateur d'un flux de travail boutique-logistique-entrepôt. Mon travail comprend le développement de pages, la modification d'API et le déploiement de pages Web.

Dans la partie back, on utilise MongoDB, une base de données distribuée, universelle et basée sur des documents, pour stocker les données fictives générées par un script Python. L'API de ce projet est construite avec Python Flask et Swagger, qui nous permet de récupérer les données persistantes.

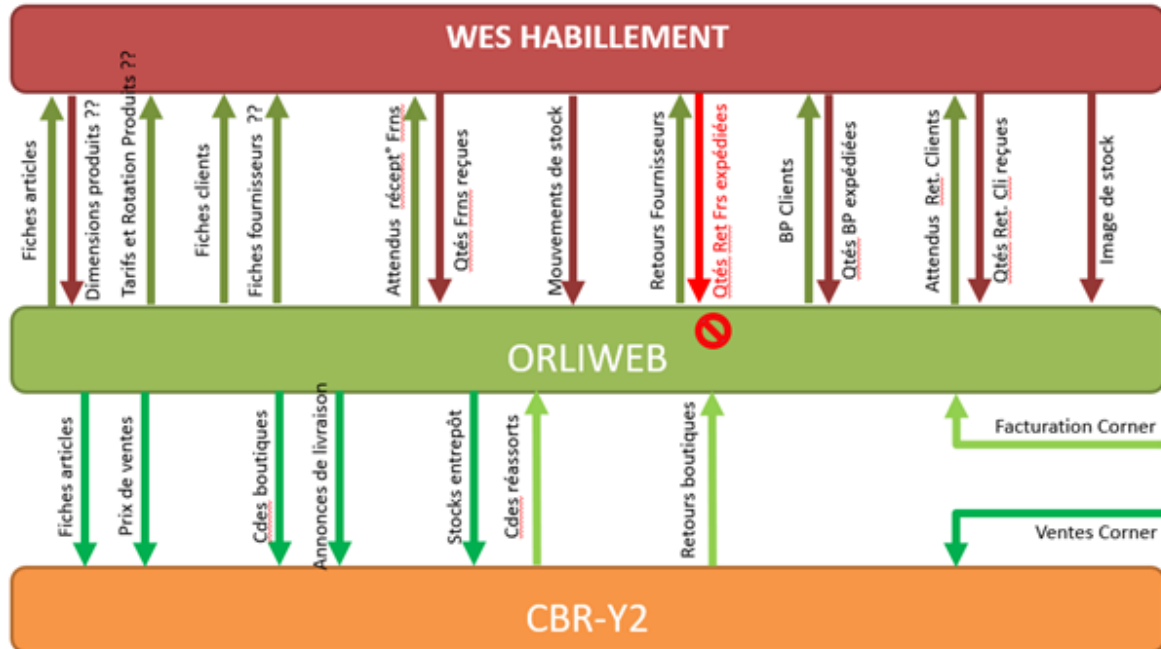
Dans la partie front, on utilise Angular, le framework front-end de Google, pour construire un site web structuré et évolutif. Typescript(un sur-ensemble de Javascript), CSS et HTML sont à la base de la construction d'un composant Angular. Angular Material, une bibliothèque de composants UI, est utilisée pour améliorer l'interface utilisateur.

Le logiciel Git est utilisé pour la synchronisation de projet avec le dépôt Git distant. Le front et le back sont connectés avec la structure JSON. Docker et docker-compose sont utilisés pour le déploiement de nos logiciels front et back à une machine Linux.

1. Introduction

Le stage de cette année est un stage de six mois en informatique dont le principal contenu est la recherche et le développement d'une démonstration pour les services.

Voici un organigramme de notre projet de gestion logistique. Il s'agit notamment de magasin (CBR-Y2), de réseau logistique (ORLIWEB) et d'entrepôt (WES HABILLEMENT).



Reprises de données :

- Dans ORLIWEB : Fiches articles dans ORLIWEB | Fiches clients négoce | Tarifs Négoce + PVC | Stocks Entrepôts
- Dans Y2 : Fiches clients Retails (hros WEB) | Stocks boutiques | Tarifs Retail Outlet | Vendeurs – Avoirs - Historiques ventes

Nous pouvons voir qu'il existe de nombreux workflows ci-dessus, ce sont les données qui seront générées dans le travail réel. Mais comme notre projet est un projet de recherche, nous utiliserons plutôt des données virtuelles.

C'est la première fois que j'entre dans une entreprise pour faire un stage, donc ce stage m'a donné l'opportunité de comprendre le flux de travail industriel et m'a permis d'apprendre à travailler professionnellement.

