

Architecture applicative de l'application « Festival de musique version allégée » niveau III

Principes d'organisation de l'application PHP festival	1
Conventions de nommage	2
Schéma de l'interaction entre les différentes pages	3
Barre de menus	3
Menu Gestion établissements	3
Menu Attribution chambres	5
Structure de chaque page de l'application	5
Illustration détaillée de l'interaction entre les pages « Liste des établissements » et « Détail établissement »	5
Détail des choix pour la gestion des données et la gestion des erreurs	7
Gestion des données (fichier _gestionBase.inc.php)	7
Gestion des erreurs (fichier _controlesEtGestionErreurs.inc.php)	7

Principes d'organisation de l'application PHP festival

Le code PHP a été organisé de façon à respecter les contraintes suivantes :

- Les traitements ont été structurés selon les principes suivants :
 - o la gestion des données (thème C2.2) est isolée dans un fichier à inclure : ce fichier comporte les fonctions de manipulation des données et les fonctions de connexion ;
 - o la cinématique des cas d'utilisation (thème C2.3) et la gestion de l'affichage (thème C2.1) sont prises en charge dans les mêmes fichiers ;
 - o la gestion des erreurs est décrite dans une bibliothèque de fonctions ad hoc.
- Réaliser un codage pédagogique (pas de préoccupation d'optimisation du code).
- Utiliser le langage PHP de base sans avoir recours à un quelconque framework tierce.
- Séparer la présentation des informations de leur description : avoir recours à une feuille de style CSS pour la mise en forme afin que le langage HTML ne soit utilisé que pour décrire les informations.

Concernant la syntaxe du langage PHP, le lecteur est invité à télécharger le cours PHP proposé dans la publication L'affleure disponible sur le site du CERTA :

<http://www.reseaucerta.org/cotecours/cotecours.php?num=95>.

Organisation du stockage des fichiers de l'application

Nom
css
images
_controlesEtGestionErreurs.inc.php
_debut.inc.php
_gestionBase.inc.php
consultationAttributions.php
creationEtablissement.php
detailEtablissement.php
donnerNbChambres.php
index.php
listeEtablissements.php
modificationAttributions.php
modificationEtablissement.php
suppressionEtablissement.php

Le répertoire `css` contient la feuille de style.

Le répertoire `images` contient l'image utilisée dans l'en-tête des tableaux.

Le fichier `index.php` est le fichier de démarrage de l'application.

Les fichiers d'inclusion :

- `_debut.inc.php` est utilisé dans chaque page pour construire le titre et le menu de la page ;
- `_controlesEtGestionErreurs.inc.php` est une bibliothèque de fonctions utilisées pour la gestion des erreurs et les contrôles de saisie ;
- `_gestionBase.inc.php` contient les fonctions de connexion et les fonctions de manipulation de la base de données.

Remarque : les contrôles de saisie qui nécessitent un accès à la base ne figurent pas dans `_controlesEtGestionErreurs.inc.php` mais dans `_gestionBase.inc.php`.

Conventions de nommage

Item	Règle de nommage ¹
Variables	
tableau associatif	accès par le nom
jeu d'enregistrements	<code>\$rs</code> suivi du rôle
résultat du jeu d'enregistrements	tableau <code>\$lg</code> suivi du rôle
chaîne contenant une requête SQL	<code>\$req</code>
autre variable	pas de règle (excepté bien sûr le respect de la norme « Camel »)
Fonctions	
fonction retournant une requête SQL	<code>obtenirReq</code> suivi du rôle (exemple : <code>obtenirReqEtablissements</code> est le nom de la fonction qui retourne la requête permettant d'obtenir les id et noms des établissements)
fonction retournant une ligne ou une valeur	<code>obtenir</code> suivi du rôle (exemple : <code>obtenirDetailEtablissement</code> est le nom de la fonction qui retourne la ligne correspondant à l'établissement demandé)
fonction de création ou modification ou suppression	verbe <i>créer</i> ou <i>modifier</i> ou <i>supprimer</i> suivi du nom de la table. Si la fonction prend en charge plusieurs actions, on accole les noms des actions (exemple : <code>creerEtablissement</code>)
fonction de test	son nom sera formé de <i>estUn</i> ou <i>verifier</i> ou <i>existe</i> suivi du rôle
Fichiers	
fichier inclus (excepté les vues)	son nom commence par <code>_</code> et est suffixé <code>.inc.php</code>

