

Qualité, conception, modélisation

Accéder aux données - PHP

JY Martin

Novembre 2023

Plan

1 Mécanisme général

2 PHP & ORM

Se connecter à partir de PHP

- API de connexion dépendant du SGBD (obsolète depuis PHP 7)
 - fonctions spécifiques de manipulation de la base de données
 - utilisation d'un driver ODBCA éviter
- Classe PDO
Fortement conseillée

PHP - PDO

La communication passe par un objet de la classe **PDO**
Le type de base de données est précisé dans le **protocole de connexion**

- La création d'un objet PDO crée une connexion à la base de données
conn = new PDO('protocole', 'login', 'password');
- Les méthodes de la classe PDO permettent de faire des requêtes et de récupérer les résultats
- la destruction de l'objet clos la connexion

D'un type de SGBD à l'autre, la seule chose qui change est le protocole de communication.

PHP - PDO

Le protocole : `protocole:host=myHost;dbname=myDatabase`

- `protocole` est le protocole utilisé
mysql, pgsql, oci, ...
- `myHost` est le nom du serveur
- `myDatabase` est le nom de la base de données

Exemple :

```
$conn = new PDO( "pgsql:host=localhost;dbname=test" , $user, $pass);
```

PHP - PDO

Les méthodes classiques

- **prepare** définit une requête SQL à exécuter
On peut définir des paramètres de la façon suivante :
 - `?` : valeur du paramètre indiquée dans execute
 - `:nomParam` : valeur du paramètre indiquée dans bindParam
- **bindParam** : définition des valeurs paramètres d'une requête
- **execute** exécute une requête (définie avec prepare). execute permet de définir les valeurs des paramètres.
- **fetchAll** récupère dans un tableau les lignes résultats
- **closeCursor** clos la requête SQL et récupère l'espace mémoire.
- ...

PHP - PDO

Exemple :

```
$conn = new PDO('pgsql:host=localhost;dbname=test', 'login', 'password');  
$query = $conn->prepare("SELECT * FROM Personne WHERE Personne_ID=?");  
$query->setFetchMode ( PDO::FETCH_ASSOC );  
$query->execute(array(25));  
$rows = $query->fetchAll();  
foreach ( $rows as $aRow ) {  
    echo $aRow["nom"]." ".$aRow["prenom"]."<br/ >\n";  
}  
$query->closeCursor();
```

PHP - PDO

- new PDO... : permet de créer l'objet PDO. Etablit la requête à la base.
- prepare : définit la requête. Les ? sont les paramètres. Notez bien l'ordre d'apparition.
- setFetchMode : définit la façon de récupérer les résultats
- execute : exécute la requête. Le tableau passé en paramètre définit les valeurs des paramètres de la requête.
- fetchAll : récupère les résultats de la requête et les place dans un tableau.

Plan

1 Mécanisme général

2 PHP & ORM

PHP - ORM

Object Relationnal Mapping :

- Introduisent une couche d'abstraction entre la base de données et le langage de programmation
- Permet de s'abstraire du type de SGBD qui est alors géré par l'ORM

Exemples :

- Propel
- Doctrine
- Zend

PHP - Doctrine

Utilise une description de chacune des tables à l'aide d'un fichier PHP. Chacun de ces fichiers est une ****Entity****.

3 façons de travailler

- De la base vers PHP
La base est utilisée pour construire les fichiers Entity
- Des fichiers Entity vers la base
Les fichiers PHP sont utilisés pour créer la base de données
- Fichier de description YAML
Un fichier YAML permet de générer les Entity, puis la base de données.

PHP - Frameworks

S'appuient généralement sur un ORM et fournissent un certain nombre d'outils au dessus de l'ORM pour faciliter la programmation.

Exemples :

- Symfony (utilise doctrine ou propel)
- Codeigniter (qui a son propre ORM)
- cakePHP (qui a son propre ORM)

