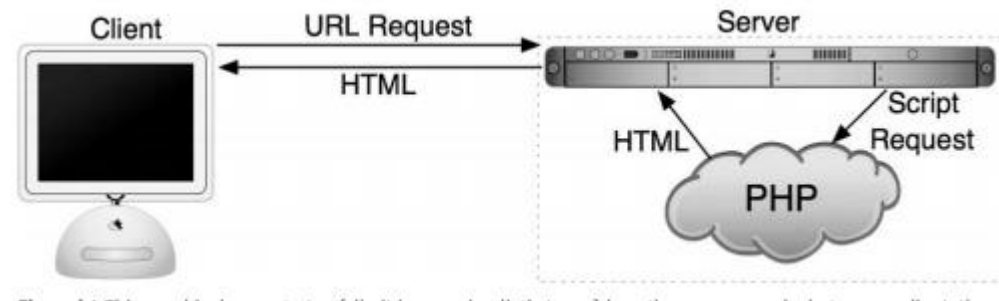


PHP & MySQL



Dr. Yoro DIA

Site Web dynamique

- Un site Web dynamique est un site Web dont le contenu change automatiquement.
- Autrement dit, le contenu visible sur la page n'est pas toujours intégré directement en HTML par le Webmaster, mais peut être généré à la volée suite à des conditions particulières comme l'historique de navigation, la localisation géographique du visiteur, les privilèges accordés à celui-ci...
- Cependant, le contenu des sites web dynamiques provient souvent de sources de données. Ces sources peuvent être des fichiers texte, fichiers XML, sources Json, et par dessous tout, il y a **les bases de données**.

Site Web dynamique

- En effet, les bases de données constituent les sources de données les plus populaires pour différentes raisons. On peut citer:
 - Les bases de données (dites relationnelles) sont structurées sous formes de tables. Chacune des tables contient des données jugées cohérentes, mais complémentées par les données qui figurent dans d'autres tables via des relations.
 - Les bases de données sont faciles à parcourir et interroger, et c'est principalement grâce au langage de requêtes SQL (Structured Query Language).

Bases de données et PHP

- PHP est connu pour sa capacité à interagir aisément avec les bases de données.
- On entend par les bases de données pratiquement tous les SGBD (Systèmes de Gestion de Bases de Données) populaires (MySQL, PostgreSQL, MariaDB, Oracle, MSSQL...).
- Cependant, le SGBD **MySQL** reste le plus largement utilisé avec **PHP**.

Bases de données et PHP

- Afin d'interagir avec une base de données MySQL, PHP propose trois méthodes différentes:
 - Extension MySQL originale
 - Extension **MySQLi**
 - Objet PDO (PHP Data Objects)

Bases de données et PHP

- **Extension MySQL originale:**
 - Il s'agit de l'extension MySQL native au PHP.
 - Pendant longtemps elle a été pratiquement la seule extension disponible, mais elle est rendue obsolète depuis la **version 5.5** de PHP, et a été supprimée en **PHP7**.

Bases de données et PHP

- **Extension MySQLi:**

- Il s'agit d'une extension un peu plus moderne.
- C'est d'ailleurs cette extension que l'on va traiter dans cette section.

Bases de données et PHP

- **Objet PDO (PHP Data Objects)**

- L'objet PDO constitue de loin la meilleure solution pour interagir avec une base de données en PHP.
- Elle dispose de plusieurs avantages, et le plus marquant c'est qu'il permet l'abstraction de la base de données, ce qui a pour effet de pouvoir communiquer avec de nombreux SGBD (et pas que MySQL) tout en maintenant le même code PHP.

Extension MySQLi

- **Pré-requis pour coder en PHP/MySQL**

- Il est évident que pour éditer du code PHP qui peut communiquer avec une base de données (en l'occurrence MySQL), il faut être à l'aise avec le **langage de requêtes SQL**.

Extension MySQLi

- **Pré-requis pour coder en PHP/MySQL**

- SQL est un langage qui permet de communiquer avec les bases de données (dites bases de données SQL).
- Il s'agit d'un langage très riche en fonctionnalités qui permettent de mener des opérations plus ou moins complexes sur les bases de données.
- Cependant, dans cette section on va se contenter des requêtes **CRUD (Create, Read, Update, Delete)** qui constituent les requêtes les plus utilisées pour créer un **site Web dynamique** ou application Web.

