

# Utiliser PHP et PDO avec SQL Server

Date de dernière mise à jour : 24/03/2014 à 14:34

Source : <http://www.vulgarisation-informatique.com/pdo-php-mssql.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)

Pour faire fonctionner Microsoft SQL Server et PHP PDO, cela n'est pas forcément aussi simple qu'il n'y paraît. Les messages les plus fréquents que vous pouvez rencontrer sont les suivants : **Échec de l'ouverture de session de l'utilisateur** ou encore **could not find driver**.

Installation du driver PDO SQL Server

Pour Windows

Depuis PHP 5.3, l'extension `php_pdo_mssql` n'existe plus et vous devez passer par les bibliothèques fournies par Microsoft. Cliquez <a href="http://www.microsoft.com/en-us/download/details.aspx?id=20098">ici pour les télécharger. Cliquez sur **Download**, décompressez le fichier et copiez les bibliothèques (fichiers dont l'extension est **.dll**) dans le répertoire **bin/ext** de PHP.

Si vous utilisez **Wamp**, le répertoire est de cette forme : **C:\Program Files\wamp\bin\php\php5.X.X\ext**, avec X correspondant à la version de PHP que vous utilisez.

Les systèmes d'exploitation (OS) supportés par la version 3.0 du driver sont les suivantes :

- Windows Server 2008 R2 SP1
- Windows Vista SP2
- Windows Server 2008 SP2
- Windows 7 SP1
- Windows 8
- Windows 8.1

Les systèmes d'exploitation (OS) supportés par la version 2.0 du driver sont les suivantes :

- Windows Server 2003 Service Pack 1
- Windows XP Service Pack 3
- Windows Vista Service Pack 1 ou supérieur
- Windows Server 2008
- Windows Server 2008 R2
- Windows 7

Lorsque vous aurez copié ces DLL, ouvrez les deux fichiers **php.ini** (ils ne sont pas situés aux mêmes endroits : l'un se trouve dans **C:\Program Files\wamp\bin\apache\Apache2.4.4\bin** et correspond à votre fichier ini si vous utilisez PHP en ligne de commandes, on appelle cela le mode CGI, l'autre fichier `php.ini` se trouve dans le répertoire **C:\Program Files\wamp\bin\php\php5.4.16**).

Repérez les lignes qui commencent par **extension=** et rajoutez après toutes les extensions la ligne suivante :

```
extension=php_pdo_sqlsrv_54_ts.dll
```

Attention ! si vous utilisez PHP 5.3, remplacez le **54** par **53**, ce qui pourra donner :

```
extension=php_pdo_sqlsrv_53_ts.dll
```

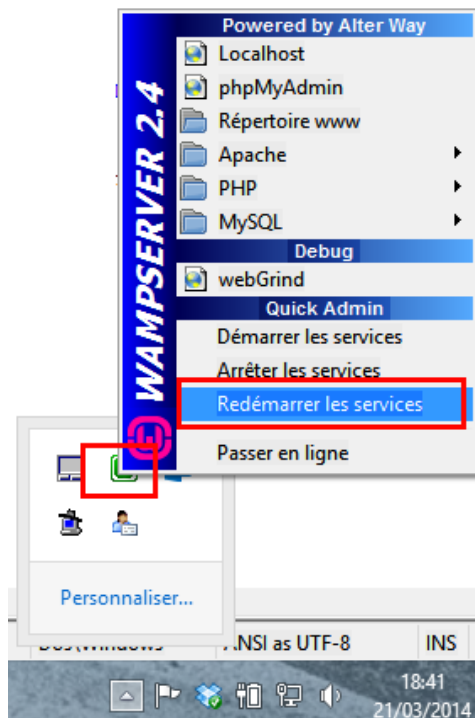
la mention **ts** signifie **thread safe**. En fonction de votre version d'Apache, vous aurez peut-être besoin d'utiliser la version **non thread safe** de la bibliothèque sql server, dans ce cas, la chaîne à utiliser est la suivante :

```
extension=php_pdo_sqlsrv_54.dll
```

... et avec PHP 5.3 :

extension=php\_pdo\_sqlsrv\_53.dll

Relancez Apache. Avec WAMP, cliquez sur l'icône **W** située à gauche de l'heure système (dans la zone appelée **Systray**) puis cliquez sur **Redémarrer les services**. Normalement, le W doit redevenir vert.



Pour Linux

Pour se connecter à Microsoft SQL Server sous Linux en PHP avec PDO, utilisez le paquet [PDO\\_DBLIB](#).

