

Notice technique du logiciel VisualGraphs

1. Installation.....	2
1.1 Logiciels requis.....	2
1.1.1 Utilisation de Neo4J.....	2
1.2 Gestion de plusieurs instances de VisualGraphs.....	2
1.2.1 Gestion multi instances de Neo4J.....	2
1.2.2 Modification de VisualGraphs.....	3
1.2 Architecture du logiciel.....	3
2. Import des données.....	4
2.1 Fonctionnement de Neo4J.....	4
2.2 Import des données en PHP.....	4
3. Création de requêtes.....	6
3.1. Les requêtes générales.....	6
3.2. Les requêtes liées à un élément.....	6
3.3. Les requêtes avec des paramètres.....	7

Cette notice est dédiée à l'utilisation avancée du logiciel VisualGraphs pour les développeurs et les personnes souhaitant créer des requêtes complexes ou mettre en ligne une instance de VisualGraphs.

1. Installation

L'application VisualGraphs est une application web, elle peut être utilisée en local sur un ordinateur ou être mise en ligne.

Cette partie décrit l'architecture du logiciel VisualGraphs afin de faciliter sa mise en ligne, si vous souhaitez installer le logiciel en locale se référer à la notice d'installation.

1.1 Logiciels requis

Pour faire fonctionner le logiciel VisualGraphs nous avons besoin d'un serveur HTTP, de PHP (version 5 minimum) et d'une base de donnée Neo4J.

1.1.1 Utilisation de Neo4J

Pour le téléchargement de Neo4J, voir la partie 1.2 de la Notice d'Installation.

Pour démarrer la base de données, il faut écrire dans le terminal « *Dossier_de_neo4j/bin/neo4j start* ».

Si la démarrage de Neo4j échoue à cause d'un problème de permission, tapez dans le terminal :
« *chmod +x Dossier_de_neo4j/bin/neo4j* »

Pour stoppez la base de donnée, il faut écrire dans le terminal « *Dossier_de_neo4j/bin/neo4j stop* ».

1.2 Gestion de plusieurs instances de VisualGraphs

Il est possible de faire fonctionner plusieurs versions de Visualgraphs sur une même machine, mais cela demande de faire quelques paramétrages, notamment l'installation d'une seconde base de donnée Neo4j.

1.2.1 Gestion multi instances de Neo4J

Pour pouvoir avoir plusieurs instances de Neo4j sur un même serveur web, il faut modifier les ports utilisé par Neo4j dans le fichier de configuration :

- Éditez le fichier de configuration 'neo4j.conf' situé dans le dossier 'conf'.
- Modifiez le port du connecteur Bolt (*exemple: dbms.connector.bolt.listen_address=:7688*).
- Modifiez le port du connecteur HTTP (*exemple: dbms.connector.http.listen_address=:7474*).
- Désactivez la connexion HTTPS (*dbms.connector.https.enabled=false*).

Une fois les modifications de paramétrage terminées vous pouvez démarrer la base Neo4J.

1.2.2 Modification de VisualGraphs

Pour que VisualGraphs puisse se connecter à la base de donnée Neo4j que nous venons de configurer, il faut modifier le fichier 'visualgraphs/ScriptsPHP/Connexion.php'.

Dans ce fichier il suffi de mettre la variable **\$port** égale au port 'du connecteur Bolt' choisi dans la configuration de Neo4j.

1.2 Architecture du logiciel

Le dossier principal du logiciel est constitué de 5 dossiers et de 2 fichiers :

- **Documents** : Contient les notices du logiciel.
- **Donnees** : Contient les interfaces de gestion des données (Import de documents CSV).
- **Parametrage** : Contient les fichiers liées au menu de paramétrage du design et des paramètres du logiciel (Fichiers PHP correspondant aux interfaces de paramétrage).
- **Requete** : Contient le générateur de requête CYPHER.
- **VisualGraphs** : Contient le logiciel. (code javascript, PHP, images, ...)
- **index.html** : Menu principale du logiciel.
- **style.css** : Définit le design des menus.

