Mise à jour de TaxRef

Présentation des étapes pour migrer TaxRef malgré l'utilisation en production d'une version ancienne de TaxHub (v1.9.4). Réaliser d'abord toutes ces étapes en local avant de les reproduire en production.

Pré-requis : la base de données ne doit pas avoir évolué... Sinon, il faut vérifier que les modifications sont compatibles avec la branche `feat/sinp`. Le code SQL de migration peut être ajouté au script SQL `pre_detection.sql` (voir ci-dessous). Rechercher les éventuelles évolutions dans la liste des releases.

Migration de TaxRef en version :

- v17 depuis v16 : évolution de la base impactant la migration entre la v1.9.4 et la v1.14.0 ⇒ nécessite de créer la table `taxonomie.t_meta_taxref` dans script SQL de pré-détection.
- v16 depuis v15 : pas d'évolution de la base impactant la migration entre la v1.9.4 et la v1.11.3.

Installation de la dernière version de TaxHub

- Se connecter sur l'hôte `web-srv`
- Se placer dans le dossier `~/www/` :

```
cd ~/www/
```

- Cloner le dépôt Taxhub :

```
git clone https://github.com/PnX-SI/TaxHub.git taxhub.taxref_v17
```

  - Mettre à jour les sous-module si utilisation en local en mode "dev" :

```
cd ~/www/taxhub.taxref_v17 ; git submodule init; git submodule update
```

- Se placer dans le dossier cloné :

```
cd ~/www/taxhub.taxref_v17
```

- Copier les fichiers de configuration :

```
cp settings.ini.sample settings.ini
cp apptax/config.py.sample apptax/config.py
```

  - Éditer le fichier `settings.ini` :

```
vi settings.ini
```

    - Modifier les 3 paramètres suivant :

```
db_host=10.0.1.20
```

```
# Database name
db_name=geonature2db

# Database owner username
user_pg=geonatadmin

# Database owner password
user_pg_pass=<geonatadmin-password>
```

- Éditer le fichier `config.py` :

```
vi apptax/config.py
```

  - Modifier les 2 paramètres suivant :

```
SQLALCHEMY_DATABASE_URI =
"postgresql://geonatadmin:<geonatadmin-
password>@10.0.1.20:5432/geonature2db"
SECRET_KEY = '<generate-super-secret-key-with-uuid>'
```

- Lancer l'installation du `venv` :

```
./install_app.sh
```

- Activer le `venv` :

```
source venv/bin/activate
```

- Lancer l'import de TaxRef :

```
flask taxref migrate-to-v17 import-taxref-v17
```

  - Pas certain que cela soit utile si on utilise le scripts SQL `pre_detection.sql` mais on peut lancer la commande suivante pour tester d'éventuel changement réalisé dans la base :

```
flask taxref migrate-to-v17 test-changes-detection
```

- Lors de la réalisation des tests en local, préparer une nouveau dossier de version de TaxRef sur le dépôt `sinp-<region>-data` :
  - Voir : PACA ou AURA
  - Copier les fichiers csv générés par le script précédent dans le dossier `taxref/v17/csv/` du dépôt.
  - Créer les scripts SQL qui permettront la migration dans le dossier `taxref/v17/sql/`. Se baser sur la version précédente de TaxRef pour avoir des exemples de script SQL qu'il faudra adapté aux informations fournies dans les fichiers CSV.
- Lors de la mise en production, récupérer les fichiers SQL permettant la migration :

```
wget https://raw.githubusercontent.com/cbn-alpin/sinp-<region>-
data/main/taxref/v<version>/sql/pre_detection.sql
wget https://raw.githubusercontent.com/cbn-alpin/sinp-<region>-
```

```
data/main/taxref/v<version>/sql/post_detection.sql
```

- Lancer la migration de TaxRef :

```
flask taxref migrate-to-v17 apply-changes --script_predetection
./pre_detection.sql --script_postdetection ./post_detection.sql
```

