



BTS SNIR

Lycée Louis Armand
94 - Nogent sur Marne



Module Web 04

Créer et exploiter une Base de données



Les compétences visées

</>	Catégorie	Je suis capable de :	
P H P	La balise PHP	Ajouter un code PHP dans la balise <?php	
		Inclure un fichier PHP	
		Positionner la balise script dans le code HTML	
	Les fonctions	Créer des fonctions	
		Utiliser des fonctions	
	Les variables	Créer et utiliser des variables PHP (\$)	
	Les formulaires	Récupérer la méthode d'envoi des données	
		Extraire les données à partir de \$_GET ou \$_POST	
	La classe mysqli	Créer une connexion avec le serveur mysql	
		Exécuter une requête SQL	
		Traiter la réponse de la requête SQL	
M y S Q L	Ecrire une requête	Sélectionner une données dans une table	
		Sélectionner une données dans plusieurs tables	
	Terminal	Utiliser la commande mysql	
		Créer un utilisateur avec des droits limités	
	Workbench	Créer une connexion	
		Créer des tables	
		Créer des enregistrements	
		Créer des relations clé primaire/clé étrangère	
		Synchroniser la BdD avec un serveur	
G I T	Le dépôt local	Créer un nouveau dépôt local (git init)	
		Ajouter des fichiers dans le dépôt local (git add)	
		Sauvegarder les modifications dans le dépôts (git commit)	
	Le dépôt distant	Créer un dépôt distant et se synchroniser avec (git remote)	
		Publier un dépôt local dans un dépôt distant (git push)	

TD 1 - A la découverte de PHP

Langage PHP

? Le langage PHP est-il interprété côté serveur ou client ?

?

Fonctionnalités de base

? Quelle balise est utilisée pour commencer à faire un code PHP ?

? A quoi sert la fonction « echo » ?

Les variables

? Quel symbole est obligatoirement utilisé au début de chaque variable ?

? Expliquer pourquoi le nom des variables suivantes ne sont pas autorisées ?

- \$Couleur de la Voiture :
- \$123Numero :
- \$prof@mail :
- \$Nom-du-pilote :
- AdresseDomicile :

Les tableaux

Les tableaux stockent des données sous forme de liste. Les données contenues dans la liste sont accessibles grâce à un index (représentant l'élément de la liste). Il est possible de stocker des éléments de types différents dans un même tableau.

La déclaration d'un tableau se fera de la façon suivante :

```
$piloteNum = array( 25 , "Jean", "Jacques");
```

Numériques

Dans la version numérique des tableaux, les index sont sous forme de valeur numérique (1, 2, 3...)

? Donner le code permettant d'obtenir d'extraire la deuxième valeur et la troisième valeur et de les stocker dans les variables \$nom et \$prenom

Dans cette représentation on n'est pas capable de savoir quelle valeur représente le nom ou le prénom de la personne. Et que représente la valeur de 25 ?

Associatifs

Les tableaux associatifs permettent d'utiliser des chaînes de caractères pour l'index d'un tableau. Ainsi le tableau précédent sera écrit de la façon suivante :

```
$piloteAssoc = array( "age=>25" , "prenom=>"Jean" , "nom"=>"Jacques");
```

? Quel est maintenant le code permettant de répondre à la question précédente ?

Les boucles

Pour un tableau à index numérique, vous utiliserez la boucle for afin de récupérer toutes les valeurs. Pour les tableaux associatifs, vous utiliserez la boucle « foreach » (<https://www.php.net/manual/en/control-structures.foreach.php>)

? Donner le code permettant d'afficher le contenu de la table « \$piloteNum » à l'aide de la fonction « echo »

? Faites de même pour le tableau « \$piloteAssoc »

La concaténation

Text et variables

La concaténation se fait en utilisant le sign « . ». Exemple :

```
$nom="les";  
$classe=array('snir1','snir2');  
echo "Bonjour ". $nom . " ".$classe[0];
```