Vérifications

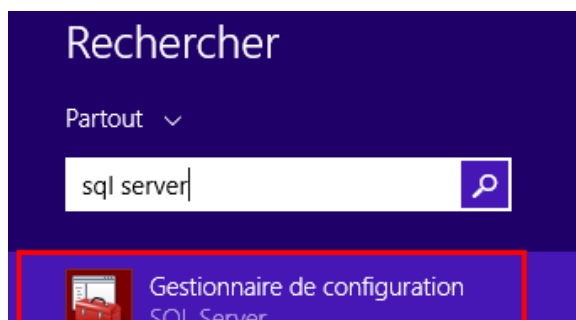
Créez une page **test.php** située à la racine de votre serveur web, et placez-y le code suivant :

```
<?php print_r
(
PDO
::
getAvailableDrivers
());
?>
```

Si l'extension PDO mssql s'est bien installée, vous devriez retrouver dans la liste des drivers disponibles un driver nommé **sqlsrv**. Ce sera notre driver à utiliser pour toute tentative de connexion avec PDO et SQL Server.

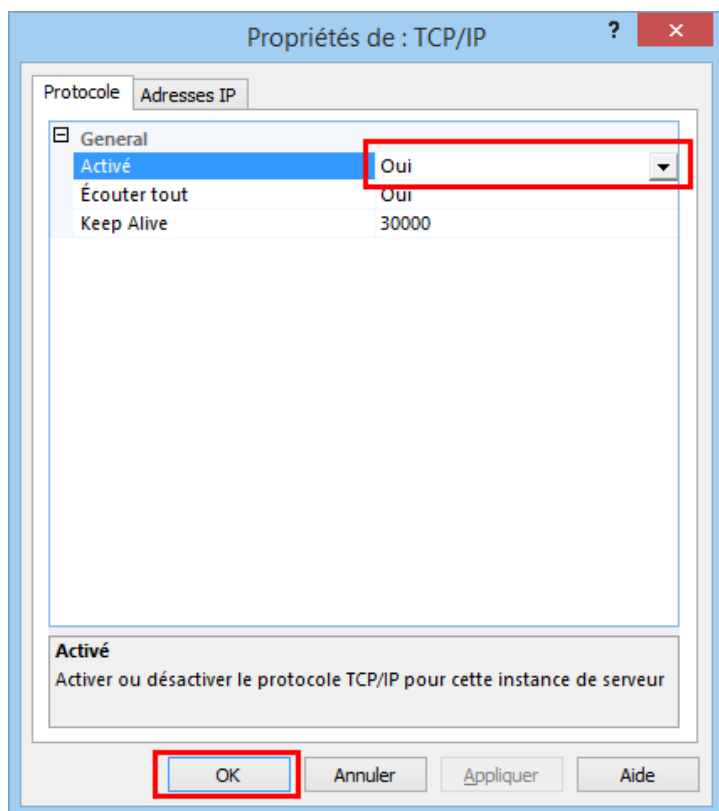
Installation de SQL Server

Pour ce tutoriel, nous utiliserons la version SQL Server 2012 Express, disponible gratuitement en téléchargement sur le [site de Microsoft](http://www.microsoft.com/fr-fr/download/details.aspx?id=29062). Installez SQL Server puis lancez-le en tapant dans Windows **8 Sql Server** puis en cliquant sur **Gestionnaire de configuration SQL Server** :



