Documentation

Présentation

Ce projet propose un système d'automatisation qui permet d'extraire un ensemble de contacts depuis une image. L'utilisateur n'a pas à se soucier des programmes utilisés, une chaîne d'automatisation est créée et nécessite simplement le dépôt de l'image source et d'un fichier de configuration.

Etapes

- **Extraction du texte brut**, qui récupère simplement l'ensemble des caractères et des mots reconnus sur l'image;
- **Extraction des informations**, qui récupère les informations demandées par l'utilisateur;
- **Conversion** des données au format souhaité, indiqué par l'utilisateur.

Sources

Les codes sources sont fournis dans l'archive `.tgz`, dans laquelle cette documentation se situait. Vous y trouverez les fichiers suivants :

- `start`, qui permet de démarrer la chaîne de conteneurs
- `stop`, qui permet de l'arrêter
- `envoyer`, qui envoie un fichier image et un fichier de configuration au début de la chaîne
- `recevoir`, qui récupère le rendu à la fin de la chaîne

Après extraction, **la première chose à faire est d'ajouter les droits d'exécution** à tous les scripts :

```
chmod +x start stop envoyer recevoir
```

Ces 4 scripts sont ceux avec lesquels un technicien ou un utilisateur doit interragir. Ils contiennent les commandes essentielles et complètes pour la gestion de la chaîne. D'autres fichiers sources, internes au fonctionnement, se situent dans le dossier `src/` :

- `tesseract.sh`, script de la première étape, qui extrait le texte brut de l'image
- `extraction.php`, script de la deuxième partie, qui extrait les informations demandées
- `conversion.sh`, script de la troisième partie, qui convertit au format souhaité
- `modele.php`, modèle utilisé par la troisième partie pour générer un HTML

Fichier de configuration

Le fichier texte de configuration doit absolument se nommer `config.txt`. L'utilisateur y rentre d'abord la liste des champs, puis le format souhaité.

Champs

Les champs permettent de préciser quels types d'information récupérer :

- `nom`
- `prenom`
- `email`
- `telephone`
- `adresse`

L'ordre entré par les utilisateurs reste le même sur le résultat, ainsi si `prenom` est entré avant `nom`, le résultat aura lui aussi d'abord le prénom puis le nom.

Les informations complémentaires

Pour les champs **nom**, **prenom** et **adresse**, il est nécessaire de renseigner des informations complémentaires. Par exemple, pour une capture d'écran de la forme :

```
{
    "last_name" : "Deschamps"
    "first_name" : "Didier"
    "address" : "Paris",
    "phone" : "07.08.09.10.11"
}
```

Un fichier de configuration ressemblera à ceci :

```
nom : "last_name"
prenom : "first_name"
telephone
adresse : "address"
```

Ces informations complémentaires permettent au programme de détecter les données souhaitées quand ces dernières n'ont pas de motif récurrent (comme c'est le cas d'un email ou d'un numéro de téléphone).

Cas du tableau

Si l'image fournie en entrée est sous forme de tableau alors l'utilisateur doit le spécifier au début du fichier, avec le nombre **total** de colonnes (y compris celles non-extraites). Dans ce cas particulier, **chaque champ** doit être accompagné du titre de sa colonne en informations complémentaires. Un fichier de configuration type ressemblerait à :

```
tableau
colonnes : 3

nom : "Name"
telephone : "Phone"
```

Format de sortie

Le format de sortie est nécessaire au bon fonctionnement de la chaine. Le système propose les formats **PDF**, **HTML** et **CSV**. Le format de sortie doit être précisé comme suit :

```
format : "PDF"
```

Règles sur le fichier de configuration

Pour éviter d'obtenir des erreurs en sortie, vous pouvez suivre les règles suivantes :

1. Mettre les informations entre guillemets
2. Ne pas rajouter d'accents sur les noms des champs (le moteur OCR ne les reconnait pas toujours)
3. Bien préciser le nom de chaque colonne pour le format tableau

Fonctionnement interne

Les 3 conteneurs suivants ont été créés à l'avance et sont utilisés. Certains apportent un programme en particulier (ex: Tesseract OCR), et tous sont fournis avec PHP 7.4 :

Conteneur	Étape
`tesseract`	Extraction du texte brut
`php-cli74`	Extraction des informations souhaitées
`weasyprint`	Conversion

Le démarrage de la chaîne fonctionne comme suit :

1. Un volume nommé `sae103` est créé. Il contiendra les fichiers traités par les conteneurs
2. Un conteneur temporaire, nommé `sae103-tmp`, est démarré et lié au volume. Il sert de port d'entrée vers le volume le temps de transférer les sources
3. Les fichiers du dossier `src/` sont tranférés dans le volume par le conteneur temporaire
4. Les dossiers de résultats `etape1/`, `etape2/` et `etape3/` sont créés dans le volume
5. Le conteneur temporaire est fermé et supprimé
6. Les 3 conteneurs finaux sont lancés en mode détaché et liés au volume. On lance dans chaque conteneur le programme associé.

Chaque programme du dossier `src/` attend qu'un fichier de configuration arrive dans le dossier qu'il regarde (selon l'étape).

Exemple d'utilisation complète

Les commandes, dans l'ordre, à lancer pour l'utilisation de la chaîne sont les suivantes. Les fichiers doivent être au préalables marqués comme exécutables.

Démarrage de la chaîne :

```
./start
```

Envoi d'une image, puis du fichier de configuration. **Le fichier de configuration doit être mis en dernier**, car il est l'élément déclancheur de la chaîne :

```
./envoyer dossier/image.png
./envoyer dossier/config.txt
```

Enfin, la récupération des éléments produits par la chaîne, dans le dossier `out_dir` :

```
./recuperer
```

Répétez autant de fois que nécessaire l'envoi et la récupération des fichiers produits. Pour finalement arrêter la chaîne, exécutez :

```
./stop
```

Cas de test

La documentation est fournie avec 2 cas de tests pour comprendre l'utilisation et le fonctionnement de la chaîne.

Cas n°1

Le premier test s'effectue avec un tableau, se trouvant de le dossier `test/tableau/`. Il possède un fichier `config.txt`, ainsi qu'une image `tableau.png`

Lorsque l'utilisateur souhaite lancer le test, il lui suffit d'entrer les commandes suivantes (une fois la chaîne lancée) :

```
./envoyer test/tableau/tableau.png
./envoyer test/tableau/config.txt
```

Puis pour récupérer les fichiers produits :

```
./recuperer
```

Cas n°2

Le second cas s'effectue avec une capture d'un fichier JSON. Il se trouve dans le dossier `test/json/`. De même que pour le premier test, il possède une image `sample.png` et un fichier `config.txt`.

Pour lancer le test, il faut lancer les commandes suivantes (une fois la chaîne lancée) :

```
./envoyer test/json/sample.png
./envoyer test/json/config.txt
```

Puis pour récupérer les fichiers produits :

```
./recuperer
```