3 - Manipuler les données

Introduction

Introduire la notion de Modélisation d'un problème

Ce que vous allez apprendre dans ce cours :

- Identifier les différents types de serveurs web
- Acquérir une bonne connaissance de l'architecture client/serveur
- Maitriser l'environnement de développement

Vous pouvez télécharger le résumé théorique complet du module de compétences **"Développer des sites web dynamiques"** sur le volet droit de votre écran.

Écrire des scripts d'accès aux données

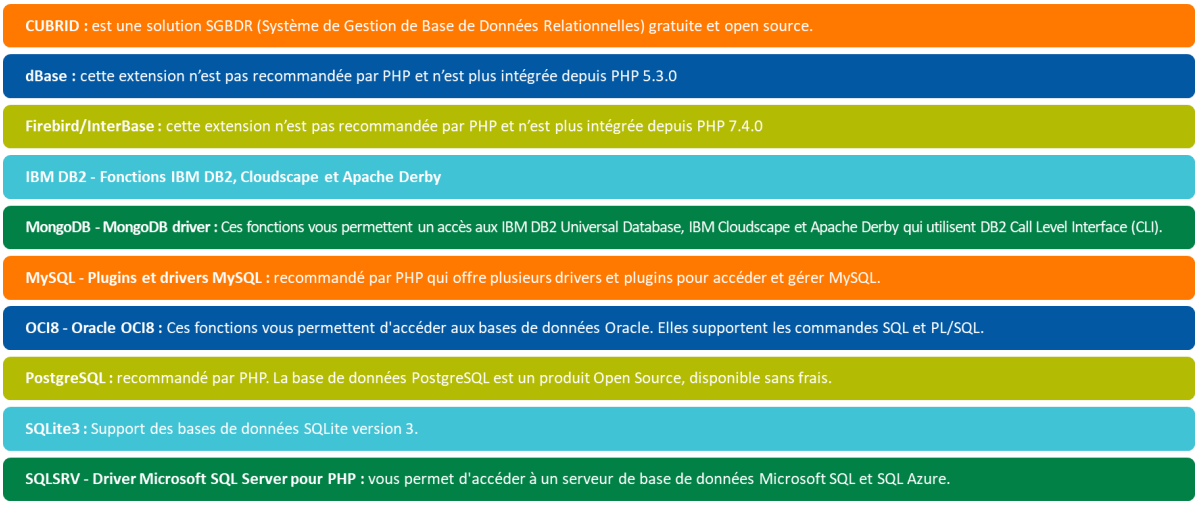
1.Connexion à une base de données MySQL avec PDO

Introduction

- Pour enregistrer les données en PHP vous pouvez utiliser :
  - Les variables
  - Les fichiers
  - Une base de données

- Une base de données (BDD) permet de stocker, organiser et analyser les données.
- Un système de gestion de base de données (SGBD) permet d'accéder aux bases de données.
- Une couche d'abstraction de base de données est une interface de programmation d'applications qui unifie la communication entre une application informatique et des bases de données.
- L'API : est une solution informatique qui permet à des applications de communiquer avec d'autres applications et de s'échanger mutuellement des services ou des données sans connaitre les détails de leur mise en œuvre.
- Une DLL : Une Dynamic Link Library est une bibliothèque logicielle dont les fonctions sont chargées en mémoire par un programme, au besoin, lors de son exécution, par opposition aux bibliothèques logicielles statiques ou partagées dont les fonctions sont chargées en mémoire avant le début de l'exécution du programme. (source Wikipédia)

Les bases de données



MySQL

Avant de créer une connexion à un serveur de base de données MySQL, vous devez avoir :

- Un serveur de base de données MySQL installé sur votre système local ou sur un serveur distant.
- Une base de données sur le serveur MySQL.
- Un compte MySQL avec un nom d'utilisateur et un mot de passe permettant d'accéder à la base de données.

Les plateformes de développement Web PHP comme EasyPHP, LAMP, WAMP, MAMP, LEMP et XAMPP fournissent MySQL.

