



Création d'une application PHP pour une association («InfoScreen Manager V2 »)

Objectif

Etre capable de fournir au client la version V2 de l'application.

Rappels

1. Munis de vos identifiants vous pouvez accéder à votre base de données MariaDB via l'interface **phpmyadmin** : <https://obiwan.univ-brest.fr/phpmyadmin>
2. Vous pouvez déposer les fichiers de votre application Web sur le serveur obiwan grâce à un logiciel client FTP (ex : Filezilla, psftp, FireFTP...) et aux identifiants de connexion fournis pendant les séances de TP.
3. Dans un navigateur Web, vous pouvez visualiser votre page d'accueil grâce à l'URL : <https://obiwan.univ-brest.fr/~e2xxxxxxx/v2/>
4. A l'aide de vos scripts PHP déposés sur le serveur obiwan (via le protocole FTP), vous pouvez réaliser des requêtes SQL et donc travailler avec les données de la base de données MariaDB.

Gestion des versions et des sauvegardes des fichiers de votre application Web

Avant de commencer à travailler sur la version 2 de votre application Web, pensez à copier, via votre accès FTP, tous les fichiers de votre répertoire v1 sur le serveur obiwan.univ-brest.fr dans votre répertoire v2.

La version v2 de votre application sera donc constituée de la version v1 augmentée des fonctionnalités demandées pour la version v2 !

Par ailleurs, faites **au moins 3 copies de sauvegarde** sur des supports différents de :

- l'export de votre base de données MariaDB avec son jeu de données,
- du fichiers de vos requêtes pour l'application Web,
- et de tous les fichiers de la version v1 (puis de votre version v2, même en cours de création) !

Réalisation de la version V2 de l'application PHP

Il s'agit de développer la deuxième version de l'application pour la présenter au client.

V2	V2.0	Visualisation (R), par un administrateur, de la liste des profils/comptes de tous les utilisateurs + Activation / désactivation d'un profil (U) par un gestionnaire + Déconnexion de l'espace d'administration
	V2.1	V2.0 + Dans l'espace rédacteur, affichage de <u>toutes</u> les catégories et de leurs informations associées (intitulé + auteur), s'il y en a + ajout d'une catégorie + suppression d'une catégorie
	V2.2	V2.1 + Gestion RD des actualités dans l'espace gestionnaire (→ affichage de toutes les actualités et suppression d'une actualité)

Rappel : vous avez déjà préparé et testé les requêtes nécessaires pour la version 2 de l'application Web (cf fiche des requêtes).

Préparation de la version V2

Après avoir travaillé sur la partie publique du site Web (Front Office), vous allez aussi travailler sur la partie cachée de l'application (Back Office).

Pour vous aider à travailler sur l'espace d'administration, on peut imaginer le menu pour la navigation dans les rubriques de cet espace privé. Ce travail de réflexion vous aidera à déterminer les fonctionnalités et les pages à développer.

Exemple de menu pour l'espace Rédacteur :

Accueil & Mon profil	Gestion des actualités	Gestion des catégories	Gestion des visuels
Gestion des URL	Gestion des informations	Déconnexion	

Exemple de menu pour l'espace Gestionnaire :

Accueil & Mon profil	Comptes	Actualités	Catégories / informations	URL	Déconnexion
-------------------------	---------	------------	------------------------------	-----	-------------

Préparation de la version V2.0

En vous aidant des manipulations faites précédemment, dans l'espace d'administration des gestionnaires, **sur la page de gestion des comptes/profils**, au-dessus de l'affichage du tableau de tous les comptes/profils qui existent dans la base de données (sans oublier la validité et le statut/rôle des profils), il faut prévoir un formulaire pour récupérer le pseudo du compte associé au profil à modifier puis traiter cette saisie pour modifier la **validité** du profil concerné.



Pour rappel : quel est le **type de fonction PHP** qui permet de récupérer les résultats ? (cf [ANNEXE 1](#))

- Puis, grâce au tableau des comptes/profils et de la Capsule vidéo sur Moodle,

- dans un premier temps :

- créez un **formulaire** avec un champ de saisie puis sur validation de la donnée saisie (ex : le pseudo de l'utilisateur dont on veut activer le profil), dans le fichier **comptes_action.php**, constituez une requête pour activer / désactiver (c'est à dire modifier la validité) du profil de l'utilisateur choisi. Vérifiez que cela fonctionne !
- Vous pouvez compléter la page **comptes_action.php** en vérifiant si le pseudo saisi existe bien dans la base, **avant de préparer la requête** pour modifier le profil.

