# TP 4 Le PHP

***Vous devrez déposer l’ensemble des fichiers par SFTP dans le dossier G?/www/TP4. Date limite jeudi 08/10/2020 à 8h00.***

*Dans tout le TP, il est conseillé de tester à chaque fois que vous écrivez une ligne de code. Si vous êtes bloqué, simplifiez le code au maximum pour isoler le problème, il ne doit rester que les 3-5 lignes qui posent problèmes. Dans 95% des cas, une fois le problème isolé, la solution apparaitra d’elle-même. Dans les 5% restants vous trouverez sans doute la réponse dans des forums (ex. stackoverflow) ou vous pouvez éventuellement poser une question sur ces forums.*

1. **Création de la table patient**

Créer un fichier data.php et copier le code suivant :

*Note pour l’année prochaine :*

*Mettre une intro général objectif tu TP*

*Renommer pour que ça ressemble exemple id =>IdPatient*

*Passer birthday en date*

*Renommer les fichiers comme dans TP3 en anglais ; Regarder pour faire sous github*

<?php

$patients=**array**(

**array**("id"=>1,"sexe"=>1,"prenom"=>"Obiwan","nom"=>"Kennoby","email"=>"ok@tatooine.com","birthday"=>-59032152),

**array**("id"=>2,"sexe"=>2,"prenom"=>"Blanche","nom"=>"Neige","email"=>"bn@disney.com","birthday"=>886974048),

**array**("id"=>3,"sexe"=>1,"prenom"=>"Chuck","nom"=>"Norris","email"=>"chuck@norris.com","birthday"=>-209876952),

**array**("id"=>4,"sexe"=>1,"prenom"=>"Oui","nom"=>"Oui","email"=>"oui@oui.com","birthday"=>673439848 )

);

$prelevements=**array**(

**array**("id"=>1,"idPatient"=>1,"date"=>1403468000,"type"=>"Sanguain","Résultat"=>"Positif à la force"),

**array**("id"=>2,"idPatient"=>1,"date"=>1503968000,"type"=>"Sanguain","Résultat"=>"Négatif au côté obscure"),

**array**("id"=>3,"idPatient"=>1,"date"=>1543968000,"type"=>"Urinaire","Résultat"=>"A abusé de

Midi-chlorien au petit déjeuner"),

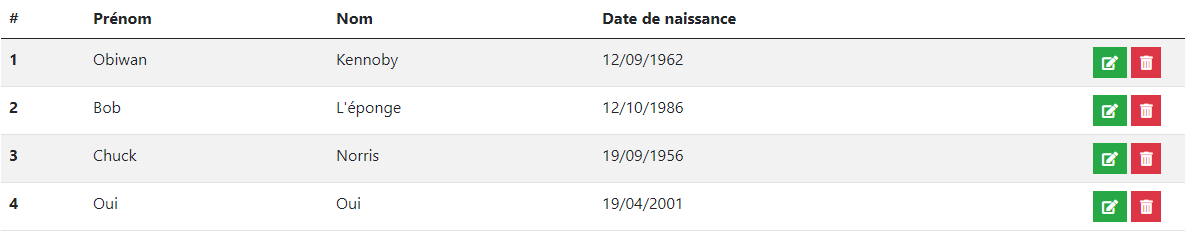
**array**("id"=>4,"idPatient"=>3,"date"=>1583968000,"type"=>"Urinaire","Résultat"=>"Echec de prélèvement, on ne prélève pas Chuck Norris!"),

**array**("id"=>5,"idPatient"=>4,"date"=>1513968000,"type"=>"Psychologique","Résultat"=>"Ne sait pas dire non")

);

?>

Renommer le fichier du TP précédent avec le tableau des patients en listpatients.php :



Ajouter en php dans ce fichier (tout en haut) <?php **include** 'data.php'; ?>;

Dans ce TP les données sont stockées sous forme de array PHP. Dans le TP5, nous verrons comment récupérer ces tableaux depuis une base de données.

En utilisant l’alternance HTML-PHP vu dans la vidéo et en utilisant une boucle foreach, modifier le code pour que l’affichage du tableau dépende du contenu de $patients.

Vous devrez donc dire pour chaque $patient dans $patients, générer une balise <tr></tr> avec à l’intérieur des td qui contiendront les informations du patient.

L’objectif est que sous chrome quand vous faites « clic droit>Afficher le code source de la page », vous ne voyez pas de différence avec ce que vous avez fait pour le TP3.

Pour vous aider à commencer, vous pouvez tester ce code dans un autre fichier test.php et ajouter au fur et à mesure.