Fonctions MySQL :

- mysql_connect — Ouvre une connexion à un serveur MySQL
- mysql_create_db — Crée une base de données MySQL
- mysql_ping — Vérifie la connexion au serveur MySQL, et s'y reconnecte au besoin
- mysql_query — Envoie une requête à un serveur MySQL

Liste complète des fonctions MySQL en PHP : https://www.php.net/manual/fr/ref.mysql.php

Se connecter à MySQL en PHP

Pour pouvoir manipuler nos bases de données MySQL en PHP (sans passer par phpMyAdmin par exemple). PHP met à notre disposition deux API (Application Programming Interface) :

- L'extension MySQLi
- L'extension PDO (PHP Data Objects)

La performance globale des deux extensions MySQLi et PDO peut être considérée comme identique. Nous allons choisir dans ce qui suit du cours l'interface PDO

Interface d'abstraction pour le langage PHP :

- DBA
- ODBC - ODBC (Unifié)
- PDO

Installation PDO

Choisissez les autres fichiers DLL spécifiques à votre base de données et utilisez soit la fonction dl() pour les charger au moment de l'exécution ou activez-les dans le fichier php.ini en-dessous de la ligne php_pdo.dll.

Fonction dl :

- Définition : Charge une extension PHP à la volée
- Syntaxe : dl(string $extension_filename) : bool

Fonctions PDO

- PDO::beginTransaction → Démarre une transaction
- PDO::commit → Valide une transaction
- PDO::__construct → Crée une instance PDO qui représente une connexion à la base
- PDO::errorCode → Retourne le SQLSTATE associé avec la dernière opération sur la base de données
- PDO::errorInfo → Retourne les informations associées à l'erreur lors de la dernière opération sur la base de données
- PDO::exec → Exécute une requête SQL et retourne le nombre de lignes affectées
- PDO::getAttribute → Récupère un attribut d'une connexion à une base de données
- PDO::getAvailableDrivers → Retourne la liste des pilotes PDO disponibles
- PDO::inTransaction → Vérifie si nous sommes dans une transaction
- PDO::lastInsertId → Retourne l'identifiant de la dernière ligne insérée ou la valeur d'une séquence
- PDO::prepare → Prépare une requête à l'exécution et retourne un objet
- PDO::query → Prépare et Exécute une requête SQL sans marque substitutive
- PDO::quote → Protège une chaîne pour l'utiliser dans une requête SQL PDO
- PDO::rollBack → Annule une transaction
- PDO::setAttribute → Configure un attribut PDO

La classe PDOStatement

PDOStatement : Représente une requête préparée et, une fois exécutée, le jeu de résultats associé.

- Méthodes : https://www.php.net/manual/fr/class.pdostatement.php

La classe PDOException

PDOException : Représente une erreur émise par PDO.

- Propriétés :
  - errorInfo
  - errorcode
- PDO::errorInfo
  - Syntaxe : public PDO::errorInfo(): array

- PDO::errorCode
  - Syntaxe : public PDO::errorCode(): ?string

Configuration

Le pilote PDO_MYSQL implémente l'interface de PDO pour autoriser l'accès de PHP aux bases de données MySQL. Pour déclarer le pilote du SGBD MySQL, il faut ajouter la ligne suivante dans le fichier php.ini : extension=php_pdo_mysql.dll

Nom	Défaut	Modifiable	Historique
pdo.dsn.*		*php.ini* seulement	

Tab.1 : Options de configuration PDO. Source : https://www.php.net/manual/fr/pdo.configuration.php

Nom	Défaut	Modifiable
pdo_mysql.default_socket	"/tmp/mysql.sock"	PHP_INI_SYSTEM
pdo_mysql.debug	NULL	PHP_INI_SYSTEM

Tab.2 : Options de configuration du driver PDO_MYSQL. Source : https://www.php.net/manual/fr/ref.pdo-mysql.php

SQL

SQL (Structured Query Language) est un langage de programmation standardisé utilisé pour gérer les bases de données au sein des SGBD et plus particulièrement les SGBD relationnelles (ex: MySQL), permettant d'effectuer des opérations sur les données qu'elles contiennent.

