



Requêtes SQL en langage PHP



TP PhpSqlPdoLMvc

Objectifs :

- interroger une base de données par l'intermédiaire d'une liaison client/serveur HTTP(web)
- utiliser un logiciel de gestion de bases de données avec interface graphique.
- s'initier à la programmation **côté serveur**
- interfacer les langages Html et Php et réaliser des pages web interactives
- s'initier au modèle **MVC Model View Controller**
- réaliser des requêtes préparées et paramétrées SQL en langage PHP avec l'API **PDO**
- réaliser un objet DAO (Data Access Object)

Mise en situation :

On décide de créer une interface web permettant d'effectuer différentes requêtes SQL sur la table jeu.

Travail sur wamp ou lamp local : votre site web est le dossier **www**
sur le serveur **192.168.168.168/sjobs** (si vous êtes Steve Jobs)
La page est découpée en cadres appelé iframes

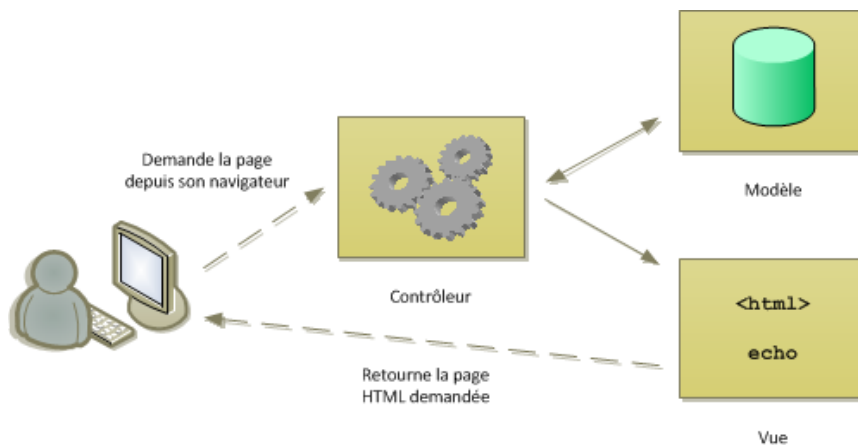
menuHaut.html
menuGauche.html
controller.php

Les différents liens des menus sont traités dans le fichier **controller.php**.

L'affichage est réalisé par les méthodes de la classe **Vue** incluse dans le fichier **view.php**

L'accès aux données est réalisé par le biais de la classe **Model**, elle-même composée de la classe **DaoJeu**.

C'est dans cette classe **DaoJeu** que doivent être implémentées les méthodes contenant l'exécution des différentes requêtes SQL



ceux habitant à
Marcq ville

ceux dont le score est < à
score

ceux dont le score est < à
et dont l'âge est < à
ok

Id n°
modifier
supprimer
Choix?
insérer

Tri par score Le + jeune	Tri par nom Le + vieux	Tri par âge score mini	Tri par ville score maxi		
Array ([cas] => score)					
Nom	Age	Ville	Score	id	passe
Moi	45	malo	3	26	50dc624377
Monique	18	Leers	3	2	1dc74ccc78
Quentin	78	quimper	3	17	21c9c6ba4c
Irene	17	Lille	3	11	651343906
Remi	54	Roubaix	5	18	2de6882f68
Nicolas	19	Roubaix	5	1	1714a0706
Gina	12	Roubaix	6	9	13325e7c06
suzie	21	tourcoing	6	19	186d5f9e39
Lucas	21	Lille	6	14	4a4f34ce43
Celine	21	Croix	6	5	1f27f01661
Benoit	20	Wattrelos	7	4	000b0a8b30
Henri	15	Lille	8	10	6aa8afa448
Damien	18	Marcq	8	6	46cc55884a
Jean	15	La Madeleine	9	12	355a40243e
Frank	12	Lens	9	8	7140fc028
Olive	44	Orlando	10	15	2332793c1f
popeye	78	paris	10	16	4bbd185b7b
Guillaume	18	Croix	56	25	647a32324c

Vous devez télécharger le dossier **tableJeuMvc** sur votre serveur et adapter les **login**, **mot de passe** et **nom** de votre base dans le fichier **login.php**.