- dans un deuxième temps :

- modifiez le **formulaire** pour ajouter un bouton d'activation / de désactivation d'un profil sur la ligne de chaque profil (cf Capsule vidéo en ligne sur Moodle) au lieu d'un champ de saisie d'une chaîne de caractères pour choisir le pseudo du profil à activer / désactiver (<=> dont on veut modifier la validité). Testez à nouveau le fonctionnement de ce nouveau formulaire.

- dans un troisième temps :

- une fois la modification traitée, réalisez une redirection vers la page de gestion des comptes/profils **pour constater la modification des profils et ajouter la déconnexion de l'espace d'administration.**

Préparation de la version V2.1

Dans l'espace d'administration, dans la rubrique de gestion des catégories (page **admin_categories.php**), il faut afficher, dans un tableau, toutes les catégories et leurs informations (+ intitulé + auteur), s'il y en a (cf fiche des requêtes SQL). Puis, il faut prévoir des **formulaires** pour permettre à l'utilisateur connecté d'ajouter une catégorie / de supprimer une catégorie.

- Conseil : vous pouvez travailler par étapes successives et prévoir,
- dans la partie administration du site Web,
 - **dans un premier temps**, d'afficher dans un tableau TOUTES les catégories et leurs informations (+ intitulé + auteur), s'il y en a
 - **Puis dans un second temps**, de rajouter un formulaire, au-dessus ou sur chaque ligne du tableau récapitulatif des catégories, pour supprimer les données de la catégorie choisie.
 - **Puis dans un troisième temps**, de réaliser une redirection vers la page de gestion des catégories pour constater le résultat de la manipulation.

- Puis toujours dans la partie privée du site Web,
 - Dans un premier temps, mettre en place, au-dessus du tableau récapitulatif des catégories, le formulaire d'ajout d'une catégorie qui demande à la personne connectée de saisir son intitulé et son texte et effectue l'ajout demandé.
 - Puis dans un second temps, réaliser une redirection vers la page de gestion des catégories pour constater le résultat de la manipulation.

Préparation de la version V2.2 (NON DEMANDEE!!)

Dans l'espace gestionnaire, il faut permettre, dans une page gestionnaire_actualites.php la gestion **RD** des Actualités.

Conseil : vous pouvez prévoir

- l'affichage de toutes les actualités présentes dans la base de données,
- puis sous la liste de toutes actualités, un formulaire pour choisir une actualité afin de la supprimer.



Vérifiez ensuite les manipulations effectuées sur la page d'accueil de la partie publique (Front Office) de l'application Web.

RAPPEL IMPORTANT : dans les annexes, vous trouverez à nouveau la description des fonctions PHP de type **fetch** !

RAPPEL 1 : récupération de l'identifiant maximum dans la table des actualités – Utilisation de la fonction `fetch_object()` :

```
$sql="SELECT MAX(act_id) AS maximum FROM t_actualite_act;";

if (!$resultat=$mysqli->query($sql))
{
    // La requête a échoué
    echo "Error: la requête a échoué \n";
    echo "Query: " . $sql . "\n";
    echo "Errno: " . $mysqli->errno . "\n";
    echo "Error: " . $mysqli->error . "\n";
}
else
{
    $obj=$resultat->fetch_object();
    $lemax=$obj->maximum;
    echo($lemax);
    //On peut ensuite incrémenter la variable $lemax pour ajouter
    // une nouvelle actualité
    // ...
}
```

On peut bien sûr utiliser une autre des 4 fonctions PHP de type `fetch` pour récupérer la valeur maximale de l'identifiant de la table de gestion des actualités.

RAPPEL 2 : récupération de l'identifiant maximum dans la table des actualités – Utilisation de la fonction `fetch_assoc()` :

```
$sql="SELECT MAX(act_id) AS maximum FROM t_actuale_act;";

if (!$resultat=$mysqli->query($sql))
{
    // La requête a échoué

    echo "Error: la requête a échoué \n";
    echo "Query: " . $sql . "\n";
    echo "Errno: " . $mysqli->errno . "\n";
    echo "Error: " . $mysqli->error . "\n";
}
else
{
    $ligne=$resultat->fetch_assoc();
    $lemax=$ligne["maximum"];
    echo($lemax);
    //On peut ensuite incrémenter la variable $lemax pour ajouter
    // une nouvelle actualité
}
```

Présentation de la version V2



Avant la présentation de la dernière version de votre application Web, n'oubliez pas de la tester en faisant **BEAUCOUP** de manipulations **réalistes** !