Le dossier VisualGraphs contient (entre autres) les dossiers :

- **json** : Contient les documents de paramétrage de design et des paramètres (Style.json et Params.json) générés à partir des menus de paramétrage.
- **Images** : Dossier à utiliser pour importer une image dans le logiciel (l'écran d'introduction et les images contenues dans les propriétés), *ces images sont automatiquement redimensionnées par le logiciel selon la taille de l'écran, il est conseillé d'utiliser des images de grande résolution pour éviter qu'elles soient pixelisées sur les écrans de grande taille.*
- **img** : Dossier contenant les images utilisées par le logiciel.

Pour la mise en ligne du logiciel, il est conseillé de ne laisser que le dossier 'VisualGraphs' sans les menus de paramétrages.

2. Import des données

Il existe deux méthodes d'import des données dans VisualGraph, soit à partir de documents CSV préformatés pour le logiciel (*voir la notice d'utilisation du logiciel partie 3*), soit grâce à un script PHP qui permettra de convertir une base de donnée relationnel en une base de donnée Neo4J.

2.1 Fonctionnement de Neo4J

La base de donnée Neo4J est une base de donnée de format graphe, elle est constituée de deux entités :

- Des nœuds représentant un 'élément' (une Personne, un Lieu, ...) dans notre base de donnée.
- Des liens reliant ces nœuds.

Les nœuds et les liens peuvent être constitués de propriétés (sous forme de nombre ou de texte).

2.2 Import des données en PHP

La classe *Importeur* qui se trouve dans le dossier « *visualgraphs\Donnees\BDDRelversBDDGraph* » possède des méthodes qui simplifient l'importation des données via des scripts PHP.

Elle permet :

- De se connecter à la base de données relationnel (par exemple Mysql) et à la base de donnée Neo4J.
- De récupérer des données sur une base de donnée relationnel via des requêtes SQL.
- D'écrire les données des nœuds et des liens dans la base Neo4J à partir des données récupérées.

Création de l'objet importeur, connexion à la base de données relationnel et à la base de données Neo4J :

```
$importeur = new Importeur();
$importeur->connexion("mysql", "localhost", "projet", "root", "passwordBDDRel", "neo4j",
"passwordNeo4J");
$importeur->connexion("Type_de_la_BDD_Relationnel", "Adresse_du_serveur_de_la_BDD_Relationnel",
"Nom_de_la_BDD_Relationnel", "utilisateur_BDD_Relationnel", "Mot_de_passe_BDD_Relationnel",
"utilisateur_neo4J", "Mot_de_passe_neo4J");
```

Récupération des informations dans la base de données relationnel et écriture dans la base de données Neo4J :

Récupération d'informations dans la base de données relationnel et création d'un Noeuds dans la base de donnée Neo4J grâce aux informations récupérées.

```
// Exemple de récupération des lieux dans la base Editef
$r = "SELECT id_lieu as id, ville_norme_fr as nom_fr, ville_norme_pays as nom, ville_source as
nom_source, pays, coord FROM lieu";
$importeur->recupNoeud($r, 'Lieu', ['id', 'nom', 'nom_fr', 'nom_source', 'pays', 'coord']);
$importeur->recupNoeud("requete_sql", "Type_de_Noeud", Tableau listant des propriétés à récupérer);
```

Récupération des informations dans la base de données relationnel et création d'un Lien dans la base de donnée Neo4J grâce aux informations récupérées.

```
// Exemple de récupération des lieux d'édition dans la base Editef
$r = "SELECT id_Edition as e1, lieu_Edition as e2, 'Edité_à' as type_relation FROM edition";
$importeur->recupLien($r, 'Edité_à', 'Edition', 'Lieu', []);
$importeur->recupLien("requete_sql", "Type_de_Lien", "Type_de_Noeud1", "Type_de_Noeud2", Tableaux
listant des propriétés à récupérer);
```

Le sens du lien va du Noeud1 vers le Noeuds2 (si l'on souhaite qu'il n'y est pas de sens, il est possible de retirer le sens d'un lien dans les paramétrage de VisualGraphs).

Dans la requête SQL : e1 = ID du Noeud1 et e2 = ID du Noeud2 (**Indispensable**).

**L'exemple d'import de la base Editef est fourni dans le dossier
« visualgraphs\Donnees\BDDRelversBDDGraph ».**

3. Création de requêtes

Cette notice a pour but d'apporter une aide à l'écriture des requêtes du logiciel VisualGraphs. Les requêtes sont écrits en langage CYPHER qui permet de récupérer des données sur les base de Neo4J.