  - **NOTE** : pour une raison inconnue, dans le cas du SINP AURA, il m'a fallu lancer à 2 reprises cette commande. Après l'import, 14 conflits sont indiqués. Après le premier lancement, il reste 8 conflits. Après le second, cela tombe 0 et la migration s'effectue correctement...
- Vérifier que la nouvelle version de TaxRef a bien été migré correctement ! Vérifier qu'un cd_nom disparu dans la v17 n'est plus présent dans votre table `taxonomie.taxref` et qu'un nouveau cd_nom ajouté à partir de la v17 s'y trouve bien.
- Si nécessaire (avant la v17), lier à nouveau les statuts de protection aux zones géographiques :

```
flask taxref link-bdc-statut-to-areas
```

  - Désactiver à nouveau les textes de la BDC Statut qui ne vous sont pas nécessaire à votre installation :
    - SINP PACA :

```
psql -h localhost -U geonatadmin -d geonature2db -f ~/data/db-
geonature/data/sql/04_disable_status_text.sql
```

    - SINP AURA :

```
psql -h localhost -U geonatadmin -d geonature2db -f ~/data/db-
geonature/data/sql/003_disable_status_text.sql
```

  - Ajouter les éventuels nouveaux textes de cette nouvelle version de Taxref à ne pas prendre en compte à votre script `04_disable_status_text.sql`.
- Vérifier que la nouvelle version de la BDC Statuts a bien été migrée correctement ! Il faut s'assure que les textes avec le champ `enable` à `true` dans la table `taxonomie.bdc_statut_text` correspondent bien à la France métropolitaine, à la région ou aux département qui concernent le SINP. Il peut s'avérer nécessaire de relancer le script SQL de sélection des textes du SINP (dépôt `sinp-<region>-data/db-geonature/data/sql/…_disable_status_text.sql`) pour activer les nouveaux textes.
  - Si pour une raison ou une autre, il est nécessaire de recharger la BDC statut, il est possible de suivre la démarche définie dans le ticket #501 de TaxHub. Il faudra juste modifier l'URL de téléchargement des fichiers zip et la version de TaxRef si nécessaire.

Détail utilisation du script "flask taxref migrate-to-vXX"

- Ressource : https://github.com/PnX-SI/TaxHub/tree/master/apptax/taxonomie/commands/migrate_taxref
- Voir aussi le ticket corrigeant des problèmes dans la migration et la détection des conflits : https://github.com/PnX-SI/TaxHub/issues/527
- Lancer le script : `flask taxref migrate-to-v17 import-taxref-v17`
  - La récupération de toutes les informations permettant la migration définitive vers la dernière version de TaxRef nécessite de lancer plusieurs fois ce script. Des fichiers de

sortie différents seront ainsi générés.