Code HTML

Si vous souhaitez ajouter des balises html il vous suffira de les ajouter dans un variable et d'utiliser la méthode précédente. Exemple :

```
1. $compte = array(array('pseudo'=>'login1', 'score'=>15), array('pseudo'=>'login2', 'score'=>5);  
2. $html="<table>";  
3. for($i=0;$i<count($compte);$i++){  
4.   $joueurs=$compte[$i];  
5.   foreach($joueurs as $c=>$v)  
6.     $$c=$v;  
7.   $html=$html."<tr><td>$pseudo</td><td>$score</td></tr>";  
8. }  
9. $html.="</table>";  
10.  
11. echo $html;
```

? Expliquer le résultat des lignes 1, 4, 5, 6 et 7 du code précédent

? Dessiner le résultat de la ligne 12

Défi 1 - Installer un serveur PHP

Module PHP

Installation du module PHP

</> Connectez vous à votre serveur Linux via smarTTY

</> Taper la commande suivante pour télécharger et installer le module PHP :

```
sudo apt-get install php php-mysql libapache2-mod-php
```

Tester l'installation

Pour tester l'installation et la mise en service du module php, nous allons créer un fichier php « phpinfo.php » et le transférer vers le dossier /var/www/html

</> Créer le fichier précédent sur votre PC et ajouter le code suivant :

```
<?php echo phpinfo(); ?>
```

</> Transférer le fichier avec filezilla vers le dossier /var/www/html de votre serveur linux

Il faut maintenant changer les droits d'accès au fichier. Pour cela il faut :

</> A l'aide de smarTTY, aller dans le dossier /var/www/html avec la commande :

```
cd /varwww/html
```

</> Changer les droit d'accès en utilisant

```
chmod 755 phpinfo.php
```

</> Ouvrez un navigateur et afficher le contenu du fichier phpinfo.php en écrivant dans la barre URL : <http://ip-serveur-linux/phpinfo.php>

Activer l'affichage des erreurs PHP

</> Aller dans le fichier de configuration php avec

```
sudo nano /etc/php/7.2/apache2/php.ini
```

</> Modifier la ligne display_errors=Off par display_errors=On. Sauvegarder et quitter le fichiers

</> Redémarrer le service apache2 : systemctl restart apache2

Configurer PHP pour le dossier public_html

Voici la procédure afin d'autoriser le public_html à exécuter du code PHP :

- 1) Ouvrir le fichier : sudo nano /etc/apache2/mods-enabled/
- 2) Modifier la ligne : php_admin_flag engine Off par php_admin_flag engine On
- 3) Enregistrer et fermer
- 4) Relancer le serveur apache : sudo systemctl restart apache2

</> Exécuter les étapes précédentes et ajouter un fichier PHP dans le dossier

</> Ouvrez un navigateur et afficher le contenu du fichier phpinfo.php en écrivant dans la barre URL : <http://ip-serveur-linux/~nom-utilisateur-linux/votre-fichier.php>