Les instructions SQL couvrent 4 domaines :

- Langage de définition de données (LDD) : permet de créer ou supprimer des objets dans la base de données.
- Langage de manipulation de données (LMD) : permet de manipuler les données contenues dans les tables.
- Langage de contrôle de données (LCD) : permet de gérer les utilisateurs d'une base de données ainsi que leurs droits sur les objets.
- Langage de contrôle des transactions (LCT) : permet de valider ou annuler des modifications de données dans la base de données.

- Liste des commandes SQL : https://sql.sh/

2.Interrogation d'une base de données à travers un formulaire

Paramétrer la connexion

Pour établir la connexion, il faut créer un DSN (Data Source Name) en renseignant des informations comme par exemple:

- Le type de base de données qui est MySQL.
- Le nom de la base de données.
- Le port de connexion.
- L'encodage.

Connexion au serveur MySQL

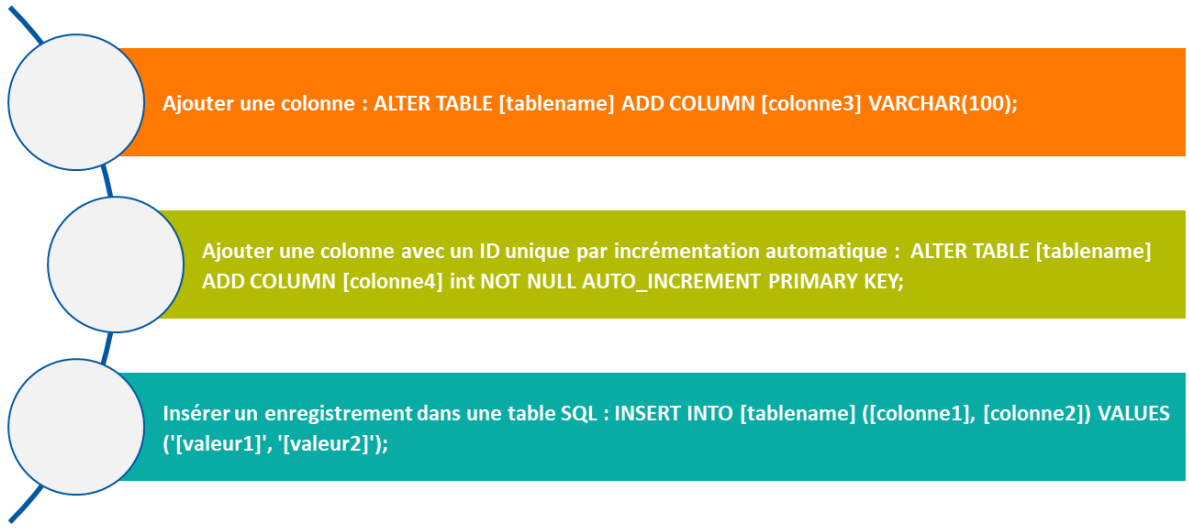
Pour connecter PHP à MySQL avec PDO, il est obligatoire de fournir les quatre renseignements suivants:

- Le nom d'hôte
- La base de données
- L'identifiant (login)
- Le mot de passe

Connexion avec gestion des erreurs

- Pour afficher des détails sur l'erreur, il faut activer les erreurs lors de la connexion à la base de données via PDO.
- Activer les erreurs lors de la connexion à la base de données permettra d'avoir un message d'erreur détaillé.

Requêtes SQL



Formulaire HTML

En HTML, il existe principalement deux méthodes pour interagir avec un utilisateur :

- Les liens (balise <a>).
- Les formulaires (balise <form>).

