**PDO : PHP Data Object**

C’est la méthode la plus récente permettant de connecter vos pages contenant du php avec une base de données.

**I Création d’une connexion**

1. **Création d’une classe**

Comme il s’agit d’un langage objet, il faut créer une classe

****

…………………………………………………………………………………………………………………………………………………………….........................

Rq1 : Attention à la syntaxe !!

Rq2 : Il faut parfois rajouter en dernier paramètre array( PDO::MYSQL\_ATTR\_INIT\_COMMAND => "SET NAMES utf8")

afin qu’il n’y ait pas de problème d’encodage

1. **Système de gestion des erreurs**

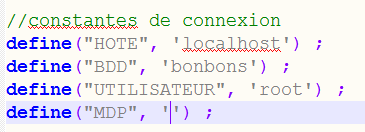
****

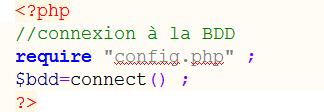
Rq : Cela permet de comprendre l’erreur mais en production, on ne concatènera pas $e->getMessage() pour que l’utilisateur n’ait pas d’information sur notre connexion.

1. **Pour prendre les bonnes habitudes**

* Mieux vaut créer une fonction de connexion dans un fichier séparé : config.php

Ce fichier contiendra la fonction de connexion ainsi que les constantes de connexion

* 
* Fonction de connexion
* Il ne reste plus qu’à faire un include du fichier contenant la fonction de connexion et à se connecter dans chaque page PHP nécessitant d’accéder à la BDD.



**II Réalisation de requêtes**

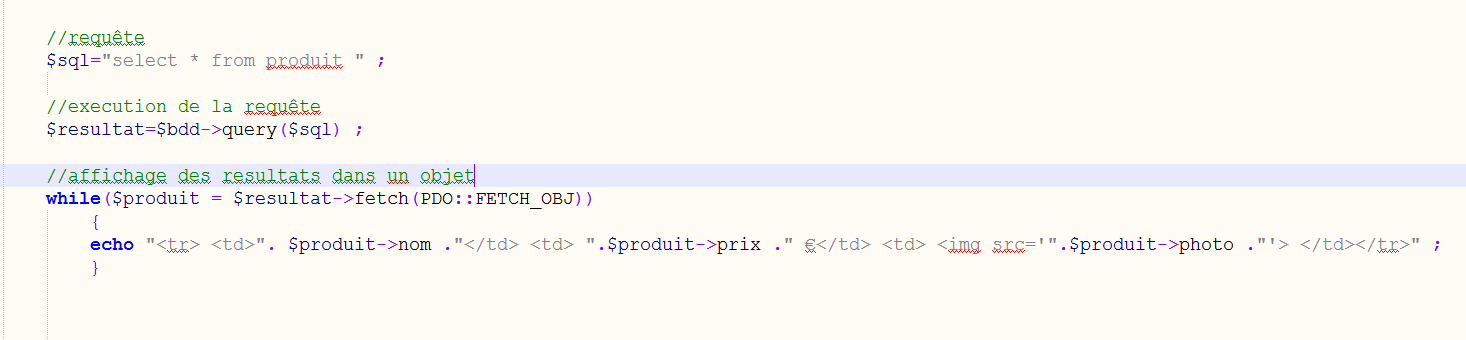
PDO sépare les requêtes en 2 familles :

* celles qui renvoient un résultat : LID
* celles qui agissent sur la BDD : LMD

1. **Les requêtes « select » (interrogation de données)**

* Création de la requête
* Exécution de la requête avec la méthode query
* Récupération du résultat pour affichage
  + Avec une boucle (si le résultat comporte plusieurs lignes)
  + Dans un objet ou dans un tableau associatif

Voir ci-après l’exemple avec un objet :



***En cas d’erreur***

Par défaut, PDO reste silencieux sur les erreurs dans les requêtes. Si par exemple, on met un S au nom de la table, l’erreur ne se produit qu’en ligne ………………….

Il faut donc explicitement préciser dans la fonction de connexion qu’on souhaite un système de gestion d’erreurs, le meilleur étant :

$connect->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION) ;

On va ensuite ajouter un try / catch au moment de la réalisation de notre requête.

1. **Les requêtes de manipulation**

Il faut bien évidemment se connecter à la BDD puis réaliser les étapes suivantes :

* Création de la requête
* Exécution de la requête avec la méthode exec
* On obtient le nombre de lignes affectées par la requête.

Rq : Pensez au try/catch

Ex : modification du prix des dragibus



Application :

1. Créer une page d’accueil avec 4 onglets : voir les produits / Ajouter / Modifier / Supprimer
2. Les faire fonctionner

**III Les requêtes préparées**

Elles permettent de lutter efficacement contre les injections-SQL.

* Avec un drapeau :

$sql=$bdd->prepare("select \* from produit where nom=**?** ");

$sql-> execute(array(‘dragibus’)) ;

* Avec des paramètres nommés :

$sql=$bdd->prepare("select \* from produit where nom=**:nom**") ;

$sql-> execute(array(

'id'=>’dragibus’

) ;

* Avec bind param :

$nom=’dragibus’ ;

$sql=$bdd->prepare("select \* from produit where nom= **:nom** ") ;

$sql->bindParam(‘nom’, $nom, PDO ::PARAM\_STR) ;

$sql-> execute() ;

Rq : Pensez au try/catch