<?php **include** 'data.php'; ?>;

<table>

<?php **foreach**($patients **as** $patient) { ?>

<tr>

<td><?php **echo** $patient['id']; ?></td>

<td><?php **echo** $patient['prenom']; ?></td>

</tr>

<?php } ?>

</table>

**Quelques indications :**

* *Pour la date de naissance, le « birthday » du tableau correspond au timestamp. Il doit être converti en date visible grâce à la fonction date.*
* *Dans la balise <a> correspondant au bouton edit vert, ajouter l’attribut* href=**"formpatient.php?id=**<?php **echo** $patient['id'];?>**"**

*($patient est à remplacé par la variable que vous avez mise)*

*Nous créerons* **formpatient.php** *plus tard.*

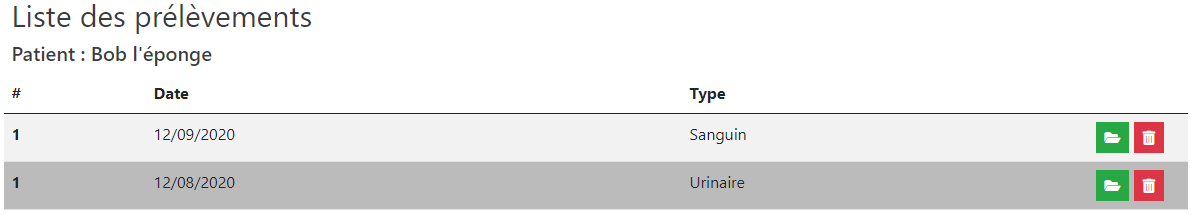
*De même pour les boutons delete, ajouter l’attribut*

href=**"deletepatient.php?id=**<?php **echo** $patient['id'];?>**"**

* *A la place de la balise <tr> écrire*<tr onclick=**"location.href='listeprelevements.php?id=**<?php **echo** $patient['id'];?>**';"**>  
  *Lorsque l’utilisateur clique sur la ligne, ça va exécuter cette ligne de code javascript, qui va ouvrir la page listedossiers.php , de la même manière qu’un lien hypertext.*
* *En dessous du tableau, ajouter un lien hypertexte (en forme de bouton bootstrap) « Ajouter un nouveau patient » vers la page formpatient.php*

1. **Création de la table prélèvement**

Créer listeprelevements.php et copier le code de la page du tableau ci-dessous, que vous avez créé au TP précédent (avec toute la page standard)



Ajouter dans ce fichier **include** 'data.php';

En utilisant l’alternance HTML-PHP et en utilisant une boucle foreach modifier le code pour que l’affichage du tableau dépende du contenu de $prelevements ainsi que l’identifiant du patient passé dans l’URL (obtenu par $\_GET['id']).

**Quelques indications :**

* Pour avoir les informations du patient dans cette page, écrire simplement $patient=$patients[$\_GET['id']-1] ; (*Le -1 est du au fait que l’id du premier patient est 1 mais que les indices des tableaux commencent à 0 en PHP)*
* *A chaque itération, tester si le champ idPatient vaut* $\_GET['id'], et n’afficher la ligne que si c’est vrai. Tester que quand on lance cette page depuis listpatients.php on a bien les prelevements du patient sur lequel nous avons cliqué.
* *Dans la balise <a> correspondant au bouton dossier vert, ajouter l’attribut* href=**"formprelevement.php?id=**<?php **echo** $prelevement['id'];?>**"**

*De même pour les boutons delete ajouter l’attribut*

href=**"deleteprelevement.php?id=**<?php **echo** $prelevement['id'];?>**"**

* *Ajouter un lien hypertext « Revenir à la liste des patients » en forme de bouton bootstrap en haut de la page*
* *En dessous du tableau, ajouter un lien hypertexte (en forme de bouton bootstrap) « Ajouter un nouveau Prélèvement » vers la page* **"formprelevement.php?idPatient=**<?php **echo** $\_GET['id']; ?>**"**

1. **Le formulaire patient**

Créer le fichier formpatient.php et mettre le code HTML du TP précédent pour la page de ce formulaire (avec toute la page standard).



L’objectif va être d’utiliser ce même formulaire soit pour ajouter un patient, soit pour éditer les informations d’un patient. Si c’est pour éditer un patient il faut remplir la valeur par défaut des champs par les valeurs du patient.

Dans l’url si l’utilisateur a écrit « formpatient.php » on sera en mode ajout

et si on met « formpatient.php?id=3 » on sera en mode edition du patient 3 et on devra donc compléter avec les valeurs par défaut.

En haut du fichier, mettre un bloc PHP avec à l’intérieur :

<?php

**include** 'data.php';

**if**(!**isset**($\_GET["id"])) { //Si il n'y a pas d'id dans l'url on est en mode Ajout

$patient=**array**('id'=>"",'sexe'=>0,'prenom'=>"",'nom'=>"",'email'=>"",'birthday'=>"");// Valeur par défaut des champs

//On déclare des variables pour les différents textes qui vont être dépendant

//du type de formualaire (Ajoute/edition)

$h2="Ajout d'un nouveau patient";

$action="addPatient.php";

$TexteButton="Ajouter";

}

**else** { //Si il y a un id dans l'url on est en mode Edition

$patient=$patients[$\_GET['id']-1]; // Valeur par défaut des champs

$h2="Editer les informations patient";

$action="editPatient.php";

$TexteButton="Sauvegarder";

}

?>

Nous avons ainsi créer des variables qui auront des valeurs différentes suivant si on est en mode ajout ou édition.

