Équipe 1

Reconnaissance automatique des matricules et compilation des notes pour optimiser le travail des enseignants

Guide du développeur

Version 1.0

Guide du développeur

Ce guide d'installation présente une configuration temporaire désignée pour un développeur en phase de développement.

Code Source

Le code source du logiciel est accessible à partir de Gitlab. Il est disponible à l'adresse suivante:

https://gitlab.com/polytechnique-montr-al/log89xx/22-3/equipe-1/LOG89XX-1

Configuration temporaire (environnement de développement)

La configuration temporaire est désignée pour une installation temporaire du logiciel dans un environnement local. L'utilisation du logiciel dans l'environnement de développement requiert quelques étapes manuelles. Il n'est donc pas recommandé d'utiliser cette configuration pour un utilisateur autre qu'un développeur.

Application Web

L'application web s'exécute sur le cadriciel Angular 11. Il est nécessaire d'installer npmjs pour la gestion des modules utilisées par l'application et pour l'installation du cadriciel Angular 11.

Il est possible d'installer npmjs à partir du lien suivant:

https://nodejs.org/en/download/

L'installation de Angular 11 peut se faire par l'exécution de la commande suivante:

```
npm install -g @angular/cli
```

De plus, il est nécessaire d'installer les modules utilisés par l'application à l'aide de npmjs par l'exécution de la commande suivante dans le répertoire *LOG89XX-1/Projet4/cadriciel/* :

```
npm install
```

Une fois toutes les dépendances installées, l'application peut être lancée:

```
npm start
```

L'application est accessible à partir du lien suivant (par défaut au port 4200):

http://localhost:4200

Serveur SocketIO

Le logiciel de reconnaissance inclut un serveur SocketIO utilisé pour établir un canal de communication bidirectionnelle entre le serveur et le client (l'application web).

Le serveur SocketIO s'exécute sur Python 3.9. Il est préférable de créer un environnement virtuel de Python 3.9 pour dissocier les dépendances du serveur SocketIO des autres modules. Une fois l'environnement virtuel créé, il est nécessaire d'installer les dépendances requises à l'aide de python-pip par l'exécution de la commande suivante à partir du répertoire *LOG89XX-1/SocketIO/* :

```
pip install -r requirements.txt
```

Une fois toutes les dépendances installées, le serveur peut être lancé à partir du répertoire *LOG89XX-1/SocketIO/* :

```
python3 app.py
```

Le serveur est accessible à l'adresse suivante au port 7000 par défaut:

http://localhost:7000

Serveur API

Le logiciel de reconnaissance inclut un serveur API utilisé pour recevoir les requêtes du client (application web).

Le serveur SocketIO s'exécute sur Python 3.9. Il est préférable de créer un environnement virtuel de Python 3.9 pour dissocier les dépendances du serveur API des autres modules. Une fois l'environnement virtuel créé, il est nécessaire d'installer les dépendances requises à l'aide de python-pip par l'exécution des commandes suivantes à partir du répertoire *LOG89XX-1/Server/* :

```
pip install -r requirements.txt
pip uninstall -y PyCrypto
pip install PyCryptodome
pip uninstall -y PyMuPDF
pip install PyMuPDF
pip uninstall -y httplib2
pip install httplib2==0.15.0
```

De plus, le serveur API requiert l'installation du logiciel TexLive pour le traitement des fichiers LaTeX. Ce logiciel peut être installé en exécutant les commandes suivantes:

```
sudo apt install -y texlive-latex-recommended
sudo apt-get install -y texlive-lang-french
```

Une fois toutes les dépendances installées, le serveur peut être lancé à partir du répertoire *LOG89XX-1/Server/* :

```
python3 app.py
```

Le serveur est accessible à l'adresse suivante au port 5000 par défaut:

http://localhost:5000

Exécuteur de tâches

Le logiciel de reconnaissance inclut un serveur SocketIO utilisé pour établir un canal de communication bi-directionnelle entre le serveur et le client (l'application web).

L'exécuteur de tâches s'exécute sur Python 3.10. Il est préférable de créer un nouvel environnement virtuel de Python 3.10 pour dissocier les dépendances de l'exécuteur des autres. Une fois l'environnement virtuel créé, il est nécessaire d'installer les dépendances requises à l'aide de python-pip par l'exécution de la commande suivante à partir du répertoire *LOG89XX-1/Executor/* :

```
pip install -r requirements.txt
```

Après avoir installé toutes les dépendances nécessaires, on peut rouler l'exécuteur avec la commande suivante:

```
python job_executor.py
```

Il est à noter que le module d'exécution de tâches n’est pas automatisé dans l’environnement de développement. Ainsi, le développeur doit s’assurer de lancer le script d’exécution lorsqu’il y a une tâche présente dans la file de tâches.

File de tâches

La file de tâches est hébergée sur Redis. On peut retrouver un guide d’installation de Redis à partir du lien suivant : https://redis.io/docs/getting-started/ .

Après qu’il soit installé, il faudra créer un serveur Redis localement. Il est possible de le créer avec la commande suivante :

```
redis-server
```

Il est hébergé sur le port 6379 par défaut. Les tâches seront ajoutées dans la clé “**job_queue**”.

Introduire des changements aux codes sources

Lors d’un ajout d’une nouvelle dépendance, il faut s’assurer de mettre à jour le fichier *dockerfile* associé. Les modules qui ont un dockerfile associé sont les suivants: l’application web (*LOG89XX-1/Projet4*), le serveur SocketIO (*LOG89XX-1/SocketIO*), le serveur API (*LOG89XX-1/Server*), ainsi que l’exécuteur de tâches (*LOG89XX-1/Executor*).