La table de données jeu doit être présente dans votre base de données. Pour ceux qui l'ont perdu reprendre le fichier **jeu.txt**

1) Finalisation du fichier « controller.php » et des classes **Model** et **DaoJeu**

On teste la valeur de la variable *cas* dans un `switch` équivalent à une succession de `if`. Cette variable indique la source du clic

1.1) Après avoir étudié la liaison entre les fichiers *menuHaut.html*, *menuGauche.html* et le fichier *controller.php* afin de comprendre les différents états de la variable *cas*,

✓ compléter le `switch` pour les cas "nom", "ville", "age"

1.2) Un clic sur le bouton " **ville** " transmet l'information `cas= "villeprecise "`.

✓ Déterminer la méthode appelée dans la classe **DaoJeu** et compléter cette méthode.

1.3) Pour les cas " **scoreinferieur** " et " **scoreAge** " vous devez appeler, pour chaque cas, une méthode de la classe **Model** qui elle-même appelle une méthode de la classe **DaoJeu**. Le résultat retourné par la classe **Model** est transmis à la méthode `afficheTab` de la classe **Vue**.

Les méthodes des classes **Model** et **DaoJeu** sont à réaliser.

1.4) Même travail pour les cas **jeune, vieux, mini et maxi**.

1.5) Compléter la méthode **formulaireModif** de façon à ce que le formulaire soit rempli avec l'enregistrement à modifier. Vérifier que cela corresponde au cas= "modifier " et choix= "modifier "

1.6) Gérer le cas " VoiciLesModif " afin de modifier l'enregistrement sélectionné.

1.7) Réaliser le travail nécessaire pour le cas "modifier " et choix "supprimer "

1.8) Le cas " insere " provoque l'appel de la méthode **formulaireNouveau** de la classe **Vue**.

✓ Réaliser cette méthode qui affiche un formulaire vide permettant l'ajout d'un nouvel enregistrement.

Le cas doit être égal à " nouveau ".

1.9) Le cas **nouveau** doit provoquer l'appel de méthodes à réaliser dans les classes **Model**, **DaoJeu** afin de permettre l'insertion de l'enregistrement avec les valeurs contenues dans le formulaire.

1.10) Les noms des colonnes (première ligne) contenant les noms des champs de l'enregistrement doivent être des liens permettant d'effectuer un tri.

2) Javascript fichier: mesfonctions.js

2.1) Réaliser la fonction javascript **validation** qui vérifie que les champs des différents formulaires ne soient pas vides.

✓ Préciser l'avantage d'utiliser le javascript plutôt que le php pour cette fonction.

2.2) Etudier la fonction **controle** qui reçoit 3 paramètres. Expliquer son fonctionnement pour les différents formulaires.

3) Export

3.1) PDF: Placer un bouton pour imprimer. Cela donnera la possibilité d'imprimer dans un fichier au format PDF.

Utiliser la commande javascript `window.print`

3.2) CSV: Placer un bouton qui appelle la méthode `exportToCsv` de la classe **Vue**.

Cette méthode doit permettre le téléchargement d'un fichier contenant un enregistrement par ligne.

Pour chaque enregistrement les valeurs seront séparées par un point virgule.

```
Nicolas;19;Roubaix;5;1;1714af0706;
Monique;18;Leers;3;2;1dc74ccc78;
```

voir l'instruction `header`

```
header('Content-Type: text/csv');
header('Content-Disposition: attachment; filename=jeu.csv');
```

Cette méthode ne doit pas envoyer d'entête HTML.

Le contenu du fichier doit contenir les enregistrements de la dernière requête ou effectuer la même requête.

Il peut être intéressant de renvoyer l'ensemble des paramètres et d'avoir une méthode **Affiche** avec un paramètre " format " égal à HTML ou CSV.