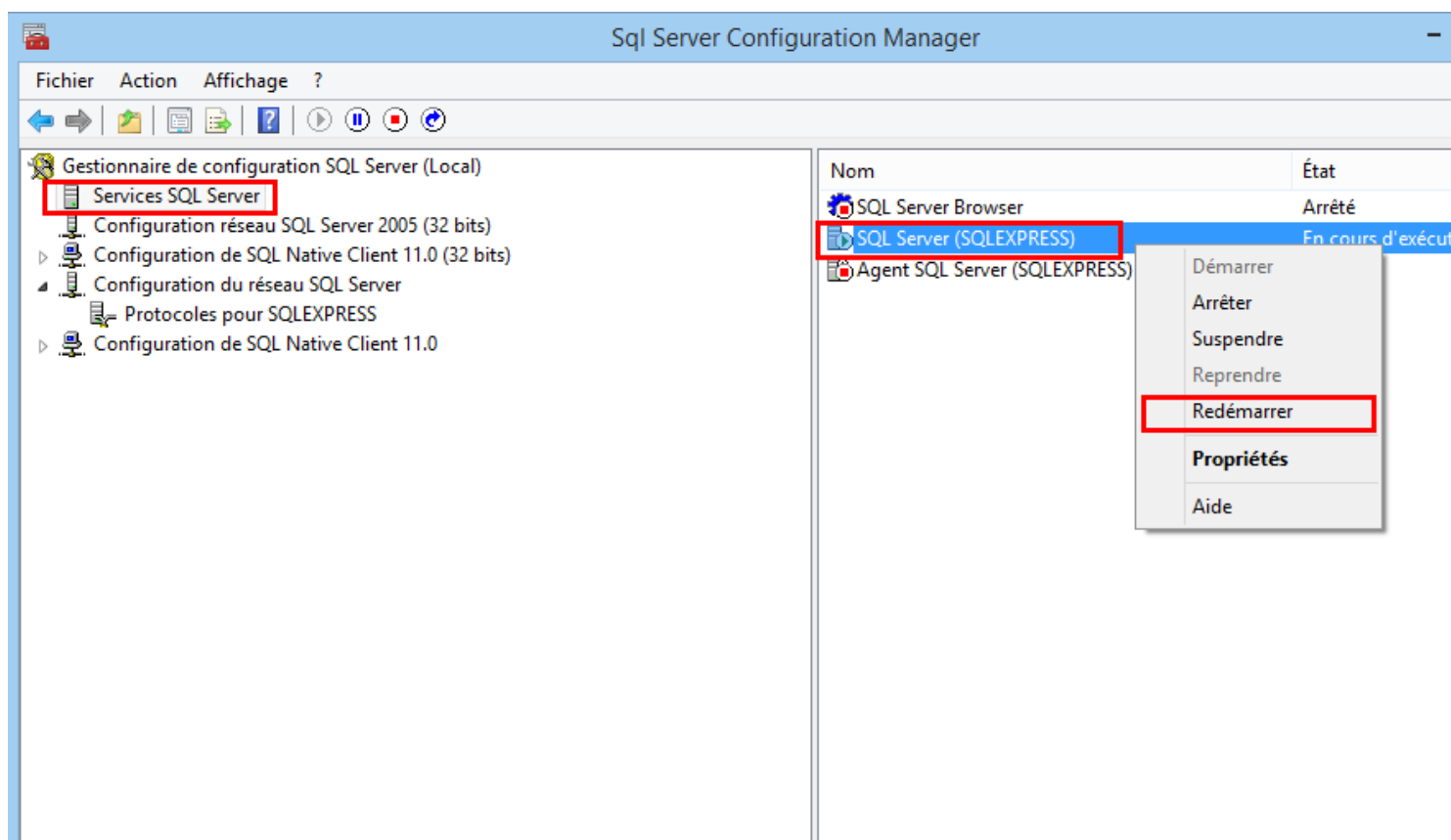
L'interface (plutôt rudimentaire) se lance. Déroulez **Configuration du réseau SQL Server** puis cliquez sur **Protocoles pour SQLEXPRESS** (le mot **SQLEXPRESS** peut varier, il correspond au nom de l'instance que vous avez choisie lors de l'installation de SQL Server).

Vous devez maintenant activer le protocole TCP-IP, les canaux nommés et la mémoire partagée. Pour ce faire, double-cliquez sur chacun des trois protocoles, puis, dans la zone **Général**, mettez **Oui** concernant l'activation.



Activation du protocole TCP-IP

Cliquez sur **Ok** à chaque fois pour valider. Lorsque vous aurez tout activé, cliquez sur **Services SQL Server**, puis sélectionnez **SQL Server (SQLEXPRESS)**. Cliquez avec le bouton droit de la souris dessus, puis sélectionnez **Redémarrer**.

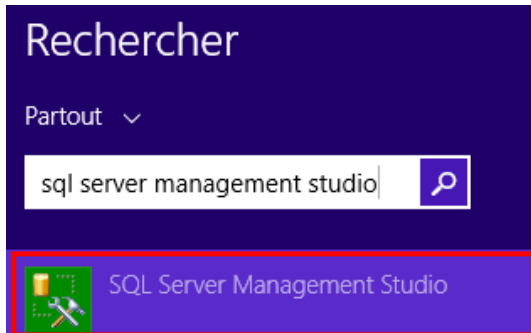


## Redémarrer SQL Server

A ce stade, il ne nous reste plus qu'à ajouter les différents utilisateurs qui auront le droit de se connecter via PHP.