- **Séance de TP: Langage SQL**

- **A suivre...**

Opérations CRUD à l'aide de MySQLi

- **Création de la base de données**

- Afin de créer la base de données sur le serveur, ainsi que les tables et aussi créer les utilisateurs en leur accordant les privilèges nécessaires, il est préférable de se servir de l'outil **PHPMyAdmin** qui propose une interface graphique pour gérer les bases de données MySQL.
- Cependant, ces opérations peuvent être faites sur la ligne de commande si vous êtes à l'aise avec le langage SQL.
- Dans notre cas, on va supposer que l'on va créer une base de données du nom de **mabase** et l'utilisateur **root** sans mot de passe .

Opérations CRUD à l'aide de MySQLi

- Dans notre base de données on va créer une table du nom de **personnes** dont le code SQL est le suivant:

```
CREATE TABLE personnes
(
    id Integer PRIMARY KEY AUTO_INCREMENT,
    nom varchar(255) NOT NULL,
    prenom varchar(255),
    login varchar(255),
    pass varchar(255),
    date timestamp NOT NULL default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP
);
```

Opérations CRUD à l'aide de MySQLi

- **Connexion au serveur et sélection d'une base de données**

- La première opération à faire si on veut interroger une base de données c'est la connexion au serveur qui héberge celle-ci à l'aide du code suivant:

```
$con=mysqli_connect("localhost","root","");
```

Opérations CRUD à l'aide de MySQLi

- **Connexion au serveur et sélection d'une base de données**
 - La fonction `mysqli_connect()` permet de connecter le serveur de base de données avec l'utilisateur **root** qui a un mot de passe **vide** (respectivement deuxième et troisième arguments).
 - Dans notre cas, le serveur de base de données est le serveur local **localhost** (premier argument).

Opérations CRUD à l'aide de MySQLi

- **Connexion au serveur et sélection d'une base de données**
 - La fonction `mysqli_connect()` retourne un identifiant de connexion (`$con` dans notre cas) ou la valeur booléenne `false` en cas d'échec.
 - Une fois la connexion au serveur établie, on peut désormais sélectionner la base de données à interroger via le code suivant:

```
mysqli_select_db($con,"mabase");
```

Opérations CRUD à l'aide de MySQLi

- **Connexion au serveur et sélection d'une base de données**
 - La fonction `mysqli_select_db()` retourne un résultat booléen (**true** en cas de succès et **false** en cas d'échec) et accepte **deux arguments** qui sont respectivement l'identifiant de la connexion au serveur et le nom de la base de données.
 - Cependant, on peut résumer les deux opérations que l'on vient de voir dans une seule instruction de la manière suivante:

```
$con=mysqli_connect("localhost","root","","mabase");
```

Opérations CRUD à l'aide de MySQLi

- **Execution des opérations CRUD**

- Afin d'exécuter une requête SQL via un script PHP on fait appel à la fonction `mysqli_query()`.
- Par exemple, si on veut insérer un nouvel enregistrement dans la table **personnes** on va exécuter le code suivant:

```
$req=mysqli_query($con,"insert into personnes (nom,prenom,login,pass)
values('Diallo','Fatima','Fdiallo',DialloFatima123");
```

Opérations CRUD à l'aide de MySQLi

```
1  <?php
2
3  $con = mysqli_connect("localhost","root","");
4      if (!$con)
5      {
6          die("ne pourrait pas se relier : " . mysqli_connect_error());
7      }
8
9      mysqli_select_db($con,"mabase");
10
11     $req ="INSERT INTO personnes (id, nom, prenom, login, pass)
12             VALUES
13             ('', 'DialLo', 'Fatima', 'Fdiallo', 'DialloFatima123')";
14
15     if (!mysqli_query($con, $req))
16     {
17         die('impossible d'ajouter cet enregistrement : ' . mysqli_connect_error());
18     }
19     echo "L'enregistrement est ajouté ";
20
21     mysqli_close($con)
22
23  ?>
```

Opérations CRUD à l'aide de MySQLi

- **Execution des opérations CRUD**

- La fonction `mysqli_query()` accepte deux arguments qui sont, respectivement, l'identifiant de la connexion précédemment établie et la requête SQL à exécuter.
- La fonction `mysqli_query()` retourne un identifiant de connexion (`$req` dans notre cas) ou `false` en cas d'échec.

