

Développement d'une application web dynamique en php sur raspberrypi-sin

1 Préambule

- Pré-requis :
 - avoir suivi le tutorial de prise en main du serveur raspberrypi-sin
 - avoir suivi le tutorial de gestion d'une base de données avec PhpMyAdmin
- L'objectif de ce tutorial est d'apprendre à créer une interface web pour gérer une base de données via un navigateur. Par "gérer", on entend :
 - lister les données
 - ajouter une donnée
 - modifier une donnée
 - supprimer une donnée
- Dans un premier temps, le tutorial explique la construction "pas à pas" de la gestion des livres. L'objectif étant d'appliquer les connaissances acquises pour créer les pages de gestion des abonnés et des emprunts.
- On se contentera de pages html minimales. On pourra améliorer l'aspect des pages en complétant ultérieurement le fichier CSS
- On supposera que la table livres existe dans la bases de données.
Sinon, exécuter la requête suivante dans userx_db avec PhpMyAdmin

```
CREATE TABLE livres (
  id_livre smallint(5) unsigned NOT NULL AUTO_INCREMENT,
  auteur varchar(20) NOT NULL,
  edition varchar(4) NOT NULL,
  titre varchar(50) NOT NULL,
  description blob,
  PRIMARY KEY ('id_livre')
);
```

- Dans la suite de ce document,
 - on désignera le compte utilisateur par *userx*.
Remplacer le x par la valeur qui vous a été attribuée
 - on supposera qu'on travaille sous Windows.
Les utilisateurs Mac ou Linux adapteront aisément
 - Pour se repérer lorsque une portion de code est à insérer dans du code existant, les lignes en bleu indiquent les lignes existantes

```
//remplissage des lignes du tableau
//on lit les resultats de la requete
while($row=mysqli_fetch_object($result)){
  echo '<tr>'; //balise debut de ligne
  //cellules de la ligne
  echo '<td>'.$row->auteur.'</td>'
  . '<td>'.$row->titre.'</td>'
  . '<td>'.$row->description.'</td>'
  . '<td>'.$row->edition.'</td>';
  echo '</tr>'; //fin de ligne
};
echo '</tbody>'
```

code à insérer

lignes de code existantes

2 Mise en place de l'environnement de travail

1. **Connecter** les lecteurs réseau :
 - a) R sur le partage //ip_raspberrypi-sin/Logiciels
 - b) S sur le partage //ip_raspberrypi-sin/homes
2. **Créer** un dossier **Mon_appli** dans le dossier public_html.
Ce dossier contiendra tous les fichiers de l'application
3. **Ouvrir** le logiciel R:\Notepad++Portable\Notepad++Portable.exe

3 Page d'accueil du site

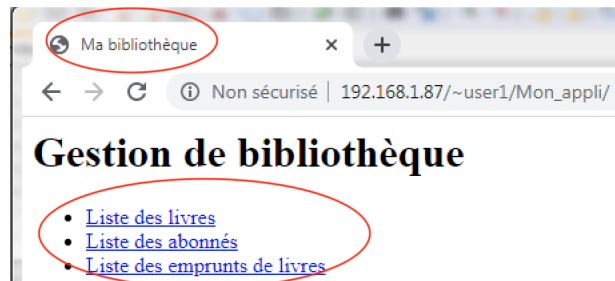
1. **Créer** une page html de base, en collant le code suivant dans Notepad++

```
<html>
<head>
  <title>Ma bibliothèque</title>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width">
</head>
<body>
<h1> Gestion de bibliothèque</h1>
  <div>
    <ul>
      <li><a href=gestion_livres.php>Liste des livres</a></li>
      <li><a href=gestion_users.php>Listes des abonnés</a></li>
      <li><a href=gestion_emprunts.php>Liste des emprunts de livres </a></li>
    </ul>
  </div>
</body>
</html>
```

La balise html title permet de donner un nom à la fenêtre (onglet)

Dans le corps de la page on a créé une liste à puces de liens vers les pages de gestions spécifiques (livres, abonnés, emprunts)

2. **Enregistrer** sous S:\public_html\Mon_appli\index.html
3. **Appeler** cette page dans un navigateur : http:\ip_raspberrypi-sin\~userx\Mon_appli\
On obtient



Les 3 liens pointent vers des fichiers php
 - gestion_livres.php
 - gestion_users.php
 - gestion_emprunts.php
qu'on va créer ci-après

4 Affichage de la liste des livres dans une page web

Ce fichier doit lire la liste des livres enregistrés dans la table sql nommée *livres*, puis les afficher dans un tableau html.