Selon la valeur de la variable " format " cette méthode appellerait `afficheTab` ou `exportToCsv`

4) Ajax avec JQuery (mesfonctions.js(ligne 27) controleurAjax.php model.php daoJeu.php)

A partir du code permettant le chargement asynchrone de propositions de noms de ville , ajouter un formulaire permettant d'effectuer une recherche par nom.

Le remplissage de 2 caractères doit provoquer une recherche et un affichage de nom commençant par ces 2 lettres.

5) Export automatique PDF

Rechercher et mettre en œuvre une librairie permettant d'exporter automatiquement sans passer par la boîte de dialogue d'impression.

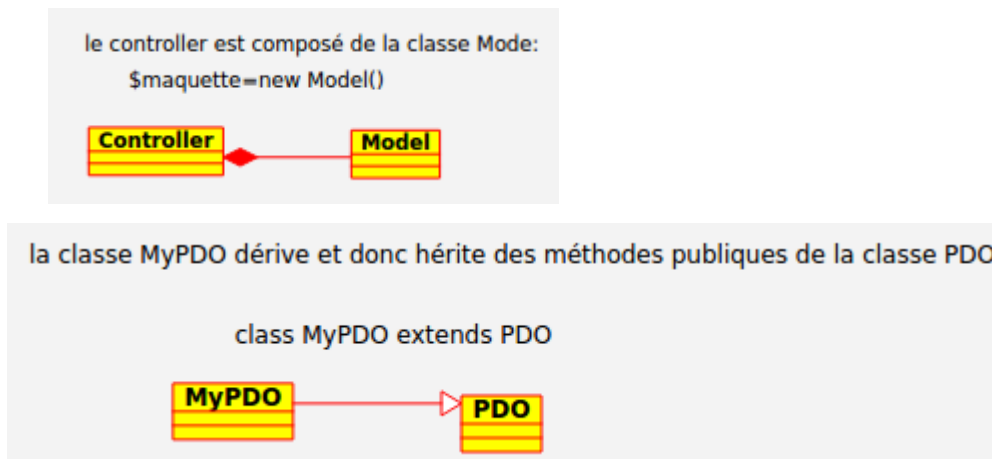
6) Style CSS (pas forcément joli :)

- ✓ Modifier le style du formulaire gauche afin que:
 - le formulaire du haut (ville) ait les angles du haut arrondis
 - le formulaire du bas (insérer) ait les angles du bas arrondis
- ✓ Modifier le style du formulaire haut afin que:
 - la police du lien survolé soit de couleur blanche et que la couleur de fond soit " navy " et de taille normale

7) Diagramme de classes

Chaque entité est représentée par un rectangle contenant son nom.

Sur un diagramme complet, devraient apparaître les attributs et méthodes. Dans un premier temps nous ne les représenterons pas: nous nous intéresserons qu'aux relations entre les classes.



- ✓ Représenter graphiquement le controller et les classes **Vue**, **Model**, **DaoJeu**, **MyPDO** et **PDO** et leurs relations.
- Pour représenter les diagrammes de la modélisation UML, on utilise un AGL (atelier de génie logiciel: Umbrello, ArgoUml, BoUml...)
- Certains IDE (environnement de développement intègre un outil Visual Studio, Eclipse..)
- Sur le serveur snir2012, on dispose de Visual Paradigme pour Windows

Aide : L'aide php se trouve sur <http://fr.php.net/> <http://php.net/manual/fr/book.spl.php>
Liste des fonctions Php sur <http://php.net/manual/fr/indexes.functions.php>

Exemple: <http://btsirinfo.free.fr/aide/base/>
<http://192.168.168.168/sflament/mvcJeu>

- Critères d'évaluation :**
- degré d'autonomie + niveau d'avancement du travail
 - synthèse des requêtes SQL, des fonctions php et javascript.
 - synthèse sur les classes **Vue**, **Model**, **DaoJeu**
 - rédaction du compte-rendu