## SQL Server Management studio

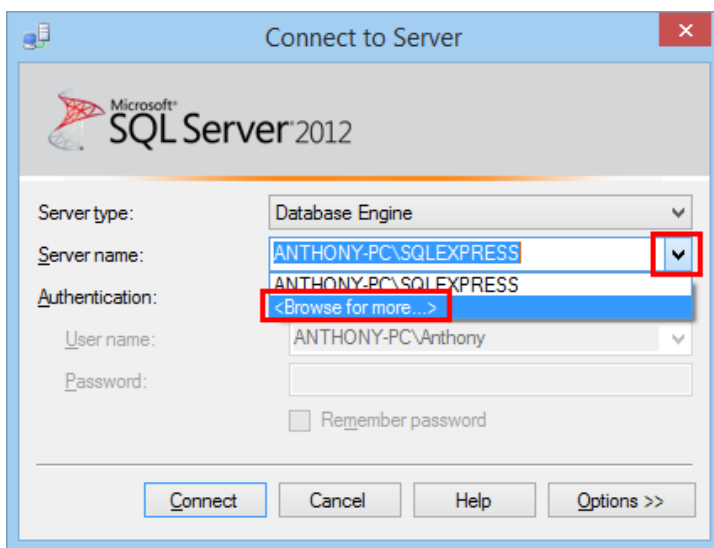
Pour administrer SQL Server, nous utiliserons un outil fourni gratuitement par Microsoft : Microsoft SQL Server Management Studio. Vous pouvez le télécharger [ici en version 64 bits](#) ou alors [ici en version 32 bits](#). Installez-le ensuite dans le répertoire de votre choix puis lancez-le en tapant dans l'interface Metro **SQL Server management studio** puis en cliquant dessus :



## SQL Server Management studio

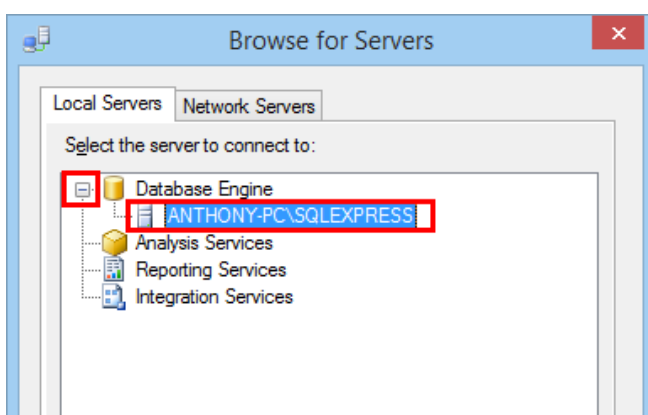
Vous pouvez aussi sur d'autres versions de Windows lancer l'outil via le menu démarrer, ou en cliquant sur les touches **+ R** puis en tapant **Ssms.exe**.

Au lancement de SMSS, vous arrivez directement sur une fenêtre de connexion. Cliquez sur la petite flèche située dans la zone **Server Name** puis cliquez sur **Browse for more** pour parcourir la liste des serveurs disponibles. Si votre serveur est déjà renseigné, sautez cette étape.