Tous ces éléments ont apporté un soutien conceptuel plus riche à la gestion de notre projet de fin d'études mené en 5ème année d'Électronique et Technologies Numériques.

2. Génération des données fictives

Dans cette partie, nous avons terminé l'opération de génération de données de simulation. Nous avons spécifiquement utilisé Python comme langage pour générer des scripts et nous nous sommes connectés à la base de données MongoDB pour implémenter l'insertion de documents. De plus, nous utilisons également JSON comme type de données pour réaliser une communication sans état entre les API.

Key	Value
▼ (1) ObjectId("5eec7c883a9723ff993d7c4c")	{ 16 fields }
_id	ObjectId("5eec7c883a9723ff993d7c4c")
status	success
protocol	Retail
reference	1126fad1-e090-ff4c-63e5-cc09ab4a15c
is_last_update	true
file_type	CMD_REASSORT
file_name	shop_ERP_CMD_REASSORT_2019-07-02_2.csv
created_at	2019-07-02T02:52:26
destination	ERP
updated_at	2019-07-04T15:54:53
source	shop
file_line	14
w_id	ce70cb96-8fc4-963e-9406-255d7a2a4ea
error	{ 3 fields }
protocol_version	1.0
json_content	{ 14 fields }

Un exemple d'une commande générée dans MongoDB

3. Modifications des APIs Python

Le serveur principal de notre projet utilise le framework Flask et Swagger pour construire ce service. Notre ancien projet est un système typique de gestion de l'information logistique des télécommunications, donc notre travail consiste à modifier l'ancien projet pour l'adapter à notre nouveau type de données et système de type de service.

etl_file

GET	/etl_file
POST	/etl_file

etl_order_list

GET	/etl_order
GET	/etl_order/{item_reference}
GET	/order_history/{item_reference}

Une exemple de Swagger UI pour visualiser l'API

4. Angular pour l'Interface Homme-Machine

Nous avons utilisé Angular comme framework frontal. Angular est une plateforme de développement qui permet de créer des applications web dynamiques et immersives. Nous avons choisi l'Angular Material pour faciliter et améliorer le design visuel.

Voici une présentation du site web.

SE DÉCONNECTER

Les filtres pour filtrer les commandes

Nous utilisons Angular pour faire beaucoup de conception fonctionnelle. Par exemple, nous pouvons ajouter une zone de recherche pour filtrer les résultats EAN, ou nous pouvons ajouter des informations supplémentaires sur le timeline.

Ici en ci-dessous nous présente les filtres qu'on a créés.

Les filtres pour filtrer les commandes

La page détail pour une commande particulière.

Détails de la commande

07/03/2019 23:23:42

Nom du fichier

ERP_WMS_ATTENDU_RECEP_FRNS_19960207_6
_csv

Destination

WMS

Source

ERP

Date de dernière mise à jour

29/05/2020 18:28:15

14/04/2019
21:52:31



STOCK_ENTREPOT

› Voir la totalité des données

11/03/2019
02:05:04



* QTES_RECU_FRNS

› Voir la totalité des données

07/03/2019
23:23:42



ATTENDU_RECEP_FRNS

› Voir la totalité des données



Le timeline est affiché dans la page détail

De plus, nous avons trié et trié certaines commandes erronées.

19/10/2018 00:32:31



FICHE_ARTICLE **ERR_STG**

Erreur de stockage du ERP

› Voir la totalité des données

Une commande erronée

5. Déploiement en Docker

Après avoir terminé la conception front-end et back-end, notre projet peut s'exécuter normalement sur le serveur. Mais si nous voulons déployer le projet afin que le projet puisse toujours fournir des services et être plus facilement accessible, nous devons utiliser Docker.

```
start@dev-zpa:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
259daac7f592	etl-tracking-front:develop	"nginx -g 'daemon of..."	7 days ago	Up 7 days	80/tcp, 0.0.0.0:8010->8000/tcp	etl-tracking-front
7a4a78156978	etl-tracking-web-api:develop	"bash /spikelabs/run..."	7 days ago	Up 7 days	0.0.0.0:8011->8000/tcp	etl-tracking-web-api
f3d86b709cd5	mongo:latest	"docker-entrypoint.s..."	3 months ago	Up 3 months	0.0.0.0:27017->27017/tcp	keen_ganguly

Les services sont stockées dans des conteneurs docker

6. Conclusion

Grâce à ce projet, j'ai maîtrisé de nombreuses technologies et outils de développement front-end et back-end avec des scénarios d'application pratiques, tels que Git et Docker. Et ce projet de recherche a été fait par moi-même, et cela m'a aussi permis d'apprendre à gérer le projet et à être responsable de mon propre projet.