

INTERFACES DE VISUALISATION DES DONNÉES

Dans le cadre du Hackathon des Archives Nationales 2018, l'un des axes de travail était la visualisation des données d'une fiche producteur.

Nous avons choisi de visualiser les cinq informations suivantes pour le notaire en cours de consultation :

- Les occupations des différents clients du notaire.
- Les types d'acte produit par le notaire.
- L'activité du notaire dans le temps.
- Les dix clients les plus suivis par le notaire
- L'aire d'influence du notaire

Si vous souhaitez être informé des mise à jours majeurs de cette documentation, veuillez watcher l'issue suivante sur GitHub : <https://github.com/UneMinuteAgo/stats/issues/1>

Installation du projet

Installation sous Docker

Si vous avez le logiciel **Docker** installé sur votre PC, vous pouvez suivre ce chapitre. Néanmoins, il faudra également disposer d'un certain nombre d'outil pour aller jusqu'au bout de l'opération.

Voici la liste des outils à installer en fonction de vore environnement

Outil	Windows	Linux	Description
Docker	X	X	Comprenant : <code>docker</code> , <code>docker-compose</code> et <code>docker-machine</code> ¹
GIT	X	X	Requis pour son <code>MinGW</code> et son intégration à Windows ²
Make	X	-	Commande <code>make</code> ³

¹ : Pour Windows avec Docker Toolbox.

² : L'intégration de GIT dans Windows permet de fusionner les commandes Linux avec celles de Windows.

³ : `make` est une commande Linux en standard.

Lorsque les outils sont installés, rendez-vous à la racine du projet où se trouve le fichier `README.md` et lancez une invite de commande. Tapez la commande : `make install` .

La première installation prendra du temps, car **Docker** doit générer l'image du système à l'aide du fichier `docker/core/Dockerfile` . Cela comprend le téléchargement de librairies et de mise à jour du système. L'étape suivante consiste à télécharger les dépendances du projet qui va prendre également un certain temps.

Lorsque le processus d'installation sera terminée, la démonstration est utilisable à l'adresse suivante : `http://localhost` .

Si la page ne se charge pas et que la commande `make ls` indique dans la colonne `STATUS` du style `Up About a minute` il y à probablement un problème de redirection de port.

CONTAINER ID	IMAGE	COMMAND	CREATED	NAMES	STATUS
6c3b0ca392db	hackhan	"/bin/bash /entrypoi..."	Less than a second ago	Up Less	than a second
	0.0.0.0:80->80/tcp, 0.0.0.0:443->443/tcp, 3307/tcp, 0.0.0.0:3307->3306/tcp			hackhan	

Si la page se charge avec des erreurs, patientez que l'installation des dépendances se termine.

Si après un certain temps, cela ne fonctionne toujours pas, tapez la commande `make hackhan` pour vous rendre dans la machine.

Tapez la commande `composer update` pour vérifier que les dépendances soient bien installées et à jour.

Si cela n'aboutit toujours pas, contactez l'équipe Une Minute Ago.

Installation Manuelle

Si vous avez déjà un serveur Web et que vous souhaitez simplement avoir les ressources, il suffit de cloner le projet pour les obtenir.

Néanmoins, il faudra se munir de l'outil `composer` pour procéder à l'installation des dépendances.

Clonage du dépôt

Lancez une invite de commande et tapez la commande suivante :

```
git clone --recurse-submodules https://github.com/UneMinuteAgo/stats.git
```

Le dossier cloné `stats` se trouvera à l'emplacement où vous avez tapé la commande. Aller dans le dossier `stats` à l'aide de la commande `cd stats`.

Si le dossier a déjà été cloné, pour intégrer les sous-modules, exécutez la commande suivante :

```
git submodule update --init --recursive
```

Installation de Composer

L'installation présente via Docker installe et exécute la commande qui procède à l'installation des dépendances.

Si vous souhaitez utiliser les ressources sur votre propre installation et que vous n'avez pas `composer`, suivez les instructions suivantes. Si vous avez déjà `composer`, vous pouvez passer au chapitre suivant.

Dans une invite de commande, tapez les commandes suivantes :

```
wget https://getcomposer.org/composer.phar
chmod +x composer.phar
mv composer.phar /usr/local/bin/composer
```

Vérifiez que l'installation s'est déroulée correctement à l'aide de la commande suivante : `composer --version`

Création du projet

Une fois que `composer` est opérationnel, si vous n'êtes pas dans le dossier cloné `stats`, rendez-vous y à l'aide de la commande `cd stats`.