1. Création d'un script (fichier) php

Créer un nouveau fichier dans Notepad++ et enregistrez le sous
S:\public_html\Mon_appli\gestion_livres.php

Coller le code suivant dans le nouveau fichier créé

```
<?php
//envoi du code html du debut de page
echo '
<html>
  <head>
    <meta charset="UTF-8">
    <title>Livres</title>
  </head>
  <body>
    <h1>Gestion des livres</h1>
    <p><a href=index.html title="Retour">Retour à l\'accueil</a></p>
    <table border="1">
      <thead>
        <tr>
          <th>Titre</th>
          <th>Auteur</th>
          <th>Année d\'édition</th>
          <th>Description</th>
        </tr>
      </thead>
      <tbody>
';
//remplissage de la table
//envoi du code html de la fin de page
echo '
  </tbody>
</table>
</body>
</html>';
?>
```

Quand le fichier php est appelé avec un navigateur, une commande echo envoie au navigateur le texte qui se trouve entre les apostrophes.

Ici, la première commande echo envoie le début du code html avec les entêtes du tableau

La seconde envoie le code html pour fermer les balises

Pour afficher le contenu du tableau, on insérera du code php entre les 2 commandes echo

Noter l'échappement de l'apostrophe dans le texte Année d'édition pour que cet apostrophe ne soit pas pris comme délimiteur de de fin dans la commande echo ' '

2. Test du code

Appeler cette page dans un navigateur en cliquant sur le lien "Liste des livres" de la page d'accueil créée précédemment

On obtient



c'est moche mais ça montre que le lien html et le script php fonctionnent !
Vérifier également le fonctionnement du lien "Retour à l'accueil"

3. Alimentation du tableau avec la base de données

Il va falloir :

- se connecter à la base de données
- exécuter une requête sql pour récupérer la liste des livres
- afficher le résultat de la requête sql dans le tableau html

a) Connexion à la base de données (bdd)

Créer un nouveau fichier avec Notepad++, et l'enregistrer sous :

S:\public_html\Mon_appli\connect.php

Coller le code suivant en corrigeant l'identifiant, le mot de passe et le nom de base de données

```
<?php
// Definition des parametres d'acces sous forme de constantes
DEFINE ('DB_USER', 'userx');
DEFINE ('DB_PASSWORD', 'votre_mdp');
DEFINE ('DB_HOST', 'localhost');
DEFINE ('DB_NAME', 'userx_db');

// Ouvrir la connexion à la base de données
$dbc = mysqli_connect (DB_HOST, DB_USER, DB_PASSWORD,DB_NAME)
    OR die ('Connexion a MySQL impossible : '.mysqli_error($dbc).'\n');
?>
```

Rmq :

En cas d'erreur de connexion, on arrête le script avec "die", et on affiche l'erreur mysql

Notez le nom de la connexion \$dbc que l'on retrouvera plus tard dans les scripts

Vérifier la connexion en appelant le fichier connect.php avec un navigateur
http://ip_raspberrypi-sin/~userx/mon_appli/connect.php

S'il n'y a pas d'erreur, on obtient une page "blanche". En cas d'erreur, vérifier les paramètres de connexion.

b) Lecture des données dans la table livres

Insérer le code suivant au début du script gestion_livres.php

```
<?php
//connexion à la bdd
include 'connect.php';
// requete sql pour extraire la liste des livres
// Creer la requete.
$req = "SELECT id_livre, auteur, titre, description, edition FROM livres ORDER BY id_livre
ASC";
// lancer la requete
$result = mysqli_query ($dbc,$req) or die (mysqli_error($dbc));
//envoi du code html du debut de page
echo '

```

Rmq :