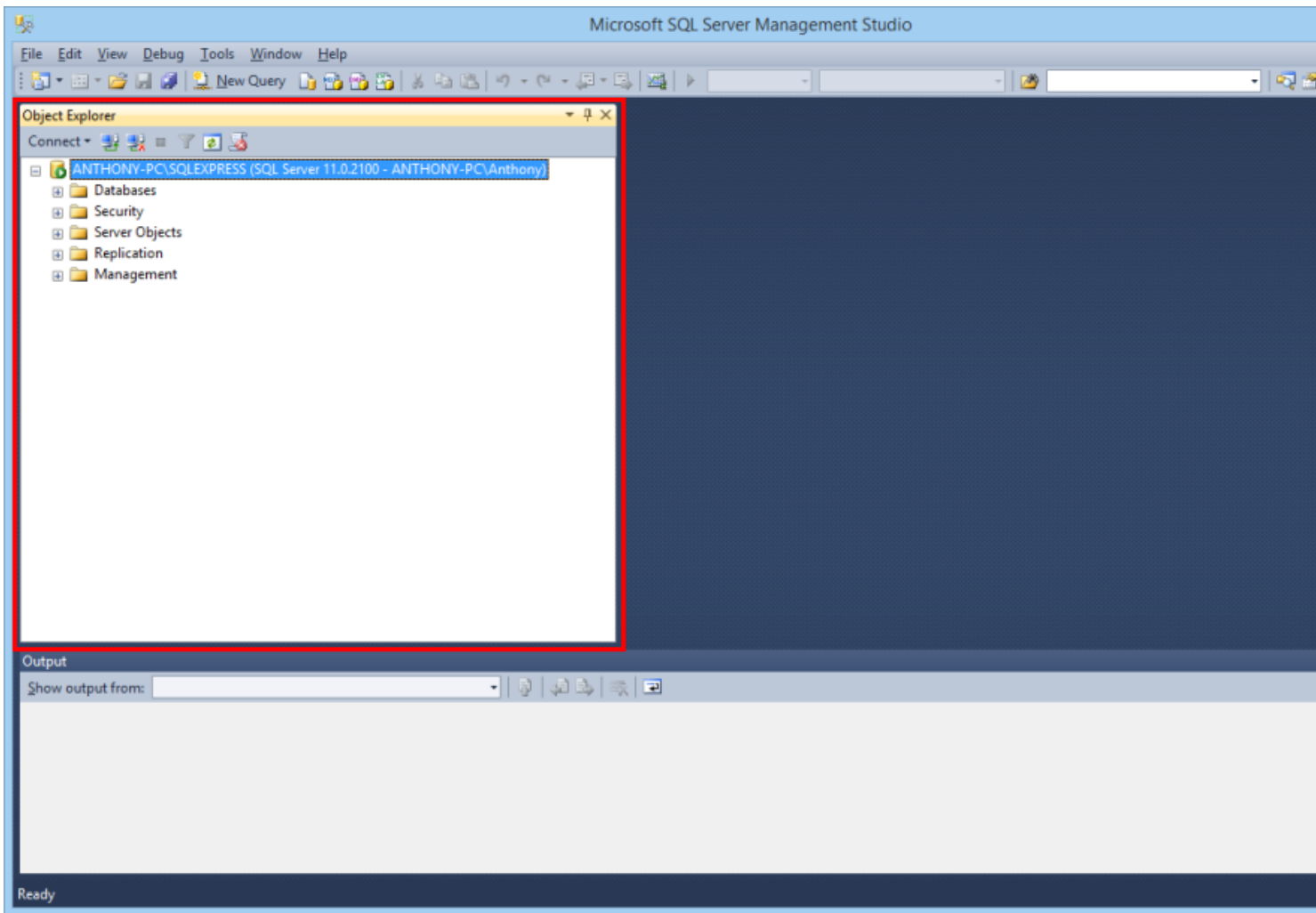
## Connexion à un serveur

Cliquez sur le petit **+** situé sur la ligne **Database Engine** puis sélectionnez votre instance locale. Validez en cliquant sur **Ok**.



Choix de l'instance

Cliquez ensuite sur le bouton **Connect**. Si tout s'est bien passé, aucune erreur ne s'affiche et votre instance se trouve à gauche dans la section **Object Explorer** de SQL Server Management Studio :