* Mettre le texte $h2 dans la balise <h2> de votre document *(Pour rappel vous devez écrire <?php echo $h2; ?> )*
* Dans la balise form, preciser method=**"POST"** action=**"**<?php **echo** $action; ?>**"**
* De la même manière remplacer le texte du bouton Ajouter/Sauvegarder
* Vérifier que les différents champs input on les attributs name valant respectivement (sexe, prenom, nom, email et birthday)
* Pour les 3 champs, prenom, nom, email mettre leur attribue value à $patient['prenom'], $patient['nom'],$patient['email'];
* Pour le input du radio Homme, mettre les attributs

value=**"1"** <?php **echo** $patient['sexe']==1?"checked":""; ?>

La syntaxe $d= ($condition?$a:$b) signifie si $condition vrai alors $d vaut $a sinon $d vaut $b.

Faire pareil pour le radio Femme avec la valeur 2.

* Pour le champ birthday mettre value=**"**<?php **echo** **date**("Y-m-d",$patient['birthday']);?>**"**

1. **Formulaire prélèvement**

Faire la même chose que la question 3 pour le formulaire prélèvement

Indication :

* Les adresse pour les « action » seront "addPrelevement.php"; (si mode ajout) "editPrelevement.php" (si mode edition);
* En mode edition vous aurez $idPatient=$prelevements[$\_GET['id']-1]['idPatient'];   
  En mode ajout vous aurez $idPatient=$\_GET['idPatient'];

Vous devez ajouter un <input type=**"hidden"** name=**"idPatient"** value=**"**<?php **echo** $idPatient; ?>**"**>

Ce champ peut être mis n’importe où dans le contenu de la balise <form></form>

C’est un champ qui n’apparaitra pas mais qui permettra de faire passer l’idPatient à la page addPrelevement.php ou editPrelevement.php .

Vous aurez aussi besoin de l’idPatient pour écrire les bons nom/prénom de patient dans les champs.

**5 Formulaire login**

*L’objectif de cette partie est de sécuriser toutes les pages du site. L’utilisateur doit rentrer le bon login et le bon mot de passe, si c’est le cas on envoie un cookie avec l’identifiant utilisateur. Sur toutes les pages du site on va tester si ce cookie existe sinon on renvoie sur la page login.php. Si l’utilisateur met un faut login ou un faut mot de passe il reste sur la page login et un message d’erreur apparait. Si l’utilisateur met le bon mot de passe, il est alors redirigé vers la page listpatient.php*

Modifier le nom de fichier du formulaire login en login.php

Dans la balise form mettre les attributs action="login.php" method="POST". Vos deux champs <input> doivent avoir pour name username et password.

Mettre le code suivant juste au-dessus du formulaire

<?php

**setcookie**("MyId","-1",**time**()-3600); // Pour effacer le cookie s'il existe (on met une date de péremption avant l’heure actuel.)

**setcookie**("typeCompte","",**time**()-3600);

**if**(**isset**($\_POST["username"])) {

**if**($\_POST["username"]=="toto" && $\_POST["password"]=="123") {

**setcookie**("MyId","1",**time**()+30\*24\*3600);

**setcookie**("typeCompte","medecin",**time**()+30\*24\*3600);

**header**('Location:listpatients.php');

}

**else** { ?>

<div class=**"alert alert-danger"** role=**"alert"**>

**Login ou mot de passe incorrect.**

</div>

<?php

}

}

?>

L’objectif est que si l’utilisateur rentre le mauvais login, il aura un message d’erreur qui apparait. Et il lui est demandé de saisir le login de nouveau. Si le login est bon **header**('Location:listpatients.php'); redirige vers pas page listpatients.php. Avant de le rediriger, pour se souvenir que l’utilisateur est logué nous envoyons un cookie avec son identifiant.

Vous remarquez que c’est la même page qui envoie et réceptionne le formulaire. Quand on arrive sur cette page, soit vous ouvrez la page pour la première fois et dans ce cas $\_POST[‘username’] n’existe pas, soit vous arriver sur cette page suite à un appui du bouton envoyer et dans ce cas $\_POST[‘username’] vaut le login saisie par l’utilisateur

Pour tester, mettre un print\_r[$\_COOKIE] ; dans la page listpatients.php . Si vous charger la page sans vous êtes logué avant $\_COOKIE doit être vide. Si vous charger la page après vous êtes logué $\_COOKIE doit contenir MyId et TypeCompte

A partir du moment où on arrive sur la page login, nous effacons le login pour deloguer l’utilisateur (tester en revenant sur listpatients.php et en affichant print\_r[$\_COOKIE] )

Toutes les pages de patients que nous avons vu aujourd’hui ne doivent pas s’afficher si l’utilisateur n’est pas logué en médecin.

Dans data.php, ajouter :

**if**(!**isset**($\_COOKIE['typeCompte']) || $\_COOKIE['typeCompte']!="medecin")

**header**('Location:login.php');

Ce code sera éxécuté dans toutes les pages, et redirigera vers la page login, si l’utilisateur n’est pas logué ou si type compte n’est pas médecin.

Bien tester ce code et bien le comprendre, il sera modifié dans le prochain TP.