- include est équivalent à un copier/coller du code du fichier appelé. -
la variable \$result contient les données demandées

c) Affichage dans le tableau

Insérer le code suivant dans le fichier gestion_livres.php

```
//remplissage de la table
//on lit les resultats de la requete
while($row=mysqli_fetch_object($result)){
echo '<tr>'; //balise debut de ligne
//cellules de la ligne avec les données recuperées
echo '<td>'. $row->titre. '</td>'
.'<td>'. $row->auteur. '</td>'
.'<td>'. $row->edition. '</td>'
.'<td>'. $row->description. '</td>';
echo '</tr>'; //fin de ligne
};
//envoi du code html de la fin de page
echo '

```

- La boucle While permet de lire, livre par livre, les données contenues dans \$result et de les mettre dans la variable \$row

- \$row contient les données d'un livre

- par exemple, on récupère le titre dans la variable \$row->titre

- on affiche une ligne de 4 cellules par livre



Titre	Auteur	Année d'édition	Description
Le Comte de Monte-Cristo	Alexandre Dumas	1844	Edmond est dénoncé
Les Misérables	Victor HUGO	1862	Jean Valjean et le destin tragique
...

Encore moche mais fonctionnel !!

5 Amélioration de la présentation

On va montrer (ou rappeler) comment on peut améliorer l'aspect des pages web en utilisant les feuilles de style CSS (*Cascading Style Sheets*)

1. **Créer** un fichier avec Notepad++, y coller le contenu ci-dessous et l'enregistrer sous : `S:\public_html\Mon_appli\table.css`

```
body {
  font-size:14px;
  color: #000000;
  font-family: arial ,Helvetica, sans-serif, verdana;
}

td.ligne{
  border-top:1px solid #6090CC;
  border-bottom:1px solid #6090CC;
  padding:1px;
  font-size:10pt;
}

td.ligne0{
  background-color:#dbe6f7;
}

td.ligne1{
  background-color:#F8F8FF;
}

th{
  font-style: italic;
  width:120px;
  background-color:#001066;
  color: #ffffff;
}

td {
  text-align: center;
}
```

Rmq : on crée des classes différentes pour les lignes paires(ligne0) et impaires (ligne1) du tableau

2. **Ajouter** le lien vers ce fichier dans l'entête (<head> </head>) du code html généré dans `gestion_livres.php`

```
<title>Livres</title>
<link rel="stylesheet" type="text/css" href="table.css" media="all" />
</head>
```

3. **Remplacer** le code qui remplit les lignes du tableau par le suivant

```
//on lit les resultats de la requete
$i=0;
while($row=mysqli_fetch_object($result)){
  $classe_ligne='class="ligne ligne'.$i.'";
  echo '<tr '.$classe_ligne.'>'; //balise debut de ligne
  //cellules de la ligne
  echo '<td '.$classe_ligne.'>'.$row->auteur.'</td>'
  .'<td '.$classe_ligne.'>'.$row->titre.'</td>'
  .'<td '.$classe_ligne.'>'.$row->edition.'</td>'
  .'<td '.$classe_ligne.'>'.$row->description.'</td>';
  echo '</tr>';//fin de ligne
  $i++;
};
echo '

```

Explication :

*on affecte un numéro de ligne avec la variable \$i
on calcule le modulo2 de \$i (\$i%2) pour savoir si la ligne est paire/impair
\$classe_ligne vaut class="ligne ligne0" pour les lignes paires
\$classe_ligne vaut class="ligne ligne1" pour les lignes impaires*

Ce qui donne



Titre	Auteur	Année d'édition	Description
Le Comte de Monte-Cristo	Alexandre Dumas	1844	Edmond est dénoncé
Les Misérables	Victor HUGO	1862	Jean Valjean et le destin tragique
Notre-Dame de PARIS	Victor HUGO	1831	une jeune appelée Esmeralda danse
Germinal	Emile ZOLA	1885	Le jeune Etienne s'est fait renvoyer

C'est mieux !

6 Ajouter une donnée

On commence par ajouter un lien html dans la page de gestion des livres pointant vers le nouveau fichier qui permettra d'ajouter un livre.