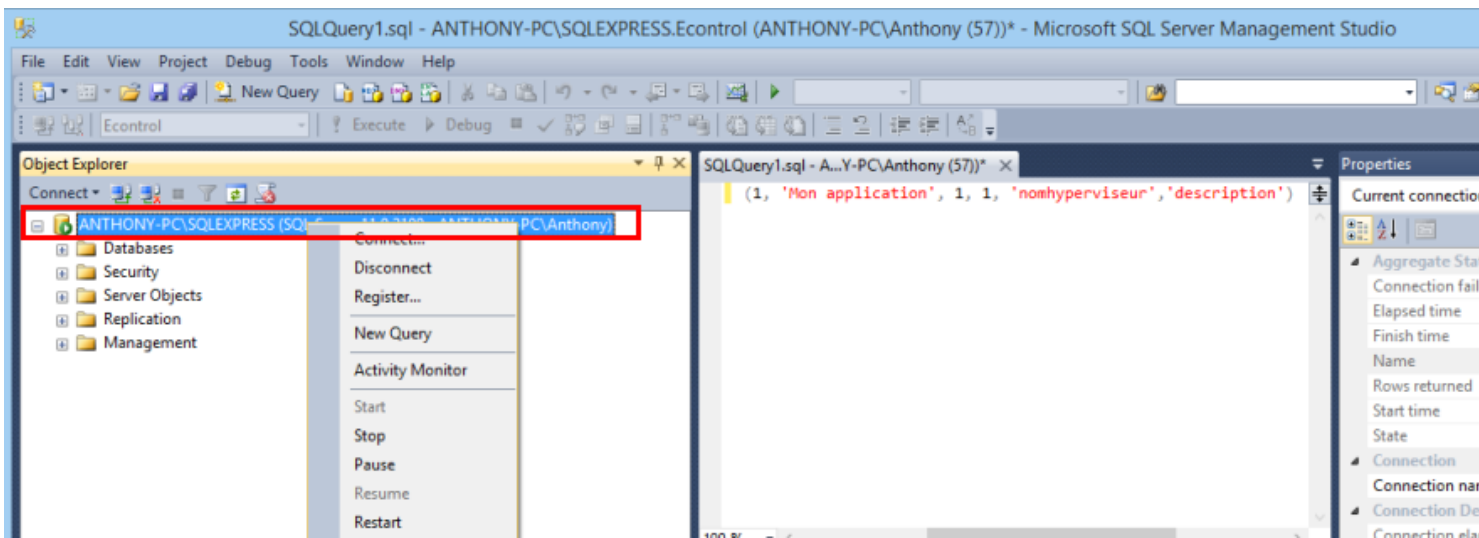


SQL Server Management studio - vue des objets disponibles

Autoriser l'authentification Windows et SQL Server

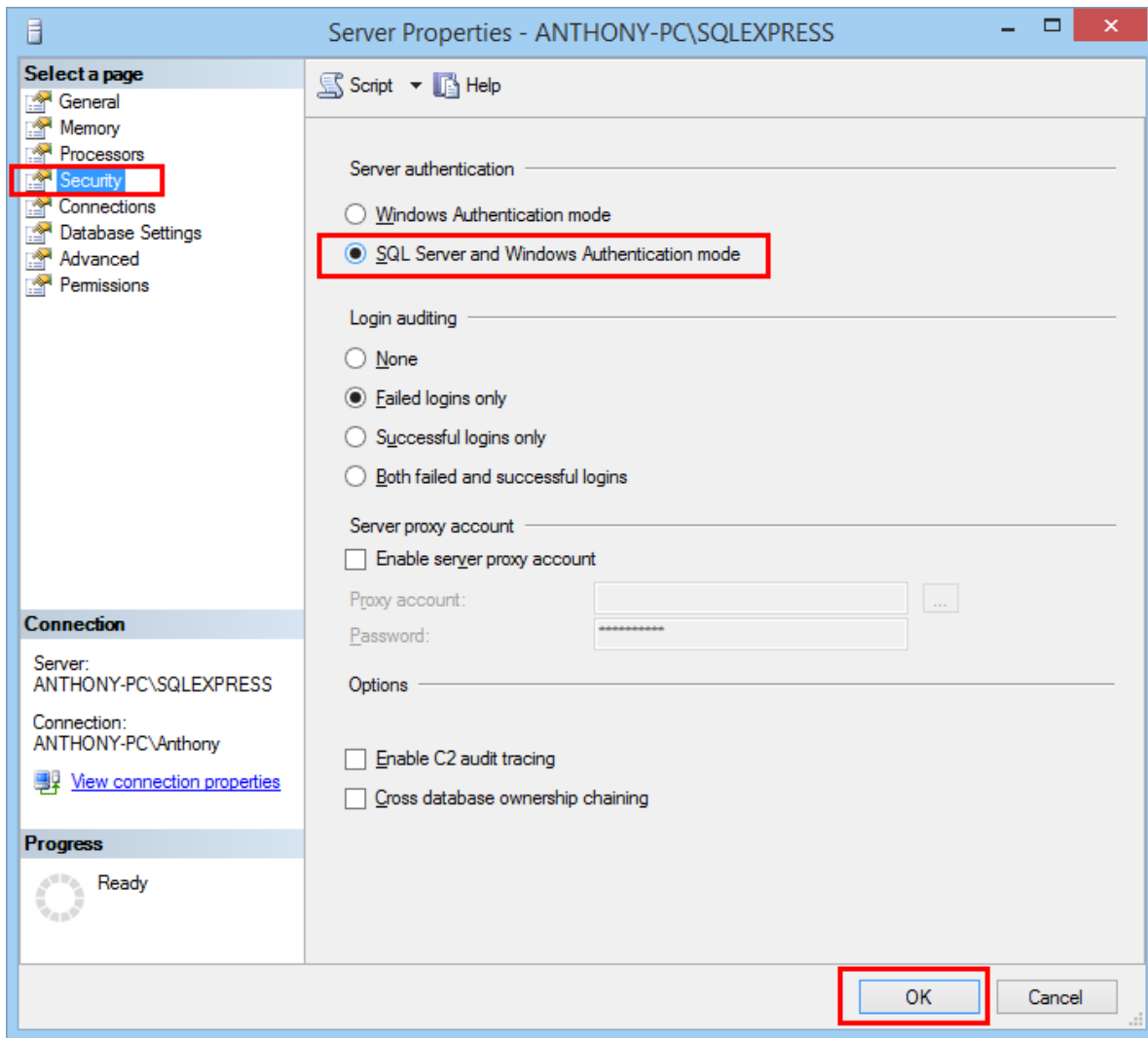
Par défaut, SQL Server ne permet pas de s'authentifier autrement que par un compte Windows. Pour pouvoir utiliser PDO avec PHP et un système de login/mot de passe classique, nous devons autoriser l'authentification de type SQL Server.