Page traitement PHP

- La page PHP permet de collecter des données à partir du formulaire.
- La valeur de l'attribut « action » dans la balise de formulaire HTML signifie que toutes les valeurs des champs d'entrée seront envoyées au fichier .php présent dans la valeur de l'attribut « action ».

Stockage des données dans MySql

Prérequis : une table « utilisateurs » créée dans la base de donnée MySQL.

3.Récupération des résultats

Requêtes SQL

Constantes PDO

- PDO::FETCH_ASSOC (int)
- PDO::FETCH_NAMED (int)
- PDO::FETCH_COLUMN (int)

Liste des constantes prédéfinies : https://www.php.net/manual/fr/pdo.constants.php

4.Manipulation des données CRUD

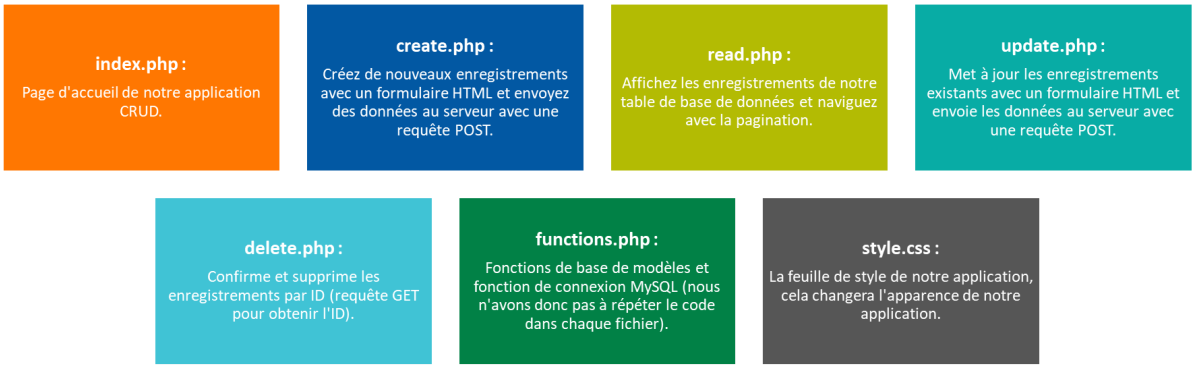
CRUD

CRUD est un acronyme des noms des quatre opérations de base de la gestion de la persistance des données et applications :

- Create (créer)
- Read ou Retrieve (lire)
- Update (mettre à jour)
- Delete ou Destroy (supprimer)

Requêtes SQL

Exemple organisation des fichiers PHP



Sécuriser les données

1.Utilisation des sessions et des cookies

- Pour transmettre des variables de page en page, on peut utiliser :
- Les divers champs des formulaires.
- Passer les variables directement à travers les liens.
- Utiliser les cookies.
- Utiliser les sessions.

Quand une session est créée sur le serveur, ce dernier envoie son identifiant (unique) au client sous forme d'un cookie.

Cookies

- Les cookies sont un mécanisme d'enregistrement d'informations sur le client, et de lecture de ces informations.
- Les cookies font partie des en-têtes HTTP, ce qui impose que les fonctions doivent être appelées avant tout affichage de texte.

La fonction setcookie()

Il faut appeler cette fonction avant toute balise <html> ou <head> et aussi des caractères d'espacement blanc.

Une fois que les cookies ont été placés, ils seront accessibles lors du prochain chargement de page dans le tableau $_COOKIE. Les valeurs des cookies peuvent aussi exister dans la variable $_REQUEST.

Syntaxe : setcookie(string $name, string $value = "", int $expires_or_options = 0, string $path = "", string $domain = "", bool $secure = false, bool $httponly = false, array $options = []) : bool

name : le nom du cookie

value : Cette valeur est stockée sur l'ordinateur du client.

expires_or_options : Le temps après lequel le cookie expire, c'est un timestamp Unix. Exemple : time()+60*60 fera expirer le cookie dans 1heure. Si vous ne spécifiez pas ce paramètre ou s'il vaut 0, le cookie expirera à la fin de la session (lorsque le navigateur sera fermé).