1. Ajouter le lien ci-dessous dans gestion_livres.php

```
<p><a href=index.html title="Retour">Retour à l'accueil</a></p>
<p><a href=add_livre.php title="Ajouter un livre">Ajouter un livre</a></p>
<table border="1">
```

En rechargeant la page dans le navigateur on obtient :



2. Créer un fichier avec Notepad++ et l'enregistrer sous S:\public_html\Mon_appli\add_livre.php

```
<?php
include 'connect.php';
//traitement du formulaire
if (isset($_POST['Annuler'])) {
header("location: gestion_livres.php");//redirection
exit;
}
if (isset($_POST['Valider'])) { //si a cliqué sur Valider
//protection minimale
$auteur_esc= mysqli_real_escape_string($dbc,$_POST['auteur']);
$titre_esc= mysqli_real_escape_string($dbc,$_POST['titre']);
$edition_esc= mysqli_real_escape_string($dbc,$_POST['edition']);
$description_esc= mysqli_real_escape_string($dbc,$_POST['description']);
//requete
$sql="INSERT INTO livres VALUES
('','$auteur_esc','$edition_esc','$titre_esc','$description_esc')";
//execution de la requete
$result = mysqli_query($dbc,$sql);
//en cas d'erreur
if (!$result) {
echo 'Erreur : '.mysqli_error($dbc);
}
//sinon on redirige vers la page gestion_livres.php
else {
header("location: gestion_livres.php");
exit;
}
}
echo '<html>
<head>
<meta charset="UTF-8">
<title>Ajoute un livre</title>
<link rel="stylesheet" type="text/css" href="table.css" media="all" />
</head>
<body>
<h1>Ajout d'un livre</h1>';

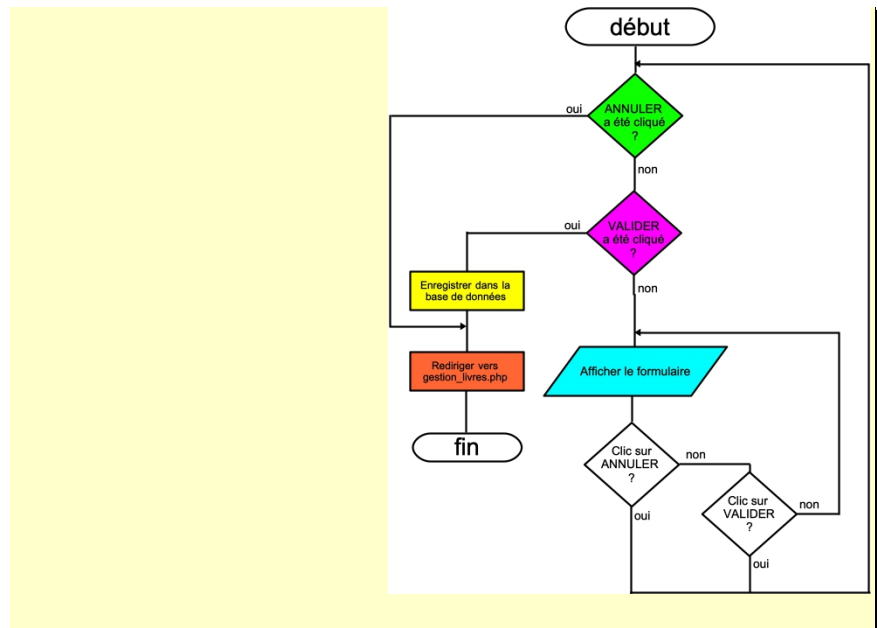
echo '<form name = "add" action = "add_livre.php" method = "POST">
<label for="nom">Auteur : <input type = "text" name = "auteur" value = "" size = "20"
/></label><br /><br />
<label for="titre">Titre : <input type = "text" name = "titre" value = "" size = "50"
/><br /><br />
<label for="edition">Année d'édition : <input type = "text" name = "edition" value = ""
size="4" /><br /><br />
<label for="description">Description : <textarea name="description" rows="4" cols="30">
</textarea><br /><br />
<input type="submit" value="Valider" name="Valider" />
<input type="submit" value="Annuler" name="Annuler" /></form>';

echo '
</body>
</html>';
?>
```

Lorsqu'on clique sur le bouton Valider, on rappelle le même fichier en lui passant en paramètres une variable qui s'appelle "Valider" plus les données qui ont été saisies.