Opérations CRUD à l'aide de MySQLi

- Si on veut modifier l'enregistrement que l'on vient d'insérer (on va changer le mot de passe par exemple) alors on va exécuter l'instruction suivante:

```
$req=mysqli_query($con, "UPDATE personnes SET pass='fatima123'  
where id=1");
```

Opérations CRUD à l'aide de MySQLi

```
1  <?php
2
3  $con = mysqli_connect("localhost","root","");
4      if (!$con)
5      {
6          die("ne pourrait pas se relier : ". mysqli_connect_error());
7      }
8
9      mysqli_select_db($con,"mabase");
10
11     $req ="UPDATE personnes SET pass='fatimal23'     WHERE id=1";
12     .....
13     .....
14     if (!mysqli_query($con, $req))
15     {
16         die('impossible d'effectuer la modification : ' . mysqli_connect_error());
17     }
18     echo "La modification est effectuee ";
19
20     mysqli_close($con)
21
22  ?>
```

Opérations CRUD à l'aide de MySQLi

- Bien entendu, on a supposé que le champ **id** associé à cet enregistrement vaut **1**.

Si maintenant on veut supprimer cet enregistrement alors on exécute le code PHP suivant:

```
$req=mysqli_query($con,"DELETE FROM personnes WHERE id=1");
```

Opérations CRUD à l'aide de MySQLi

```
1 <?php
2
3 $con = mysqli_connect("localhost","root","");
4     if (!$con)
5     {
6         die("ne pourrait pas se relier : ". mysqli_connect_error());
7     }
8
9     mysqli_select_db($con, "mabase");
10
11     $req = "DELETE FROM personnes WHERE id=1";
12
13     .....
14     if (!mysqli_query($con, $req))
15     {
16         die('impossible de supprimer enregistrement : ' . mysqli_connect_error());
17     }
18     echo "Enregistrement est bien supprimé";
19
20     mysqli_close($con)
21
22 ?>
```

Opérations CRUD à l'aide de MySQLi

- **Exécution de la requête de sélection**

- Bien que la sélection fait aussi partie des opérations CRUD, elle demande quand même un peu plus de traitement.
- Alors, supposons que l'on veut afficher les noms et prénoms de tous les enregistrements qui figurent dans la table **personnes** (s'il y'en a).

Opérations CRUD à l'aide de MySQLi

- **Dans ce cas on exécute les instructions suivantes:**

```
$req=mysqli_query($con,"SELECT nom, prenom FROM personnes");  
while($stab=mysqli_fetch_assoc($req)){  
    echo $stab["nom"]." ".$stab["prenom"];  
}
```

Opérations CRUD à l'aide de MySQLi

- Une fois la requête de sélection exécutée, on prévoit la boucle **while** qui sera exécutée autant de fois qu'il y a des enregistrements dans la table.
- La fonction **mysqli_fetch_assoc()** qui accepte comme argument l'identifiant de la requête (**\$req**) retourne l'enregistrement courant sous forme d'un tableau PHP (**\$tab**), ou **false** s'il n'y a plus aucun enregistrement, ce qui a pour effet d'interrompre la boucle **while**.

Opérations CRUD à l'aide de MySQLi

```
1  <?php
2  $con = mysqli_connect("localhost","root","");
3      if (!$con)
4      {
5          die("ne pourrait pas se relier : ". mysqli_connect_error());
6      }
7      mysqli_select_db($con,"mabase");
8
9      $req=mysqli_query($con,"SELECT nom, prenom FROM personnes");
10     while($stab=mysqli_fetch_assoc($req)){
11         echo $stab["prenom"] . " " . $stab["nom"] . "<br>" ;
12     }
13  ?>
```

Opérations CRUD à l'aide de MySQLi

- Le tableau PHP **\$tab** contient autant d'indexes que de champs sélectionnés. Dans ce cas, il contient deux indexes qui correspondent respectivement au **nom** et au **prénom**.
- Si on veut connaître le nombre de résultats retournés par la requête de sélection on fait appel à la fonction **mysqli_num_rows()** comme ceci:

```
$nbr_resultats=mysqli_num_rows($req);
```

Opérations CRUD à l'aide de MySQLi

```
1  <?php
2  $con = mysqli_connect("localhost","root","");
3      if (!$con)
4      {
5          ..... die("ne pourrait pas se relier : ". mysqli_connect_error());
6      }
7      mysqli_select_db($con,"mabase");
8
9      $req=mysqli_query($con,"SELECT nom, prenom FROM personnes");
10     while($stab=mysqli_fetch_assoc($req)){
11         ..... echo $stab["prenom"] . " " . $stab["nom"] . "<br>" ;
12     }
13
14     $nbr_resultats=mysqli_num_rows($req);
15     echo "Le nombre total enregistrements trouvés est: $nbr_resultats";
16     ?>
```

- A suivre...