Pour ce faire, effectuez un clic avec le bouton droit sur l'instance de base de données puis cliquez sur **Properties** :



## Autorisations d'authentification

Cliquez à gauche sur la section **Security** puis, dans la zone intitulée **Server authentication**, choisissez l'option **SQL Server and Windows authentication mode**. Validez en cliquant sur **Ok**.



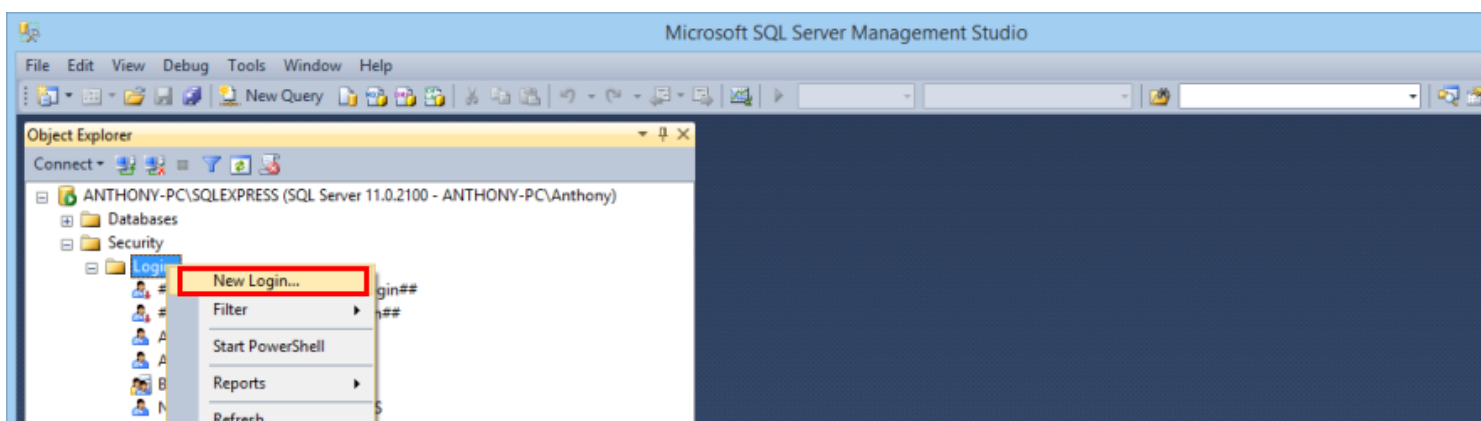
## Modes d'authentification autorisés

Maintenant que tous nos types d'utilisateurs peuvent se connecter, nous allons créer un utilisateur !

Créer un utilisateur PDO pour MS SQL

Vous devez vous trouver dans SQL Server Management Studio.

Dans votre instance (la mienne étant nommée **SQLEXPRESS** et située sur **ANTHONY-PC**, déroulez les dossiers **Security => Logins**. Faites un clic droit sur **Logins** puis cliquez sur **New login**.



## Création d'un nouvel utilisateur

Remplissez ensuite les cases suivantes :

**Login name** : ce sera le nom d'utilisateur (login) que vous utiliserez pour vous connecter avec PDO.

-Choisissez ensuite l'option **SQL Server authentication** : cela permettra de spécifier un login et un mot de passe pour vous connecter, l'authentification ne sera pas liée au compte utilisateur de PHP.

-Décochez la case **Enforce password policy**.

-Dans la zone **Default database** spécifiez la base de données par défaut sur laquelle vous souhaitez vous connecter (la mienne est nommée **Econtrol**). Si vous ne savez pas, utilisez la valeur **Master**.

-Choisissez **French** comme langage par défaut.

The screenshot shows the 'Login - New' dialog box with the following settings:

- Select a page:** User Mapping (highlighted in red)
- Login name:** anthony
- Authentication:** SQL Server authentication (selected, highlighted in red)
- Password:** [masked]
- Confirm password:** [masked]
- Enforce password policy:** [unchecked, highlighted in red]
- Default database:** Econtrol (highlighted in red)
- Default language:** French (highlighted in red)

## Paramètres de connexion de l'utilisateur

Notre utilisateur est presque créé. A ce stade, il ne peut encore se connecter à aucune base de données. Il va falloir effectuer un **mapping** permettant de lui autoriser l'accès à toutes ou certaines bases spécifiées.