PHP Version 7.2.24-0ubuntu0.18.04.7



System	Linux ubuntu 4.15.0-122-generic #124-Ubuntu SMP Thu Oct 15 13:03:05 UTC 2020 x86_64
Build Date	Oct 7 2020 15:24:25
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/apache2
Loaded Configuration File	/etc/php/7.2/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/apache2/conf.d
Additional .ini files parsed	/etc/php/7.2/apache2/conf.d/10-mysql.ini, /etc/php/7.2/apache2/conf.d/15-xml.ini, /etc/php/7.2/apache2/conf.d/20-bz2.ini, /etc/php/7.2/apache2/conf.d/20-calendar.ini, /etc/php/7.2/apache2/conf.d/20-ctype.ini, /etc/php/7.2/apache2/conf.d/20-curl.ini, /etc/php/7.2/apache2/conf.d/20-dom.ini, /etc/php/7.2/apache2/conf.d/20-exif.ini, /etc/php/7.2/apache2/conf.d/20-fileinfo.ini, /etc/php/7.2/apache2/conf.d/20-ftp.ini, /etc/php/7.2/apache2/conf.d/20-gd.ini, /etc/php/7.2/apache2/conf.d/20-gettext.ini, /etc/php/7.2/apache2/conf.d/20-iconv.ini, /etc/php/7.2/apache2/conf.d/20-json.ini, /etc/php/7.2/apache2/conf.d/20-mbstring.ini, /etc/php/7.2/apache2/conf.d/20-mysql.ini, /etc/php/7.2/apache2/conf.d/20-pdo.ini, /etc/php/7.2/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.2/apache2/conf.d/20-posix.ini, /etc/php/7.2/apache2/conf.d/20-readline.ini, /etc/php/7.2/apache2/conf.d/20-shmop.ini, /etc/php/7.2/apache2/conf.d/20-simplexml.ini, /etc/php/7.2/apache2/conf.d/20-sockets.ini, /etc/php/7.2/apache2/conf.d/20-sysmsg.ini, /etc/php/7.2/apache2/conf.d/20-sysvsem.ini, /etc/php/7.2/apache2/conf.d/20-sysvshm.ini, /etc/php/7.2/apache2/conf.d/20-tokenizer.ini, /etc/php/7.2/apache2/conf.d/20-xmlreader.ini, /etc/php/7.2/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.2/apache2/conf.d/20-xsl.ini, /etc/php/7.2/apache2/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS

Défi 2 - Installer un serveur MySQL

Serveur de base de données MySQL

Installation

</> Taper la ligne de commande suivante pour installer le serveur MySQL

```
sudo apt install mysql-server mysql-client
```

Mot de passe root pour le serveur de base de données : root

Tester l'installation

</> Vérification que le processus mysql est bien activé :

```
systemctl is-active mysql
```

La réponse est

</> Pour activer le serveur mysql, taper le code suivant :

```
systemctl start mysql
```

Créer des utilisateurs

Dans cette partie, vous allez créer des utilisateurs afin d'accéder à la base de données et de gérer uniquement vos données.

</> Commencez par taper la ligne de commande suivante permettant d'établir une connexion au serveur mysql en ligne de commande.

```
sudo mysql
```

Utilisateur snirlla

</> Taper les commandes suivantes dans le terminal smartTTY et appuyer sur entrée :

```
CREATE USER IF NOT EXISTS 'snirlla'@'%';  
ALTER USER 'snirlla'@ '%' IDENTIFIED BY 'snirlla';
```

Autre utilisateurs

Nous allons modifier la commande précédente afin de créer un autre utilisateur n'ayant accès qu'à la base de données « drone ».

```
CREATE DATABASE nom-de-la-base;  
CREATE USER IF NOT EXISTS 'votre-utilisateur'@'%';  
ALTER USER 'votre-utilisateur'@ '%' IDENTIFIED BY 'votre-mot-de-passe';  
GRANT ALL PRIVILEGES ON nom-de-la-base.* TO 'votre-utilisateur';  
FLUSH PRIVILEGES;  
exit;
```

</> Créer autant d'utilisateur et de base de données que nécessaire dans votre serveur web. Nous utiliserons la base de données « MW04_drone »

</> Vérifier que chaque utilisateur a bien accès à sa base de données avec les commandes suivantes

```
mysql -u votre-utilisateur -h localhost -p  
taper votre-mot-de-passe
```

```
show databases;
```

Ce qui affichera uniquement votre base de données.