Lorsqu'on clique sur le bouton Annuler, on rappelle le même fichier en lui passant en paramètre une variable qui s'appelle "Annuler"

Le chronogramme ci-après montre la structure d'un formulaire html et son traitement.



On obtient le formulaire suivant :

7 Suppression d'une donnée

Pour chaque livre affiché, on va ajouter en fin de ligne un lien html pour le supprimer.

- Il faut donc ajouter une colonne sans titre à droite du tableau dans gestion_livres.php

```
<th>Description</th>
<th></th>
</tr>
```

- Puis pour chaque ligne, une cellule avec le lien

```
.'<td ',$classe_ligne.'>'.$row->edition.'</td>';
echo '<td ',$classe_ligne.'><a href=gestion_livres.php?delete_id='.$row->id_livre.'
title=Supprimer livre '.$row->id_livre.'>Supprimer</a></td>';
echo '</tr>';//fin de ligne
```

*Le lien pointe vers la même page, mais passe une variable **delete_id** qui contient l'id de le livre à supprimer*

Dans le script appelé, on récupère la variable avec `$_GET['delete_id']`

- En début de script, si on détecte l'existence de la variable "delete_id", on lance la requete sql qui supprimera le livre de la table qui a l'id correspondant

```
include 'connect.php';
//Suppression d'un livre
if (isset($_GET['delete_id'])){
    //par precaution on echappe la variable
    $id_to_delete=mysqli_real_escape_string($dbc,$_GET['delete_id']);
    $rq="DELETE FROM livres WHERE id_livre=$id_to_delete";
    $result = mysqli_query($dbc,$rq) or die (mysqli_error($dbc));
}
// requete sql pour extraire la liste des livres
```

Rmq : `mysqli_real_escape_string` permet d'échapper les caractères spéciaux pour se prémunir des injections sql.

8 Modification d'une donnée

- On ajoute une colonne sans nom au tableau de la page livres.php

```
<th></th>
<th></th>
</tr>
```

- On ajoute en fin de ligne une cellule contenant un lien pour chaque livre repéré par son id

```
echo '<td '.$classe_ligne.'><a href=gestion_livres.php?delete_id='.$row->id_livre.'
title="Supprimer livre '.$row->id_livre.'">Supprimer</a></td>';
echo '<td '.$classe_ligne.'><a href=modif_livre.php?modif_id='.$row-
>id_livre.'">Modifier</a></td>';
echo '</tr>';//fin de ligne
```

- Le lien pointe vers un nouveau fichier `modif_livre.php`, et passe une variable `modif_id` qui contient l'id du livre à modifier
Dans le script appelé, on récupère la variable avec `$_GET['modif_id']`

- Créer un fichier avec Notepad++ et l'enregistrer sous `S:\public_html\Mon_appli\modif_livre.php`

```
<?php
include 'connect.php';
if (isset($_GET['modif_id'])){
    //par precaution on echappe la variable
    $id_to_modif= mysqli_real_escape_string($dbc,$_GET['modif_id']);
    $rq="SELECT id_livre,auteur,edition,titre,description FROM livres WHERE
id_livre='$id_to_modif'";
    $result = mysqli_query($dbc,$rq) or die (mysqli_error($dbc));
    if (mysqli_num_rows($result)>0)$row=mysqli_fetch_object($result);
}
```

si on détecte l'existence de la variable "modif_id", on lance la requete sql qui récupère les données du livre pour les afficher dans le formulaire

