

<http://vader-fr.fr/spip.php?article180>



Fonctions pour PDO

- Développement web - Les pages dynamiques avec PHP/MySQL - Exemples de fonctions PHP -



Publication date: dimanche 1er juillet 2018

Copyright © Vader[FR] : ce n'est pas un blog, c'est un Sith - Tous droits

réservés

PDO est une couche d'abstraction en PHP permettant d'adresser une base de données indépendamment de son type (MySQL, PostGres, Oracle...)

Par sécurité, il est préférable de faire des requêtes paramétrées afin d'éviter l'injection SQL.

Le contrecoup, c'est le rajout de code qui devient plus long à écrire.

Le code étant toujours le même, il est toutefois possible de le simplifier en créant une fonction simple qu'il suffira d'appeler.

Cette première fonction permet de préparer et exécuter une requête, laquelle peut être paramétrée, et dont le résultat pourra être :

- plusieurs lignes, traitées ensuite dans une boucle foreach
- un nombre de ligne modifié

Elle effectue donc un prepare, puis bindValue pour 0-n paramètres, puis un execute, et renvoie l'objet requête PDO. les paramètres de la requête sont fournis dans un tableau qui commence à 1 (de préférence) ou zéro la requête doit donc avoir " ? " en paramètre(s) afin de pouvoir les affecter depuis un tableau par indice, plus facilement traitable. voir l'exemple plus bas

```
function prepexec($rek,$params,$db){
    if (isset($params[0])){
        $debut=0;
        $correc=1;
        $fin=sizeof($params)-1;
    }else{
        $debut=1;
        $correc=0;
        $fin=sizeof($params);
    }
    $rk=$db->prepare($rek);
    // si tableau param vide, pas de bindValue => fonction unique prepare avec/sans param + exec
    for ($i=$debut;$i<=$fin;$i++){
        $typ_param=PDO::PARAM_STR; // PAR DEFALT
        $param=$params[($i+$correc)];
        if (preg_match("/[\w]+/", $param)){ $typ_param=PDO::PARAM_STR; } // PDO::PARAM_STR
        if (preg_match("/^[+-]?\d+$/", $param)){ $typ_param=PDO::PARAM_INT; } // PDO::PARAM_INT
        // PDO::PARAM_BOOL - MySQL might not support PARAM_BOOL
        if (preg_match("/^$/", $param)){ $typ_param=PDO::PARAM_NULL; } // PDO::PARAM_NULL
        $rk->bindValue($i,$param,$typ_param);
    }
    // renvoie la requête préparée et exécutée
    $rk->execute();
    // $nb=$rk->execute(); OU foreach ($rk as $result){
    // donc $nb=prepexec(... ou $rk=prepexec puis foreach($rk... ou même foreach(prepexec...
    return $rk;
}
```

exemple d'appel :

```
$rek="select count(login) as existe from utilisateur where login=?";
$params[1]=$user;
$rk=prepexec($rek,$params,$db);
$existe=0;
foreach ($rk as $result){
    $existe=$result['existe'];
}
```

Cependant, cela est encore trop long.

Cette deuxième fonction va donc faire le boulot en appelant la première et en renvoyant la valeur attendue, unique, et dont le nom du champ est indiqué.

```
function simplexec($rek,$params,$db,$nomvar){
    $resultat=null;
    $rk=prepexec($rek,$params,$db);
    foreach ($rk as $result){
        $resultat=$result["$nomvar"];
    }
    return $resultat;
}
```