Tapez la commande suivante pour obtenir les dépendances : `composer create-project`

Se mettre à jour

Pour maintenir à jour les ressources obtenu depuis le **GitHub**, effectuez de temps en temps la commande suivante : `git fetch --all`

S'il y a eu des mises à jour, il faudra les appliquer à l'aide de la commande `git pull origin master`. Cela implique que vous n'ayez modifiés aucun fichier.

Pour mettre à jour les sous-modules, lancer la commande suivante :

```
git submodule update --remote --merge
```

Pour mettre à jour les dépendances PHP, tapez la commande suivante :

```
composer update
```

Interfaces de visualisation

Le projet étant fourni avec un `docker-compose.yml`, très simplement utilisable à l'aide de la commande `make install`, les URL présentés dans la suite de la documentation se baseront sur des adresses en `localhost` pointant vers le site web installé dans Docker.

Si vous utilisez les ressources sur une autre installation, pensez à adapter les chemins (si nécessaire).

Interfaces pour les Pies Charts

L'interface pour les Pies Charts sert à obtenir depuis un fichier XML producteur un jeu de données calculé afin de produire un graphique. Le Pie Chart est le graphique que nous avons choisis, mais ce jeu de données peut être utilisé pour n'importe quel autre graphique¹.

¹ Compatibilité à vérifier sur les autres formats de graphiques.

Prévisualisation en standalone

La projet offre par le biais de la page `http://localhost/index.php`, une interface pour visualiser, tester et manipuler un graphique. Cette page utilise le script (`site/lib/iface/getStats.php`) qui produit le jeu de données exploitable par le moteur `Pie` disponible dans `site/lib/js/Pie.js`.

Obtention des données via "API"

L'interface `site/lib/iface/getStats.php` produit un JSON à partir d'un fichier XML producteur. Dans le cadre de ce projet, les fichiers XML doivent se trouver dans le dossier `site/lib/xml`.

Cette interface accepte les requêtes via les méthodes `GET` and `POST`. Pour la documentation, nous opterons pour la méthode `GET` qui est plus simple et plus visuelle.

Voici la liste des arguments acceptés :

- `file` : Nom du fichier XML à traiter (de type `String`).
- `nbResultValue` : Définit le x premières valeurs que doit retourner l'interface. Le reste est agrégé sous l'unité `Autres`. L'argument est de type `Integer` et doit être supérieur ou égal à `3` pour être effectif.
- `data` : Il s'agit de l'information que l'on souhaite agrégé depuis l'XML de DTD `ead` au niveau d'un item. Exemple pour agrégé les occupations, l'argument vaut `controlaccess:occupation`

URL pour récupérer les 10 premières occupations du notaire **Claude Royer II** :
`http://localhost/lib/iface/getStats.php?file=FRAN_IR_041106.xml&data=controlaccess:occupation&nbResultValue=10`.

Vous pouvez tester directement sur l'interface en ligne utilisé pour le Hackathon :

[https://hackhan.neoblaster.fr/lib/iface/getStats.php?
file=FRAN_IR_041106.xml&data=controlaccess:occupation&nbResultValue=10](https://hackhan.neoblaster.fr/lib/iface/getStats.php?file=FRAN_IR_041106.xml&data=controlaccess:occupation&nbResultValue=10)

Resultat :

```
[{"label": "commer\u00e7ant", "value": 88}, {"label": "conseiller du roi", "value": 26}, {"label": "avocat au  
parlement (Ancien R\u00e9gime)", "value": 22}, {"label": "officier (arm\u00e9e)", "value": 21},  
{"label": "officier de la maison militaire du roi (Ancien R\u00e9gime)", "value": 21},  
{"label": "procureur de justice (Ancien R\u00e9gime)", "value": 16}, {"label": "ma\u00e7on", "value": 15},  
{"label": "employ\u00e9 de maison", "value": 13}, {"label": "notaire", "value": 12},  
{"label": "jardinier", "value": 12}, {"label": "Autres", "value": 320}]
```

Interfaces pour les GeoJson

Prévisualisation en standalone

Obtention des données via "API"