Concernant le schéma de la base de données les règles d'écriture suivantes ont été appliquées :

- pas de blanc ni de caractère accentué dans les noms de table ou d'attribut ;
- chaque nom de table commence par une majuscule et est suivi de minuscules. Si elle est composée de deux mots, ils sont collés et distingués par une majuscule ;

¹ Tous les noms respectent la règle « Camel » qui est notamment utilisée pour la programmation en langage Java.

- chaque nom d'attribut est écrit en minuscule. S'il est composé de deux mots, ils sont collés et distingués par une majuscule. Le nom choisi pour l'attribut représente le rôle de son domaine dans la table ;
- une clef étrangère porte un nom significatif de son rôle dans la table.

Schéma de l'interaction entre les différentes pages

Barre de menus

Les menus correspondent aux cas d'utilisation ; chaque cas comporte plusieurs sous-cas.

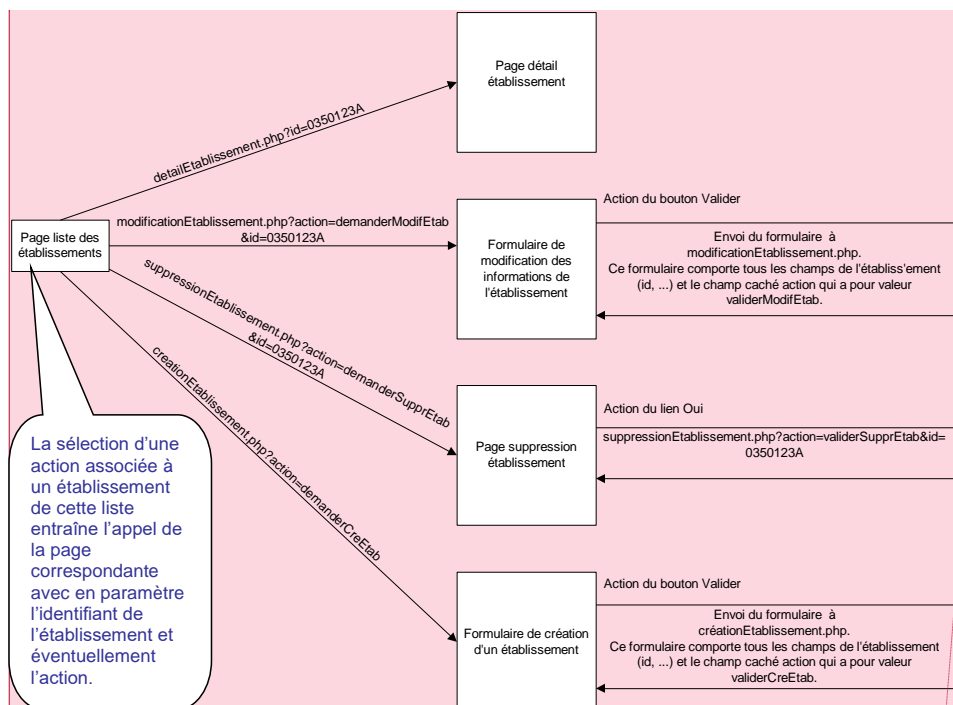
Accueil	Gestion établissements	Attributions chambres
-------------------------	--	---------------------------------------

Les menus sont gérés dans un tableau figurant dans le fichier `_debut.inc.php` (fichier inclus dans chaque page PHP, voir explications supra) :

```
<!-- Tableau contenant les menus -->
<table width="80%" cellpadding="0" cellspacing="0" class="tabMenu" align="center">
  <tr>
    <td class="menu"><a href="index.php">Accueil</a></td>
    <td class="menu"><a href="listeEtablissements.php">
      Gestion établissements</a></td>
    <td class="menu"><a href="consultationAttributions.php">
      Attributions chambres</a></td>
  </tr>
</table>
```

Menu Gestion établissements

Le choix de ce menu mène à la page `listeEtablissements.php`.