Path : Le chemin sur le serveur sur lequel le cookie sera disponible.

Domain : Le (sous-)domaine pour lequel le cookie est disponible.

Secure : Indique si le cookie doit uniquement être transmis à travers une connexion sécurisée HTTPS depuis le client. Lorsque ce paramètre vaut true, le cookie ne sera envoyé que si la connexion est sécurisée.

Httponly : Lorsque ce paramètre vaut true, le cookie ne sera accessible que par le protocole HTTP. Cela signifie que le cookie ne sera pas accessible via des langages de scripts, comme Javascript.

Options : Un tableau associatif qui peut avoir comme clés expires, path, domain, secure, httponly et samesite.

La fonction setrawcookie()

Envoie un cookie sans encoder sa valeur en URL.

Syntaxe : setrawcookie (string $name, string $value = ?, int $expires_or_options = 0, string $path = ?, string $domain = ?, bool $secure = false, bool $httponly = false, array $options = []) : bool

setrawcookie() est identique à setcookie() excepté que la valeur du cookie ne sera pas automatiquement encodée URL lors de l'envoi au navigateur.

Les sessions : Introduction

- Le support des sessions de PHP est un moyen de préserver des données entre plusieurs accès.
- Le support des sessions vous permet de stocker des données entre les requêtes dans le tableau super-globale $_SESSION.
- ...

Les sessions : Configuration à l'exécution

Le comportement de ces fonctions est affecté par la configuration dans le fichier php.ini.

- Options de configuration pour les sessions : Liste des options de configuration : https://www.php.net/manual/fr/session.configuration.php

Les sessions : Fonctions

Les sessions : Constantes prédéfinies

Ces constantes sont définies par cette extension, et ne sont disponibles que si cette extension a été compilée avec PHP, ou bien chargée au moment de l'exécution.

- SID (string) : Constante contenant le nom de la session et l'identifiant en cours, sous la forme "name=ID" ou une chaîne vide si l'identifiant de session a été défini dans un cookie de session. C'est la même valeur que celle retournée par la fonction session_id().
- PHP_SESSION_DISABLED (int) : Valeur retournée par session_status() si la session est désactivée.
- PHP_SESSION_NONE (int) : Valeur retournée par session_status() si la session est activée, mais que la session n'existe pas.
- PHP_SESSION_ACTIVE (int) : Valeur retournée par session_status() si la session est activée, et que la session existe.

Les sessions : Classes et Interfaces

- SessionHandler
- SessionHandlerInterface
- SessionIdInterface
- SessionUpdateTimestampHandlerInterface

2.Sécurisation des données

Introduction

- La sécurité d'un serveur web PHP est fragile puisqu'il permet d'accéder aux fichiers, d'exécuter des commandes, et d'ouvrir des connexions réseaux.
- La common gateway interface (CGI) est une interface de serveurs Web qui permet un échange de données normalisé entre des applications et des serveurs externes. Ainsi, les pages HTML ne sont pas entièrement disponible sur le serveur HTTP qui exécute un autre programme, puis retourne le contenu généré.
- Utiliser PHP comme un exécutable CGI est une possibilité pour les cas où l'on ne veut pas l'utiliser comme un module du serveur web (comme Apache), ou bien lorsque l'on souhaite l'utiliser en combinaison avec un gestionnaire CGI complémentaire, afin de créer un environnement de script sécurisé.
- Lorsque PHP est utilisé en tant que module Apache, celui-ci hérite des permissions accordées à l'utilisateur faisant tourner Apache ce qui peut impacter la sécurité et les autorisations.
- Une erreur de sécurité fréquente est de donner à l'utilisateur Apache les droits de superadministrateur ("root"), ou d'accroitre les possibilités d'Apache d'une quelconque autre façon.
- PHP, est constamment testé et amélioré. Chaque nouvelle version rassemble des modifications majeures ou mineures, aussi bien pour renforcer la sécurité, que pour réparer des problèmes de conception et de configuration, ainsi que d'autres points qui peuvent affecter la sécurité et la stabilité globale de votre système.