- Le fichier liste_changements.csv est généré. Le charger dans le Drive. Figer la ligne 1 et y ajouter des filtres.
    - Dans ce fichier tous les changements ne sont pas à corriger. À l'aide des filtres de la ligne 1, afficher seulement les lignes dont la colonne `action` afficher seulement lignes commençant par "Conflicts with attributes". Sélectionner ces lignes et les colorer en rouge.
    - Avec la colonne `f_cd_ref` trier les résultat de A à Z : cela va regrouper les lignes par f_cd_ref et il sera plus simple de repérer les cd_ref à remplacer.
    - La colonne `f_cd_ref` indique le cd_ref retenu et la colonne `i_cd_ref` indique le cd_ref remplacé. Pour chaque `f_cd_ref` on va trouver une ligne avec la même valeur dans `i_cd_ref` et dans `f_cd_ref`, elle est à ignorer. Il faut repérer les lignes où ils sont différents.
        - Pour chaque ligne :
            - récupérer le `i_cd_ref` et s'en servir pour supprimer les attributs correspondant de la table `cor_taxon_attributs` dans le script SQL de prédétection.
            - si la colonne `media_nb` indique une valeur non NULL, il faut aussi remplacer l'attribution du cd_ref pour les médias correspondant dans le script SQL de prédétection en récupérant les valeurs des colonnes `i_cd_ref` et `f_cd_ref`. Si plusieurs `i_cd_ref` sont fusionnés dans le même `f_cd_ref`, utiliser une clause IN.
- Relancer le script en essayant d'appliquer les changements `flask taxref migrate-to-v17 apply-changes` et en indiquant cette fois l'utilisation des fichiers SQL de pré-détection et post-détection : `flask taxref migrate-to-v17 apply-changes --script_predetection ./pre_detection.sql --script_postdetection ./post_detection.sql`
- Le fichier missing_cd_nom_into_database.csv est généré. Le charger dans le Drive dans un second onglet. Figer la ligne 1 et y ajouter des filtres.
    - La colonne `table_name` permet de filtrer sur la table :
        - `gn_synthese.synthese` les changements à y effectuer :
            - les lignes ayant la valeur "2" ou "3" dans la colonne `cd_raison_suppression` nécessitent de mettre à NULL le champ `cd_nom` dans la table `gn_synthese.synthese` à l'aide de requêtes SQL qui seront à ajouter dans le fichier ''pre_detection.sql''.
            - les lignes ayant la valeur "1" dans la colonne `cd_raison_suppression` nécessitent de remplacer la valeur du champ `cd_nom` dans la table `gn_synthese.synthese` à l'aide de requêtes SQL qui seront à ajouter dans le fichier ''pre_detection.sql''.
        - `taxonomie.bib_noms` les changements à y effectuer :
            - quelques soit les valeurs de la colonne `cd_raison_suppression`, il faudra supprimer les lignes correspondantes de la table `cor_nom_liste` et `bib_noms` à l'aide de requêtes SQL qui seront à ajouter dans le fichier ''pre_detection.sql''.
            - les lignes ayant la valeur "1" dans la colonne `cd_raison_suppression` nécessitent de remplacer la valeur du champ `cd_nom` dans la table `taxonomie.bib_noms` à l'aide de requêtes SQL qui seront à ajouter dans le fichier ''pre_detection.sql''. Si les requêtes de mise à jour pose problème vis à vis du maintient temporaire de l'ancienne version de TaxRef, il est possible d'ajouter les cd_nom correspondant à la liste des lignes à supprimer dans la table `cor_nom_liste` et des lignes à supprimer dans la table

`taxonomie.bib_noms`.

- Lancer à nouveau le script en essayant d'appliquer les changements `flask taxref migrate-to-v17 apply-changes`, normalement le script SQL `pre_detection.sql` devrait s'exécuter correctement mais le script `post_detection.sql` devrait générer des erreurs. En effet, ce script "post-détection" est exécuté avant d'avoir réaliser le changement de référentiel TaxRef, ainsi la réactivation des contraintes désactiver dans le script `pre_detection.sql` peuvent générer de nouvelles erreurs. Voir ticket #495. Il est donc nécessaire de modifier temporairement TaxRef pour qu'ils ressemblent à la future version sur laquelle nous cherchons à migrer.
- Si la base de données de TaxHub a évolué, il se peut que l'utilisation des scripts déclenche des erreurs. Il est alors nécessaire de rechercher dans les migrations Alembic de TaxHub le code SQL correspondant. Il est possible d'appliquer ce code à l'ajoutant au script "pre_detection.sql".
- Enfin, lorsque nous avons réussi à appliquer sans erreur tous les changements `flask taxref migrate-to-v17 apply-changes`, il devient possible de copier toutes les données dans une base de données GeoNature de la branche "develop" à l'aide du script Bash copy_all.sh.
    - Lors de l'exécution de ce script, de nouvelles erreurs peuvent être générée lors de la copie des données dans les tables `gn_synthese.synthese` et `taxonomie.bib_noms`. Ces erreurs sont dues à l'absence de certaines lignes du TaxRef de la version générée avec la branche "develop". Pourquoi ces lignes n'ont elles pas été supprimées dans la version migrée ? Il est nécessaire de supprimer les lignes correspondant à ces cd_noms de la table gn_synthese.synthese et des tables cor_nom_liste et bib_noms du schéma taxonomie.
- Dans le cas présent, migration des données SINP, il est également nécessaire de désactiver les triggers gérant la mise à jour des dates et de la sensibilité de la table synthese dans le script "pre_detection.sql" puis de les réactiver dans le script "post_detection.sql".

From:
https://sinp-wiki.cbn-alpin.fr/ - **CBNA SINP**

Permanent link:
https://sinp-wiki.cbn-alpin.fr/serveurs/installation/web-srv/geonature-taxref-upgrade?rev=1723470199

Last update: **2024/08/12 13:43**