- On affiche le formulaire avec les données récupérées dans les champs correspondants

```
if (mysqli_num_rows($result)>0)$row=mysqli_fetch_object($result);
}
echo '<html>
<head>
    <meta charset="UTF-8">
    <title>Modifier un livre</title>
    <link rel="stylesheet" type="text/css" href="table.css" media="all" />
</head>
<body>
<h1>Modification d\'un livre</h1>';

echo '<form name = "add" action = "modif_livre.php" method = "POST">
<label for="auteur">Auteur :
<input type = "text" name = "auteur" value = "'.$row->auteur.'" size = "20"
/></label><br /><br />
<label for="edition">Année d\'édition :<input type = "text" name = "edition" value = "'.$row->edition.'" size = "20" /><br /><br />
<label for="titre">Titre :
<input type = "text" name = "titre" value = "'.$row->titre.'" /><br /><br />
<label for="description">
Description : <textarea name="description" rows="4" cols="20">'.$row->description.'
</textarea><br /><br />
<input type="hidden" value="'.$row->id_livre.'" name="livre" />
<input type="submit" value="Valider" name="updater" />
<input type="submit" value="Annuler" name="Annuler" /></form>';

echo '
</body>
</html>';
?>
```

Remarquer :

L'affichage des valeurs dans les champs

- Traitement du formulaire (à placer au début du script)

- Si on détecte la présence de la variable `$_POST['updater']`, on lance la requête de mise à jour de la table avec les données modifiées et renvoyées par le formulaire.

```
include 'connect.php';
//traitement du formulaire
if (isset($_POST['Annuler'])){
header("location: gestion_livres.php");
exit;
}
if (isset($_POST['updater'])){
    //protection minimale
    $auteur_esc= mysqli_real_escape_string($dbc,$_POST['auteur']);
    $edition_esc= mysqli_real_escape_string($dbc,$_POST['edition']);
    $titre_esc= mysqli_real_escape_string($dbc,$_POST['titre']);
    $description_esc = mysqli_real_escape_string($dbc,$_POST['description']);
    $livre_esc = mysqli_real_escape_string($dbc,$_POST['livre']);
    //requete
    $rq="UPDATE livres SET
auteur='$auteur_esc',edition='$edition_esc',titre='$titre_esc',description='$description_e
sc' WHERE id_livre='$livre_esc'";
    //execution de la requete
    $result = mysqli_query ($dbc,$rq) ;
    //en cas d'erreur
    if (!$result) {echo 'Erreur : '.mysqli_error($dbc);}
    //sinon on redirige vers la page livres.php
    else {
header("location: gestion_livres.php");
}
```

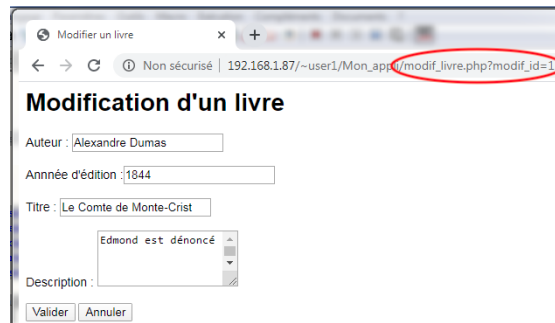


```

exit;
}}
if (isset($_GET['modif_id'])){

```

on obtient



Noter comment l'id du livre est passé dans l'url

9 le petit plus

Pour améliorer la présentation on peut

1. remplacer le texte des liens html par des images dans gestion_livres.php

```

<p><a href=index.html title="Retour">Retour à l'accueil</a></p>
<p><a href=add_livre.php title="Ajouter un livre">
</a></p>
<table border="1">

```

```

echo '<td '.$classe_ligne.'><a href=gestion_livres.php?delete_id='.$row->id_livre.'
title="Supprimer livre '.$row->id_livre.'">
</a></td>';
echo '<td '.$classe_ligne.'><a href=modif_livres.php?modif_id='.$row->id_livre.'">
</a></td>';
echo '</tr>';//fin de ligne

```

2. ajouter une couleur de fond à la page dans le fichier table.css

```

body {
font-family: arial ,Helvetica, sans-serif, verdana;
background-color:#DCECF9;
}

```

On obtient finalement



10 Application

1. En se basant sur ce qui a été fait pour les livres, créer le dispositif (mysql+php) permettant de gérer les abonnés.
Les informations à enregistrer sont : Nom, prénom, adresse mail , n° compte (nombre à 5 chiffres).
2. Créer le dispositif (mysql+php) permettant de gérer les emprunts.
Les informations à enregistrer sont : id abonné, id livre emprunté, date d'emprunt