Sécurité des sessions

- Pour protéger les utilisateurs d'une tactique simple, la directive session.use_only_cookies doit être activée. Dans ce cas, les cookies doivent être activés obligatoirement côté client sinon les sessions ne fonctionneront pas.
- Implémenter SSL/TLS sur le serveur et le rendre obligatoire pour les utilisateurs. HSTS devrait être utilisé pour améliorer également la sécurité.
- ...

Sécurisation des configurations INI de session

- session.cookie_lifetime=0
- session.use_cookies=On et session.use_only_cookies=On

- ...

Sécurité des fichiers

Puisque PHP a été fait pour permettre aux utilisateurs d'accéder aux fichiers, il est possible de créer un script PHP qui vous permet de lire des fichiers tels que /etc/password, de modifier les connexions ethernet, lancer des impressions de documents, etc.

Sécurité des bases de données

Pour lire ou stocker des informations, vous devez vous connecter au serveur de bases de données, envoyer une requête valide, lire le résultat et refermer la connexion. De nos jours, le langage le plus courant pour ce type de communication est le langage SQL. Cela signifie qu'une requête SQL est capable de contourner les contrôles et vérifications, comme les identifications, et parfois, les requêtes SQL ont accès aux commandes d'administration.

Sécurité des données transmises par les internautes

Il est vivement recommandé d'examiner minutieusement votre code pour vous assurer qu'il n'y a pas de variable envoyée par le client web qui ne soit pas suffisamment vérifiée avant utilisation.

Sécurité de rapport d'erreurs

Une tactique d'attaque standard consiste à faire faire des erreurs au système, et à analyser les types des erreurs qui sont retournées, ainsi que leur contexte. Cela permet à l'attaquant d'obtenir des informations à propos du serveur, en vue de détecter de possibles faiblesses.

Il est particulièrement dangereux d'exécuter du code de sources connues avec des gestionnaires de débogage inclus, ou de travailler avec des techniques de débogage répandues.

Exemples d'attaques

Injection SQL : La solution à cela est d'utiliser des requêtes SQL paramétrées et des objets de données PHP (PDO) pour différencier les données des parties de requêtes.

Traversée du répertoire : Il faut appliquer les autorisations appropriées en fonction du statut de l'utilisateur.

Scripts intersites (attaques XSS) : Parmi les solutions filtrer toutes les données externes ainsi que d'utiliser les fonctions existantes PHP.

Falsification de demandes intersites : Il est recommandé d'utiliser HTTPS pour crypter la connexion. L'utilisation de POST au lieu de GET ainsi qu'un mécanisme d'authentification dans les formulaires.

Stockage des mots de passe : Il existe deux façons importantes de stocker des mots de passe en toute sécurité, en utilisant un algorithme de hachage et du sel.

Détournement de session : Vérifier les détails de l'emplacement et les informations du navigateur et les faire correspondre avec les données historiques peut fournir des informations sur la session en cours.

Fixation de session : L'utilisation des cookies et la régénération d'identifiant de session servent à limiter ces attaques.

Données de session exposées : Éviter d'utiliser un magasin de sessions partagé.

Attaques XML : De préférence utiliser un ensemble de caractères sur liste blanche pour vous assurer que les caractères indésirables ou spéciaux ne sont pas acceptés.

Références et ressources

- https://www.lebigdata.fr/base-de-donnees
- https://sql.sh/
- https://www.w3schools.com/sql/sql_intro.asp
- https://dev.mysql.com/doc/connectors/en/apis-php-pdo-mysql.html
- https://www.php.net/manual/fr/features.cookies.php
- https://www.php.net/manual/fr/book.session.php
- https://www.getastra.com/blog/php-security/php-security-guide/
- https://www.ionos.fr/digitalguide/sites-internet/developpement-web/quest-ce-que-la-cgi/