Configuration mysql

Ouvrir le port 3306 sous linux

</> Taper la ligne de commande suivante afin d'autoriser l'accès par le port 3306 (celui de mysql) :

```
sudo /sbin/iptables -A INPUT -i eth0 -p tcp --destination-port 3306 -j ACCEPT
```

Autoriser les accès externe à mysql

</> Ouvrez le fichier mysqld.cnf avec la ligne de commande suivante :

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

</> Modifier la ligne suivante :

```
« bind-address : 127.0.0.1 » par « #bind-address : 127.0.0.1 »  
« mysqlx-bind-address : 127.0.0.1 » par « #mysqlx-bind-address : 127.0.0.1 »
```

Gestion de la base de données

Pour gérer la base de données, nous allons utiliser une interface web « phpmyadmin ». C'est ce que l'on appelle également un Système de Gestion de Base de Données (SGBD).

Installation

</> Taper la commande suivante pour télécharger et installer le site phpmyadmin

```
sudo apt install phpmyadmin
```

Lors de l'installation de phpmyadmin :

- Sélectionner le serveur web apache2
- Accepter d'utiliser l'outil dbconfig
- Indiquer le mot de passe root du serveur mysql : root

Tester l'installation

</> Dans un navigateur web, saisir l'adresse IP de votre serveur suivi de /phpmyadmin.

Le site de phpmyadmin devrait s'afficher. S'il ne s'affiche pas, c'est généralement qu'il ne s'est pas installé dans le répertoire attendu. Pour régler ce problème ajouter un lien dynamique vers phpmyadmin comme suit :

```
sudo ln -s /usr/share/phpmyadmin /var/www/html/phpmyadmin
```

</> Connectez vous avec les utilisateurs précédemment créés

</> Vérifiez la création de chaque base de données pour chaque utilisateur

TD 2 - Comprendre les bases de données

Structure d'une Bdd

Une base de données contient des tables qui elles même contiennent des champs qui contiennent des enregistrement.

? Lister le nom des tables

? Lister les enregistrements de la table utilisateur

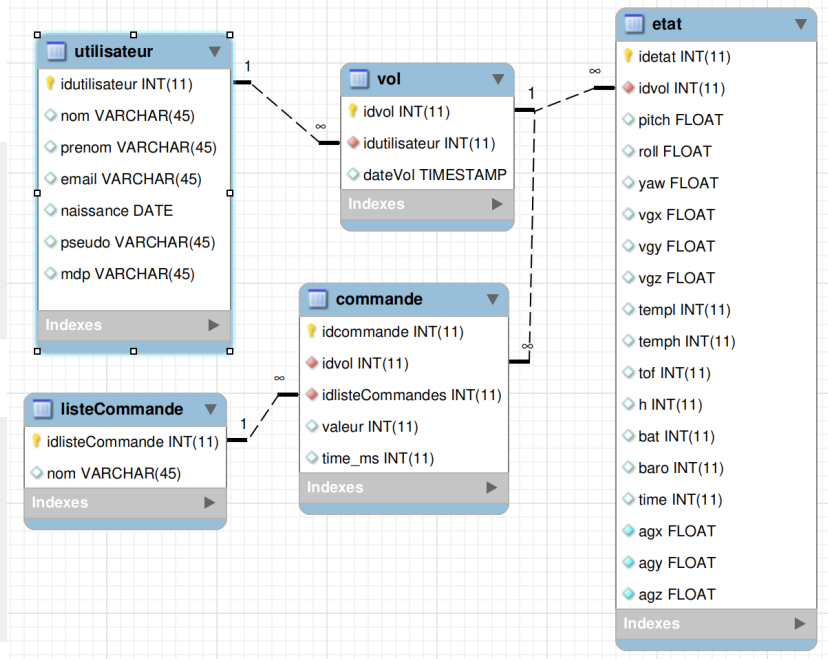


Figure 1: Base de données "MW04_drone"

Relations entre les tables

Chaque enregistrement contient des données mais également des relations avec les autres tables. Ainsi à partir de la table vol, nous pouvons obtenir les informations concernant les utilisateurs, les états de vol et les commandes de vol. Les liens sont stockés sous forme de clé primaire / clé étrangère.

? En vous aidant du net donnez la définition de clé primaire et clé étrangère.

