Extensions serveur Gargantua

Module complémentaire scripting bases de données

Table des matières

Description du module

Mode d'emploi

Extensions Gargantua

Ce module ne comprend pas d’extensions.

Scripting

Le module extension serveur `siatel-plugin-scpdb` propose propose une API pour le scripting. L'objet qui comprend toutes les fonctions est nommé `SQL`.

SQL.setConnectionData(Object _connectionData)

```
public void setConnectionData(Object _connectionData)
```

Permet d'indiquer les données pour la connexion vers une base de données.

JS call:

```
SQL.setConnectionData(_connectionData)
```

Paramètres :

`_connectionData` - les données de connexion; un objet JavaScript contenant les clés `url`, `user`, `pass` et `autoCommit`

Retourne :

-

Exceptions :

–

Examples :

```
SQL.setConnectionData({
    'url'          : 'jdbc:postgresql://localhost/siatel.test',
    'user'         : 'siatel',
    'pass'         : 'siatel123',
    'autoCommit'   : true
}) ;
```

SQL.setConnectionData(url, user, pass, autoCommit)

```
public void SQL.setConnectionData(url, user, pass, autoCommit)
```

Permet d'indiquer les données pour la connexion vers une base de données.

JS call:

```
SQL.SQL.setConnectionData(url, user, pass, autoCommit)
```

Paramètres :

`url` - URL de connexion pour la base de données

`user` - nom de l'utilisateur pour la connexion

`pass` - mot de passe pour la connexion

`autoCommit` - paramètre autoCommikt pour la connexion

Retourne :

-

Exceptions :

–

Examples :

```
SQL.setConnectionData({
    'jdbc:postgresql://localhost/siatel.test',
    'siatel',
    'siatel123',
    true
}) ;
```

SQL.PUSHCONNECTIONDATA(_CONNECTIONDATA)

`public void pushConnectionData(Object _connectionData)`

Permet d'indiquer les données pour la connexion vers une base de données, tout en sauvegardant les données de connexion en cours d'utilisation. Voir aussi SQL.popConnectionData().

JS call:

`SQL.pushConnectionData(_connectionData)`

Paramètres :

`_connectionData` - les données de connexion; un objet JavaScript contenant les clés `url`, `user`, `pass` et `autoCommit`

Retourne :

-

Exceptions :

–

Examples :

```
SQL.pushConnectionData({
    'url'          : 'jdbc:postgresql://localhost/siatel.test',
    'user'         : 'siatel',
```

```
    'pass'        : 'siatel123',
    'autoCommit'  : true
}) ;
```

SQL.PUSHCONNECTIONDATA(URL, USER, PASS, AUTOCOMMIT)

`public void SQL.pushConnectionData(url, user, pass, autoCommit)`

Permet d'indiquer les données pour la connexion vers une base de données, tout en sauvegardant les données de connexion en cours d'utilisation. Voir aussi SQL.popConnectionData().

JS call:

`SQL.SQL.pushConnectionData(url, user, pass, autoCommit)`

Paramètres :

`url` - URL de connexion pour la base de données
`user` - nom de l'utilisateur pour la connexion
`pass` - mot de passe pour la connexion
`autoCommit` - paramètre autoCommikt pour la connexion

Retourne :

-

Exceptions :

–

Examples :

```
SQL.pushConnectionData({
    'jdbc:postgresql://localhost/siatel.test',
    'siatel',
    'siatel123',
    true
}) ;
```

SQL.POPCONNECTIONDATA()

`public void SQL.popConnectionData()`

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```

SQL.EXECUTEQUERY(SQUERY)

```
public void SQL.popConnectionData()
```

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```

SQL.executeQuery(sQuery, arrParams)

```
public void SQL.popConnectionData()
```

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```

SQL.executeQuery(sQuery, arrParams, arrResultData)

```
public void SQL.popConnectionData()
```

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```

SQL.prepareQuery(sQuery, markers)

```
public void SQL.popConnectionData()
```

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```

SQL.getAttrId(attribute)

```
public void SQL.popConnectionData()
```

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```

SQL.getAttrId(attribute, database)

```
public void SQL.popConnectionData()
```

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```

SQL.getAttrTableId(attribute)

```
public void SQL.popConnectionData()
```

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```

SQL.getAttrTableId(attribute, database)

```
public void SQL.popConnectionData()
```

Permet de restaurer un ensemble de paramètres de connexion sauvegardé precedamment par un appel pushConnecionData();Voir aussi SQL.pushConnectionData(_connectionData) et SQL.pushConnectionData(url, user, pass, autoCommit).

JS call:

```
SQL.SQL.popConnectionData()
```

Paramètres :

-

Retourne :

-

Exceptions :

–

Examples :

```
SQL.popConnectionData( ) ;
```