Cliquez ensuite en haut à gauche sur **User Mapping**.

Cochez ensuite la ou les bases de données que l'utilisateur pourra voir ou contrôler :

The screenshot shows the 'Login - New' dialog box with the 'User Mapping' page selected in the left sidebar.

## Droits de l'utilisateur

Pour chacune des bases de données que vous cocherez dans la case **Users mapped to this login**, vous pourrez définir des droits dans la case **Database role membership for : [NOM DE LA BASE COCHÉE]**. Ces permissions sont les suivantes :

**db\_accessadmin**- : Les membres du rôle db\_accessadmin peuvent ajouter ou supprimer l'accès des connexions Windows, des groupes Windows et des connexions SQL Server, à la base de données.

**db\_backupoperator**- : Les membres du rôle db\_backupoperator peuvent sauvegarder la base de données.

**db\_datareader**- : Les membres du rôle db\_datareader peuvent exécuter une instruction SELECT sur une table ou une vue de la base de données.

**db\_datawriter**- : Les membres du rôle db\_datawriter peuvent ajouter, supprimer et modifier des données dans toutes les tables utilisateur.

**db\_ddladmin**- : Les membres du rôle db\_ddladmin peuvent exécuter n'importe quelle commande DDL (Data Definition Language) dans une base de données.

**db\_denydatareader**- : Les membres du rôle db\_denydatareader ne peuvent lire aucune donnée des tables utilisateur d'une base de données.

**db\_denydatawriter**- : Les membres du rôle db\_denydatawriter ne peuvent ajouter, modifier ou supprimer aucune donnée des tables utilisateur d'une base de données.

**db\_owner**- : Les membres du rôle db\_owner peuvent réaliser toutes les activités de configuration et de maintenance sur la base de données.

**db\_securityadmin**- : Les membres du rôle db\_securityadmin peuvent modifier l'appartenance au rôle et gérer les autorisations.

Cochez les cases correspondant aux permissions choisies, puis cliquez sur **Ok**. L'utilisateur a été créé. Vous pouvez le voir dans la section générale **Security => Logins** ainsi que dans **Databases** et pour chacune des bases de données, dans la section **Security => Users**.

Vous pouvez maintenant vous connecter en PHP avec PDO à votre base de données SQL Server.

## Connexion avec PDO

Nous utiliserons un code très simple qui utilise la classe de base PDO, fournie avec PHP.

```
<?php
class
sql
{
    private
$connexion_sql
;
    function
__construct
()
{
$this
->
connexion_bdd
= new
PDO
(
'sqlsrv:Server=ANTHONY-PC\SQLEXPRESS;Database=econtrol'
,
'anthony'
,
'anthony'
);
// Fixe les options d'erreur (ici nous utiliserons les exceptions)
$this
->
connexion_bdd
->
setAttribute
```

```

(
PDO
::
ATTR_ERRMODE
,
PDO
::
ERRMODE_EXCEPTION
); } public function
requete
(
$requete
) {
$prepare
=
$this
->
connexion_bdd
->
prepare
(
$requete
);
$prepare
->
execute
(); return
$prepare
; } }
$sql
= new
sql
();
?>

```

Remplacez les valeurs que vous pouvez trouver dans ce code par vos valeurs, et exécutez le code [PHP](#) dans votre navigateur Web. Aucune erreur ne devrait apparaître.

Ajoutez maintenant la logique suivante, en remplaçant le nom de ma table **applications** par une table qui existe dans votre base de données :

```

<?php
class
sql
{ private
$connexion_sql
; function
__construct
() {
$this
->
connexion_bdd
= new

```

```

PDO
(
'sqlsrv:Server=ANTHONY-PC\SQLEXPRESS;Database=econtrol'
,
'anthony'
,
'anthony'
);
// Fixe les options d'erreur (ici nous utiliserons les exceptions)
$this
->
connexion_bdd
->
setAttribute
(
PDO
::
ATTR_ERRMODE
,
PDO
::
ERRMODE_EXCEPTION
); } public function
requete
(
$requete
) {
$prepare
=
$this
->
connexion_bdd
->
prepare
(
$requete
);
$prepare
->
execute
(); return
$prepare
; } }
$sql
= new
sql
();
$req
=
$sql
->
requete
(

```

```
'SELECT * FROM applications'  
); while (  
$r  
=  
$req  
->  
fetch  
()) {  
print_r  
(  
$r  
); }  
?>
```

Source : <http://www.vulgarisation-informatique.com/pdo-php-mssql.php>.

Distribution interdite sans accord écrit d'Anthony ROSSETTO (<http://www.vulgarisation-informatique.com/contact.php>)