? Dans la Bdd précédente, listez toutes les clés primaires et clés étrangères dans chaque tableaux

Chaque relation est également définie par une cardinalité.

? Donner les cardinalités des relations précédentes

Défi 3 - Créer une base de données

Concevoir une base de données

Installation et configuration

- </> Installer MysqlWorkbench.
- </> Configurer la version MySQL utilisé dans Mysqlworkbench afin qu'elle corresponde à celle utilisé par votre serveur SQL en tapant la commande linux suivante :

```
mysql --version
```

- </> Dans mysqlwokbench aller dans Edit→Preferences→Modeling/Mysql : modifier la version Mysql
- Maintenant il faut configurer la connection entre mysqlworkbench et le serveur mysql linux. Voici les étapes à suivre :
- </> Ouvrez le menu Tools et cliquez sur Connect to database
 - </> Nommer la connexion « Mini-PC »
 - </> Compléter les paramètres en ajoutant l'adresse IP du mini-PC, l'utilisateur de le mot de passe utilisée pour la base de données Mysql.

Création d'une base de donnée

En vous aidant du document « Utilisation de MysqlWorkbench.pdf »,

- </> Créer la base de données précédente en respectant :
 - Nommer la base de données « MW04_drone »
 - Les relations clé primaire / clé étrangère,
 - Cocher les cases PK, NN et AI pour les clés primaires
 - Les clés étrangères portent le même nom que les clés primaires

Synchronisation vers le serveur mysql

- </> Synchroniser votre fichier mysqlworkbench vers le serveur mysql à l'aide de la connection « Mini-PC » en allant dans le menu Database→Synchronize Model... Sélectionner votre connexion et suivre les étapes.
- </> Ouvrir phpmyadmin et vérifier que la base de données a bien été synchronisée

Défi 4 - Importer des données

Une fois la BdD créée, il faut maintenant importer des données dans les différentes tables. Nous focaliserons notre travail sur la table « utilisateur »

Créer des données avec phpmyadmin

Ici nous allons utiliser phpmyadmin pour créer des données. Voici les étapes

</> Ouvrir la table « utilisateur » en cliquant dessus.

</> Aller dans l'onglet « insert »

Compléter le formulaire en ajoutant l'utilisateur suivant : « Jean, Jacques, jean.jacques@protonmail.com,

</> 1970-01-01, jeanjean, mdpDeJean01! ». **Attention il ne faut surtout pas remplir la clé primaire car elle a été définie comme auto-incrémentable.**

Créer des données dans un fichier externe

Pour compléter une base de données, la méthode précédente est pratique si nous avons une ou deux ligne à ajouter. Or dans la plus part des cas, vous avez plusieurs dizaines de données à importer. Nous allons donc utiliser un fichier « .csv »

</> Exporter la table utilisateur au format « csv for Ms excel » et cocher la case permettant d'ajouter le nom des colonnes au fichier

</> Compléter le fichier précédent avec les données suivantes

1	ABATAN	Evan	Evan.ABATAN@protonmail.com	1970-01-02	login1	mdp1		
2	AIT MANSOUR	Inas	Inas.AIT MANSOUR@protonmail.com	1970-01-03	login2	mdp2		
3	BECKER	Maxime	Maxime.BECKER@protonmail.com	1970-01-04	login3	mdp3		
4	BERANGER	Aurélien	Aurélien.BERANGER@protonmail.com	1970-01-05	login4	mdp4		
5	BLIN Florent	Florent	Florent.BLIN@protonmail.com	1970-01-06	login5	mdp5		
6	CESARI Jeffrey	Jeffrey	Jeffrey.CESARI@protonmail.com	1970-01-07	login6	mdp6		
7	CHARLES	Dimitry	Dimitry.CHARLES@protonmail.com	1970-01-08	login7	mdp7		
8	CHARTIER	Pierre	Pierre.CHARTIER@protonmail.com	1970-01-09	login8	mdp8		
9	CHRISTON	Vincent	Vincent.CHRISTON@protonmail.com	1970-01-10	login9	mdp9		
10	CONTE-LAY	Corentin	Corentin.CONTE-LAY@protonmail.com	1970-01-11	login10	mdp10		