ANNEXES

Annexe 1 : les formulaires HTML

Code HTML	Code PHP associé
<p>Champ de saisie de type texte :</p> <pre><form action="action.php" method="post"> <fieldset> <legend>Données personnelles :</legend> <p>Pseudo : <input type="text" name="pseudo" placeholder="pseudo" maxlength="20" required="required" /> </p> </fieldset> <p><input type="submit" value="Valider"></p> </form></pre>	<pre><?php if (\$_POST["pseudo"]) { echo htmlspecialchars(\$_POST["pseudo"]); } ?></pre>
<p>Champ de saisie de type mot de passe :</p> <pre><form action="action.php" method="post"> <p>Mdp : <input type="password" name="mdp" /></p> <p><input type="submit" value="Valider"></p> </form></pre>	<pre><?php if (\$_POST["mdp"]) { echo htmlspecialchars(\$_POST["mdp"]); } ?></pre>
<p>Liste déroulante à choix unique :</p> <pre><form action="action.php" method="post"> <select name="pays"> <option value="F">France</option> <option value="E">Espagne</option> </select> <p><input type="submit" value="Valider"></p> </form></pre>	<pre><?php if(\$_POST["pays"]) { echo \$_POST["pays"]; } ?></pre>
<p>Liste de cases à cocher :</p> <pre><form action="action.php" method="post"> <input type="checkbox" name="pays[]" value="F" /> France
 <input type="checkbox" name="pays[]" value="E" /> Espagne
 ... </form></pre>	<pre><?php if(\$_POST["pays"]) { //Affiche le contenu du tableau print_r(\$_POST["pays"]); } ?></pre>
<p>Liste de boutons radio :</p> <pre><form action="action.php" method="post"> <input type="radio" name="pays" value="F" /> France
 <input type="radio" name="pays" value="E" /> Espagne
 ... </form></pre>	<pre><?php if(\$_POST["pays"]) { echo \$_POST["pays"]; } ?></pre>

Input type :

<ul style="list-style-type: none"> • text • password • submit • radio • checkbox • button • color 	<ul style="list-style-type: none"> • date • datetime • datetime-local • email • month • number • range 	<ul style="list-style-type: none"> • search • tel • time • url • week
--	---	--

Annexe 2 : les fonctions PHP de type « fetch »

A noter : le verbe anglais « to fetch » signifie trouver, rechercher, prélever, en français.

Pour traiter en PHP le (ou les) résultats obtenus suite à l'exécution d'une requête via l'appel de la fonction **query()**, via l'API PHP mysqli, il faut utiliser les fonctions de type « fetch » de [la classe mysqli_result](#) mises à notre disposition.

Attribut num_rows :

mysqli_result::\$num_rows → Contient le nombre de lignes dans un résultat

Pour en savoir plus sur cette fonction, voir le manuel en ligne de PHP à ce sujet : <http://php.net/manual/fr/mysqli-result.num-rows.php>

Cf exemple ci-dessous

→ fonction correspondante de l'API mysqli procédurale :
`int mysqli_num_rows (mysqli_result $result);`

Fonction fetch_assoc() :

mysqli_result::fetch_assoc → Récupère une ligne de résultat sous forme de tableau associatif ou **NULL** s'il n'y a plus de ligne.

Pour en savoir plus sur cette fonction, voir le manuel en ligne de PHP à ce sujet : <http://php.net/manual/fr/mysqli-result.fetch-assoc.php>

Exemple :

```
$query = "SELECT Name, CountryCode FROM City ORDER by ID DESC LIMIT 50,5";  
  
if ($result = $mysqli->query($query)) {  
    /* Détermine le nombre de lignes du jeu de résultats */  
    $row_cnt = $result->num_rows;  
    printf("Le jeu de résultats a %d lignes.\n", $row_cnt);  
  
    /* Récupère un tableau associatif */  
    while ($row = $result->fetch_assoc()) {  
        printf ("%s (%s)\n", $row["Name"], $row["CountryCode"]);  
    }  
  
    /* Libération des résultats */  
    $result->free();  
}
```