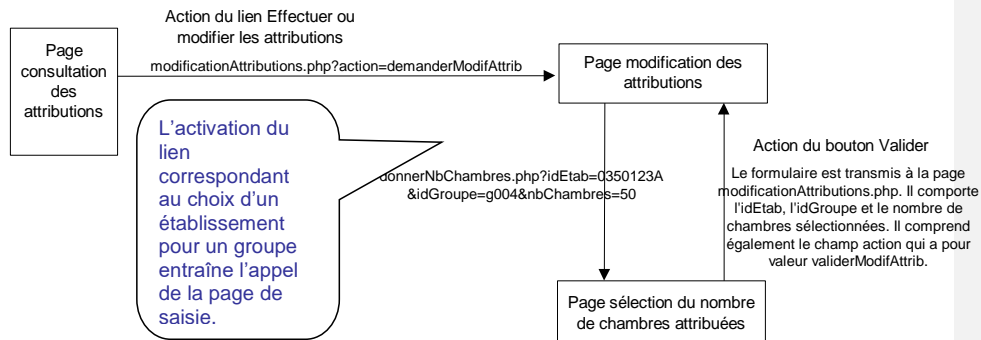


Commented [r1]:

Ce schéma est illustré avec le collège Lamartine (*id 0350123A*). Les liens Retour ne figurent pas dans la cinématique ci-dessus.

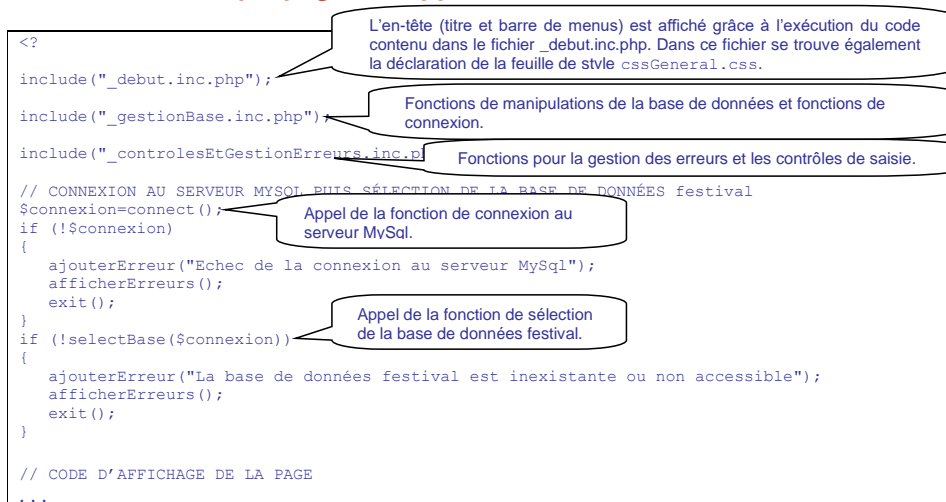
Menu Attribution chambres

Le choix de ce menu mène à la page `consultationAttributions.php`.



Ce schéma est illustré avec le collège Lamartine (*id 0350123A*) et le groupe *g004*.

Structure de chaque page de l'application



Toutes les pages sont construites selon cette structure à l'exception de la page `index.php` qui inclut uniquement le fichier `_debut.inc.php`. Ceci est logique car `index.php` est la page d'accueil et elle ne nécessite aucun accès à la base de données et ne peut produire aucune erreur donc les fichiers `_gestionBase.inc.php` et `_controlesEtGestionErreurs.inc.php` ne sont pas utiles.