</> Importer à nouveau le fichier dans phpmyadmin en sélectionnant le fichier csv et en complétant les bonnes cases:

TD 3 - Interroger une base de données : comprendre les requêtes SQL

En utilisant la base de donnée droneEcrire la requête permettant de :

Sectionner une donnée

Dans une table

Sans condition

```
Select champ1 From table1
```

Exemple : On souhaite sélectionner le nom des utilisateurs

```
Select nom from utilisateur
```

? Donner la requête permettant de sélectionner le nom et prénom des utilisateurs

? Donner la requête permettant de sélectionner tous les champs des utilisateurs

Avec condition

```
Select champ1 From table1 where condition
```

Exemple : On souhaite sélectionner le nom des utilisateurs dont le pseudo est jeanjean

```
Select nom from utilisateur where pseudo='jeanjean'
```

? Donner la requête permettant de sélectionner l'email des utilisateurs dont la date de naissance est « 2000-01-01 »

Dans deux tables

```
Select champ1 From table1 inner join table2 on table1.pk=table2.fk_pktable1
```

Exemple : On souhaite sélectionner le nom des utilisateurs ayant effectué un vol

```
Select nom from utilisateur inner join vol on utilisateur.idutilisateur=vol.idutilisateur
```

? Donner la requête permettant de sélectionner le nom et prénom des utilisateurs ayant effectué un vol le 2020-09-10

Insérer une donnée

La requête permettant d'insérer une donnée à la structure suivante :

```
Insert into table1 (champ1,champ2,..) values (valeur1,valeur2,...)
```

? Donner la requête permettant d'insérer un nouvel utilisateur

Mettre à jour une donnée

La requête permettant d'insérer une donnée à la structure suivante :

```
Update table1 set champ1=valeur1 , champ2=valeur2 where condition
```

 Donner la requête permettant de mettre à jour le pseudo de l'utilisateur dont le nom est Jean

Défi 5 - Connexion entre PHP et une base de données MySQL

Objectif

Créer un fichier « **rest.php** » qui sera utilisé pour interroger la base de données et envoyer la réponse à l'utilisateur

Comprendre la classe mysqli

La connexion entre un code PHP et une base de données MySQL s'effectue en utilisant la classe mysqli (<https://www.php.net/manual/fr/class.mysqli.php>). Voici les étapes et les différentes méthodes utilisées :

- 1) Déclarer un nouvel objet de la classe mysqli (<https://www.php.net/manual/fr/mysqli.construct.php>)
- 2) Créer une variable contenant la requête SQL
- 3) Exécuter la requête à l'aide de la méthode mysqli::query (<https://www.php.net/manual/fr/mysqli.query.php>)
- 4) Extraire les données à l'aide de mysqli_query::fetch_assoc (<https://www.php.net/manual/fr/mysqli-result.fetch-assoc.php>)
- 5) Répéter les étapes 2 et 3 pour exécuter d'autres requêtes

Voici un exemple permettant de récupérer la valeur du champs1 dans la table1 de votre base de données et qui sera stocké dans la variable \$champs1

```
$mysqli = new mysqli("127.0.0.1","utilisateur-BdD","mot-de-passe","nom-de-la-bdd");  
$req="select champs1 from table1";  
$res = $mysqli->query($req);  
$rep = $res->fetch_assoc();  
// Afficher le tableau $rep avec : print_r($rep);  
$champs1 = $rep['champs1'];  
// Afficher le contenu d'une variable avec : echo $champs1;
```

- ? Sur le code précédent repérer les étapes 1,2 et 3
- ? Donner le code complet permettant de récupérer le nom et prénom des utilisateurs dans la base de données drone en utilisant le compte « root » et mot de passe « mdpRoot »