Nom des colonnes du résultat

→ fonction correspondante de l'API mysqli procédurale :
`array mysqli_fetch_assoc (mysqli_result $result);`

Fonction `fetch_array()` :

mysqli_result::fetch_array → Retourne une ligne de résultat sous la forme d'un tableau associatif, d'un tableau indexé, ou les deux

Pour en savoir plus sur cette fonction, voir le manuel en ligne de PHP à ce sujet :

<http://php.net/manual/fr/mysqli-result.fetch-array.php>

Exemple :

```
$query = "SELECT Name, CountryCode FROM City ORDER by ID LIMIT 3";  
$result = $mysqli->query($query);
```

```
/* Tableau numérique */
```

```
$row = $result->fetch_array(MYSQLI_NUM);  
printf ("%s (%s)\n", $row[0], $row[1]);
```

Indices à partir de 0 correspondant à l'ordre des colonnes du résultat

```
/* Tableau associatif */
```

```
$row = $result->fetch_array(MYSQLI_ASSOC);  
printf ("%s (%s)\n", $row["Name"], $row["CountryCode"]);
```

Nom des colonnes du résultat

```
/* Tableau associatif et numérique */
```

```
$row = $result->fetch_array(MYSQLI_BOTH);  
printf ("%s (%s)\n", $row[0], $row["CountryCode"]);
```

```
/* Libération des résultats */  
$result->free();
```

→ fonction correspondante de l'API **mysqli** procédurale :

```
mixed mysqli_fetch_array (mysqli_result $result [, int $resulttype = MYSQLI_BOTH]);
```

Fonction `fetch_object()` :

mysqli_result::fetch_object → Retourne la ligne courante du jeu de résultat sous forme d'objet dont les attributs représentent les noms des colonnes trouvés dans le jeu de résultats.

Pour en savoir plus sur cette fonction, voir le manuel en ligne de PHP à ce sujet :

<http://php.net/manual/fr/mysqli-result.fetch-object.php>

Exemple :

```
$query = "SELECT Name, CountryCode FROM City ORDER by ID DESC LIMIT 50,5";
```

```
if ($result = $mysqli->query($query)) {
```

```
    /* Récupère un tableau d'objets */
```

```
    while ($obj = $result->fetch_object()) {  
        printf ("%s (%s)\n", $obj->Name, $obj->CountryCode);  
    }
```

Attributs de l'objet \$obj correspondant aux noms des colonnes

```
}
```

→ fonction correspondante de l'API **mysqli** procédurale :

```
object mysqli_fetch_object ( mysqli_result $result [, string $class_name = "stdClass" [, array $params ]]);
```


Fonction `fetch_row()` :

mysqli_result::fetch_row → Récupère une ligne de données à partir du jeu de résultats et le retourne en tant que tableau indexé où chaque colonne est une case du tableau, celui-ci commençant à 0 (zéro). Chaque nouvel appel à `fetch_row()` retournera la prochaine ligne dans le jeu de résultats, ou **NULL** s'il n'y a plus de lignes.

Pour en savoir plus sur cette fonction, voir le manuel en ligne de PHP à ce sujet :

<http://php.net/manual/fr/mysqli-result.fetch-row.php>

Exemple :

```
$query = "SELECT Name, CountryCode FROM City ORDER by ID DESC LIMIT 50,5";  
  
if ($result = $mysqli->query($query)) {  
    /* Récupère un tableau indexé */  
    while ($row = $result->fetch_row()) {  
        printf ("%s (%s)\n", $row[0], $row[1]);  
    }  
}
```

Indices à partir de 0 correspondant à l'ordre des colonnes du résultat

→ fonction correspondante de l'API `mysqli` procédurale :

```
mixed mysqli_fetch_row ( mysqli_result $result);
```

Annexe 2 : les variables PHP superglobales

Les **variables superglobales** sont des variables internes qui sont toujours disponibles, quel que soit le contexte du script.

Parmi ces variables, on peut citer le tableau superglobal `$_POST` utilisé dans le cadre de la saisie des données de l'utilisateur via un formulaire.

Voici la liste des variables superglobales qui existent :

```
$GLOBALS  
$_SERVER  
$_GET  
$_POST  
$_FILES  
$_COOKIE  
$_SESSION  
$_REQUEST  
$_ENV
```

Pour en savoir plus sur ces variables particulières, reportez-vous au manuel PHP en ligne :

<http://php.net/manual/fr/language.variables.superglobals.php>