Illustration détaillée de l'interaction entre les pages « Liste des établissements » et « Détail établissement »

L'illustration s'appuie sur la situation suivante : l'utilisateur a choisi le menu « Gestion des établissements » ; il dispose de la liste des établissements (page `listeEtablissements.php`) et il demande les détails concernant le collège de Moka (page `detailEtablissement.php`).

Page `listeEtablissements.php` :

L'utilisateur clique sur le lien Voir Détail de la ligne Collège Moka :

Etablissements			
Collège Lamartine	Voir détail		
Collège de Moka	Voir détail		
Collège Léonard de Vinci	Voir détail	Modifier	
Centre de rencontres internationales	Voir détail	Modifier	Supprimer
Création d'un établissement			

<http://127.0.0.1/festival/detailEtablissement.php?id=0350785N>

Gestion des données

Fonctions utilisées dans detailEtablissement.php :

```
function connect()
{
    $hote="localhost";
    $login="festival";
    $mdp="secret";
    return mysql_connect($hote,$login,$mdp);
}

function selectBase($connexion)
{
    $bd="festival";
    $query="SET CHARACTER SET utf8";
    // Modification du jeu de caractères de la connexion
    $res=mysql_query($query, $connexion);
    $ok=mysql_select_db($bd, $connexion);
    return $ok;
}

function obtenirDetailEtablissement($connexion, $id)
{
    $req="select * from Etablissement where id='$id'";
    $rsEtab=mysql_query($req,$connexion);
    return mysql_fetch_array($rsEtab);
}
```

Le fait de cliquer sur le lien nommé [Voir détail](#) permet l'appel de la page detailEtablissement.php avec comme paramètre l'identifiant de l'établissement.

L'action de suppression n'est possible que pour les établissements ne disposant pas de chambres attribuées.

Page detailEtablissement.php :

```
<?
include("_debut.inc.php");
include("_gestionBase.inc.php");
include("_controlesEtGestionErreurs.inc.php");

// CONNEXION AU SERVEUR MYSQL PUIS SÉLECTION DE LA BASE DE
DONNÉES festival
$connexion=mysql_connect();
if (!$connexion)
{
    ajouterErreur("Echec de la connexion au serveur MySql");
    afficherErreurs();
    exit();
}
if (!selectBase($connexion))
{
    ajouterErreur("La base de données festival est
inexistante ou non accessible");
    afficherErreurs();
    exit();
}

$id=$_REQUEST['id'];

// OBTENIR LE DÉTAIL DE L'ÉTABLISSEMENT SÉLECTIONNÉ
$lgEtab=obtenirDetailEtablissement($connexion, $id);
$nom=$lgEtab['nom'];
...
echo "
<table width='60%' cellspacing='0' cellpadding='0'
align='center' class='tabNonCadrille'>
<tr class='enTeteTabNonCadrille'>
<td colspan='3'>$nom</td>
</tr>
```

Festival Folklores du Monde
Hébergement des groupes

Accueil	Gestion établissements	Attribution
Collège de Moka		
Id:	0350785N	
Adresse:	2 avenue Alexandre Brizard BP 6	
Code postal:	35400	
Ville:	Saint-Malo	
Téléphone:	0299206990	
Email:		
Type:	Etablissement scolaire	
Responsable:	Monsieur Dupont Alain	
Offre:	20 chambre(s)	

Appel de la fonction de connexion au serveur MySQL.

Appel de la fonction de sélection de la base de données festival.

Appel d'une fonction d'accès à la base de données.

Détail des choix pour la gestion des données et la gestion des erreurs

Gestion des données (fichier `_gestionBase.inc.php`)

- Les interrogations de la base retournant une seule ligne sont entièrement prises en charge dans une fonction déportée ; cette fonction retourne alors le résultat dans un tableau (si plusieurs colonnes ont été demandées) ou dans une variable élémentaire.

Exemples :

`obtenirDetailEtablissement($connexion, $id)` : retourne un tableau contenant les données de l'établissement dont l'id a été fourni.

`obtenirNbEtab($connexion)` : retourne une variable élémentaire qui contient le nombre d'établissements.