Mettre en place le code

Nous allons tester ces étapes dans un fichier php que vous nomerez « rest.php »

- </> Créer le fichier rest.php et le mettre à la racine de votre site web
- </> Compléter le fichier avec le code précédent
- </> Ajouter le code permettant d'afficher la réponse sur la page web en utilisant soit la fonction echo soit print_r() ;
- </> Tester le bon fonctionnement en appelant le fichier rest.php dans le navigateur en utilisant l'url suivante par exemple : 127.0.0.1/MW04/rest.php

Défi 6 - BdD à travers un formulaire de connexion

Objectif

Nous allons ici compléter le fichier « **rest.php** » afin d'exploiter le pseudo et le mot de passe d'un formulaire de connexion afin de connecter un utilisateur.

Récupérer les données du formulaire

Comprendre \$_GET ou \$_POST

Pour récupérer les données envoyées par un formulaire, nous devons connaître la méthode utilisée dans la balise form du code HTML.

```
// Si la méthode vaut get alors les variables seront stockées dans un tableau associatif $_GET.  
// Si la méthode vaut post alors les variables seront stockées dans un tableau associatif $_POST.
```

Exemple :

```
// Code HTML  
<form action="rest.php" method="get">  
  <input type="text" name="pseudo"/>  
  <input type="submit" name="validForm"/>  
</form>  
  
// Code rest.php  
if (isset($_GET['validForm'])){  
  $pseudo=$_GET['pseudo'];  
  echo "le pseudo vaut ".$pseudo;  
}
```



Expliquer les différentes lignes du code précédent

Application



En utilisant le formulaire de connexion du module MW02, donner le nom des différentes associations clé=>valeur qui seront envoyées.

Mise en œuvre

</> Avant de faire la suite, il faut mettre la méthode **GET** du formulaire de connexion



Donner la requête SQL qui permet de sélectionner le nom, le prénom ainsi que l'email connaissant son pseudo.

</> Dans le fichier rest.php, écrire le code permettant : d'interroger la table utilisateur connaissant le pseudo et le mot de passe, en utilisant la requête précédente et les étapes 2-3 du Défi-5. Vous utiliserez le contenu de la méthode GET pour obtenir le pseudo et le mot de passe

Pour savoir si la réponse d'une base de données n'est pas vide nous pouvons utiliser la méthode « num_rows » de la classe mysqli_result (<https://www.php.net/manual/fr/mysqli-result.num-rows.php>)

? Que signifie si la réponse retournée par num_rows est différente de zéro, appliqué à la requête précédente ?

</> Compléter le code précédent utilisant la méthode « num_rows » appliqué au résultat de la requête. Si la réponse est zéro alors il n'y a pas d'utilisateur et on affiche un message d'alerte. Sinon on affiche le nom et prénom de l'utilisateur

</> Compléter le code afin d'afficher le nom et prénom de l'utilisateur si la réponse de la base de données n'est pas vide sinon afficher un message d'erreur à l'utilisateur

? Quelle est le problème de la méthode GET ?

</> Réécrire le code avec la méthode POST

ANNEXE

Que faire en cas d'erreur PHP ?

Erreur 500 dans l'inspecteur d'élément

1. Vérifier que votre service PHP est capable d'afficher les erreurs en activant l'option « display_errors » dans le fichier php.ini. Régler l'erreur si besoin.

```
Sudo nano /etc/php/7.2/apache2/php.ini
```

2. Vérifier les permissions d'accès des fichiers. Faire un chmod si besoin :

```
chmod -R 755 ./votre-dossier
```

Aucun changement n'est visible sur la page web

3. Vérifier que vous avez la dernière version de votre fichier sur le serveur web
4. Vérifier que votre base de données est accessible

La réponse de la BdD affiche une erreur

5. Vérifier votre requête en affichant la variable \$req par exemple. Cela vous permettra de voir la structure de la requête SQL. Puis tester la requête dans phpmyadmin en allant dans l'onglet « SQL »