Introduction aux requêtes Neo4j : <http://logisima.developpez.com/tutoriel/nosql/neo4j/introduction-neo4j/>

Refcard du CYPHER : <https://neo4j.com/docs/cypher-refcard/current/>

Dans VisualGraphs les requêtes s'écrivent à partir du menu de paramétrage (cf la partie 4.1.2 de la notice d'utilisation).

3.1. Les requêtes générales

Les requêtes générales correspondent aux requêtes liées aux boutons présents dans l'interface de recherche.

Exemples :

Requête basique récupérant les personnes aillant une lien Né_à avec un lieu.

```
MATCH (n1:Personne)-[r:Né_à]-(n2:Lieu) RETURN n1, n2, r
```

Les requêtes peuvent être constituées de plusieurs requêtes séparées par un point-virgule.

Requête récupérant les éditions et les exemplaires liées à une collectivité sur deux niveaux.

```
MATCH (e:Collectivité) RETURN e;  
MATCH (e:Collectivité)-[r1]-(edi:Edition) RETURN e, r1, edi;  
MATCH (e:Collectivité)-[r1]-(edi:Edition)-[r2]-(ex:Exemplaire) RETURN e, r1, edi, r2, ex;  
MATCH (e:Collectivité)-[r3]-(ex2:Exemplaire) RETURN e, r3, ex2;  
MATCH (e:Collectivité)-[r3]-(ex2:Exemplaire)-[r2]-(edi:Edition) RETURN e, r3, ex2, r2, edi
```

3.2. Les requêtes liées à un élément

Cette partie explique la création des requêtes qui seront appelé par les boutons présents dans l'interface d'information. Ces boutons sont liées à un type de nœud ou de lien particulier et ne sont affiché que lorsqu'un élément du même type est sélectionné.

Exemples :

Récupération de la personne sélectionné et de sa famille proche.

```
MATCH (p:Personne) WHERE id(p)=|this| RETURN p;  
MATCH (p:Personne)-[r:Relation_Familiale]-(p2:Personne) WHERE id(p)=|this| RETURN p, r, p2;
```

- |this| → correspond à l'identifiant de notre élément (soit identifiant de l'élément dans neo4j ou soit la propriété id de l'élément).

Exemple avec propriété la propriété 'id' d'un nœud :

Récupération de la personne sélectionné et de sa famille proche

```
MATCH (p:Personne) WHERE p.id=|this| RETURN p;  
MATCH (p:Personne)-[r:Relation_Familiale]-(p2:Personne) WHERE p.id=|this| RETURN p, r, p2;
```

On peut également utilisé ce mot clé dans la description de la requête afin de récupérer automatiquement le nom de l'élément sélectionnée dans la description.

Exemple de description d'une visualisation de la famille d'une personne

```
Visualisation de la famille de |this|
```

Si la personne sélectionné s'appelle « Pierre Dupont », nous obtiendrons la description suivante

```
Visualisation de la famille de « Pierre Dupont »
```

3.3. Les requêtes avec des paramètres

Les requêtes avec paramètres sont des requêtes pour lesquelles des informations sont demandées à l'utilisateur. Elle peuvent également être liées à un élément (un noeud ou un lien).

Exemple :

Choix par l'utilisateur de deux lieux dans lesquels ont travaillé des personnes, ce qui permettra de récupérer les personnes aillant travaillé dans ces deux lieux.

```
MATCH (l1:Lieu)-[r]-(prof1:Etape_Professionnel)-[r2]-(perso:Personne)-[r3]-(prof2:Etape_Professionnel)-[r4]-(l2:Lieu)  
WHERE l1.nom = "#0|#" AND l2.nom = "#1|#"  
RETURN l1, l2, prof1, prof2, perso, r, r2, r3, r4
```

- #0|# et #1|# → Propriétés entrées par l'utilisateur.

Pour ces requêtes il faut remplir la ligne 'Paramètres d'indication : ' du menu de paramétrage dans laquelle ont indique les messages qui seront affichés à l'utilisateur :

Exemple de paramètres d'indication pour la requête précédente :

```
Indiquez le premier lieu; Indiquez le second lieu;
```

Les différents messages doivent être séparés par un point-virgule (;), il doit y avoir autant de messages que de numéro de paramètres.

Dans notre exemple :

- |#0|# → Indiquez le premier lieu.
- |#1|# → Indiquez le second lieu.