

# Comment utiliser SQLInsérer ?

## Généralités

L'Insérer utilise le client mysql pour effectuer ses opérations. Il l'appelle en ligne de commande via la classe Process. L'Insérer est donc *lent*.

Pour pouvoir dialoguer avec le client MySQL, il remappe les flux entrée, sortie et erreur. Voir l'Object **Process** pour plus de détails.

Globalement il existe 2 fonctions :

- **insert()** qui exécute un script sql passé en paramètre
- **select()** qui exécute une commande SQL passée en paramètre, la commande en question étant de type SELECT.

Dans le cas du **select()**, on a donc un retour. Ce retour est mappé par une classe **ListFacade** qui encapsule une **List**.

## Configuration

On passe un objet **Properties** au constructeur du **Insérer**. Cet objet **Properties** doit contenir les entrées suivantes :

- `driverName` : le nom du driver (toujours `com.mysql.jdbc.Driver`)
- `url` : l'url de la base
- `user` : le login pour se connecter à la base
- `password` : le mot de passe
- `mysql` : le chemin complet du client mysql (ex : `C:/data/mysql-5.5.61-winx64/bin/mysql.exe`)
- `database` : la base à utiliser

## Exécuter un script

Typiquement, le script sert à créer ou dropper une base. Les fichiers doivent être directement à la racine du projet (source folder *test*, typiquement).

Ex : `insérer.insert("create.sql");`

En cas de problème, une exception est levée.

## Faire un SELECT

Il y a deux cas de figure :

Il n'y a qu'une entrée en retour

Dans ce cas, on peut directement utiliser ce retour.

ex :

```
int initNb = getInsérer().select("select count(id) from Equipe").getDataAsInt();
```

Dans ce cas, l'entête est enlevée de façon à n'avoir que la donnée.

Il y a plusieurs entrées (et surtout plusieurs lignes) en retour.

Dans ce cas, c'est plus compliqué. On peut utiliser **select()** en spécifiant qu'on veut l'entête. Dans ce cas, il faut récupérer une **List** de **Maps** via **ListFacade**, chaque entrée des **Maps** étant un des champs en retour.

Ex : `select("SELECT name,age FROM Machin",false)` (le *false* signifie qu'on garde l'entête)

On a une **List<Map<String,String>>** en retour avec comme clé des **Maps** *name* et *age*.

ATTENTION : si on fait `SELECT s.name,a.name FROM machin as s,bidule as a`, on aura pas *s.name* et *a.name* en retour. On aura que *name*, donc une aberration dans les **Maps**. Il faut donc faire quelque chose du genre :

```
SELECT s.name as sname,a.name as aname FROM machin as s,bidule as a
```

Et dans ce cas, les **Maps** auront comme clés, *sname* et *aname*.

Si on sait qu'on n'a qu'une seule ligne en retour, on peut récupérer cette ligne comme `String` (sans l'entête).

```
SELECT * FROM machin WHERE id=1 ;
```

Et donc

```
String ret = insérer.select("SELECT * FROM machin WHERE id=1").getDataAsString();
```

Dans ce cas, on sait que les différents champs sont séparés par des tabulations (`\t`). Il n'y a plus qu'à faire un `split()` et à récupérer ces valeurs.