- Les interrogations de la base pouvant retourner plus d'un enregistrement sont traitées ainsi :
 - constitution de la requête dans une fonction,
 - exécution de la requête et traitement du jeu d'enregistrements dans le code de la page appelante.

Exemple : `obtenirReqEtablissements`

Appel à la fonction pour constituer la requête :

```
$req=obtenirReqEtablissements(); // obtenirReqEtablissements est la fonction qui constitue
                                //la requête permettant d'obtenir la liste des établissements
```

Exécution de la requête :

```
$rsEtab=mysql_query($req,$connexion);
```

Traitement du jeu d'enregistrements :

```
$lgEtab=mysql_fetch_array($rsEtab);
```

```
while ($lgEtab!=FALSE)
```

```
{
    . . .
    $lgEtab=mysql_fetch_array($rsEtab);
}
```

- Les mises à jour au sens large (modification, insertion, suppression) sont entièrement réalisées dans des fonctions.

Exemple : `creerEtablissement(...)`

- Les fonctions de gestion de données comportent éventuellement des paramètres pour la construction de requêtes dynamiques. En outre, si la fonction traite plusieurs actions, par exemple pour les fonctions prenant en charge la création et la modification, un paramètre `$mode` (valant 'C' ou 'M') sera nécessaire.

Exemple : `estUnNomEtablissement($connexion, $mode, $id, $nom)`

Gestion des erreurs (fichier `_controlesEtGestionErreurs.inc.php`)

Principes de la bibliothèque de fonctions de gestion des erreurs

Par convention dans cette application, lorsqu'une erreur est détectée, un message d'erreur approprié est construit et fourni au système de gestion des erreurs. Ceci est simplifié par l'utilisation de la fonction `ajouterErreur`.

```
function ajouterErreur($msg)
{
    if (!isset($_REQUEST['erreurs']))
        $_REQUEST['erreurs']=array();
    $_REQUEST['erreurs'][]=$msg;
}
```

`$_REQUEST['erreurs']` est un tableau destiné à recevoir les différents messages d'erreur. Quand la 1^{ère} erreur survient après la soumission de la requête http, le tableau n'existe pas encore, il est donc créé.

Une fonction *nbErreurs* a été écrite pour retourner le nombre d'erreurs ; cela permet de tester le nombre d'erreurs avant d'appeler la fonction d'affichage des erreurs.

```
function nbErreurs()
{
    if (!isset($_REQUEST['erreurs']))
    {
        return 0;
    }
    else
    {
        return count($_REQUEST['erreurs']);
    }
}
```

La fonction d'affichage des erreurs parcourt le tableau des erreurs et les affiche les unes sous les autres.

```
function afficherErreurs()
{
    echo '<div class="msgErreur">';
    echo '<ul>';
    foreach($_REQUEST['erreurs'] as $erreur)
    {
        echo "<li>$erreur</li>";
    }
    echo '</ul>';
    echo '</div>';
}
```

Principes d'utilisation des fonctions de gestion d'erreurs

Nous illustrons ces principes grâce aux contrôles effectués sur le formulaire de modification d'un établissement.

```
// Fonction qui vérifie la saisie lors de la modification d'un établissement.
// Pour chaque champ non valide, un message est ajouté à la liste des erreurs
function verifierDonneesEtabM($connexion, $id, $nom, $adresseRue, $codePostal, $ville, $tel,
    $nomResponsable, $nombreChambresOffertes)
```

```
{
    if ($nom=="" || $adresseRue=="" || $codePostal=="" || $ville=="" ||
        $tel=="" || $nomResponsable=="" || $nombreChambresOffertes=="")
    {
        ajouterErreur("Chaque champ suivi du caractère * est obligatoire");
    }
    . . .
}
```

```
// Lors de la validation du formulaire : affichage des erreurs ou du message de confirmation
if ($action==validerModifEtab)
{
    if (nbErreurs()!=0)
    {
        afficherErreurs();
    }
    else
    {
        echo "
        <h5><center>La modification de l'établissement a été effectuée</center></h5>";
    }
}
```