

LOG4420
TP2
PHP, MySQL

Julien Gascon-Samson
École Polytechnique de Montréal

21 septembre 2009

Contexte

L'évolution d'Internet, de la puissance des machines et de l'interactivité des systèmes a inévitablement conduit, ces dernières années, au besoin de disposer de médias davantage interactifs afin d'offrir une expérience personnalisée à l'utilisateur.

Dans une perspective orientée-web, bien qu'intéressants, le langage HTML et les outils classiques auxiliaires tels que CSS ne réussissent pas à combler tous les besoins. Afin de bénéficier d'un niveau supérieur d'interactivité, plusieurs outils et langages "orienté-serveur" ont vu le jour, permettant ainsi de développer des applications qui dépassent le cadre du simple affichage statique de pages de contenu. Ces nouvelles technologies permettent maintenant de générer de façon complètement dynamique en totalité ou en partie les pages de contenu.

De plus, aujourd'hui, afin d'accomoder l'énorme volume de données qui sont traitées, des systèmes de gestion de bases de données évolués ont vu le jour. Les données ne sont désormais plus quelque chose de statique, mais bien une entité dynamique qu'il est possible de consulter et modifier à volonté, notamment grâce à l'introduction du Web 2.0.

Présentation du travail

Dans ce travail, nous vous demanderons d'utiliser le langage de programmation PHP et le système de gestion de bases de données MySQL pour réaliser le projet proposé dans l'énoncé. Tel que décrit, vous devrez prendre en considération que l'utilisateur de votre application web pourra prendre plusieurs rôles :

- visiteur (non authentifié) ;
- membre du club (authentifié) ;
- administrateur du site (authentifié).

Les différents rôles sont décrits de manière plus détaillée dans les sections qui suivent.

Visiteur

Le rôle du visiteur est très limité. Le visiteur a accès à la page d'accueil du site et au formulaire d'inscription. Il peut également, à titre informatif seulement, consulter la liste des matchs et faire apparaître les détails de chacun, mais ne peut pas effectuer d'achat. Pour avoir accès à plus de fonctionnalités, s'il possède un compte, il peut se connecter en utilisant le formulaire de connexion en haut à droite.

Ce rôle ressemble beaucoup à ce que vous avez fait à présent, en y ajoutant la liste des matchs et leur affichage détaillé.

Membre du club

Ce rôle nécessite une authentification préalable. Comme le visiteur, il peut voir la liste des matchs et l'affichage détaillé pour chacun. À partir de la page d'affichage détaillé, le membre a la possibilité de placer une commande de billets pour ce match : il entre le nombre de billets désirés pour le match et confirme l'ajout. **Attention : il faut s'assurer que la disponibilité des sièges pour le match choisi à l'aréna choisi le permet !**

Les billets demandés sont ensuite placés dans un panier et sont **temporairement réservés**. Cela signifie qu'un autre membre ne pourra prendre ces places puisqu'elles auront été réservées. Évidemment, la réservation doit expirer au bout d'un certain temps puisque l'utilisateur peut décider de quitter le site sans confirmer l'achat ; dans ce cas, il faudrait qu'au bout d'un certain temps, la réservation devienne invalide et les places se libèrent pour un autre membre. Choisissez un délai raisonnable pour l'expiration de la réservation.

Lorsque l'utilisateur a effectué son magasinage et désire *passer à la caisse*, c'est-à-dire confirmer l'achat des billets dans son panier, il accède à une page "panier d'achat" qui contient tous les matchs placés dans son panier avec quelques informations supplémentaires pour chacun des matchs (équipes, aréna, nombre de billets, coût total pour tous les billets). L'utilisateur a la possibilité de retirer des items de son panier¹, de vider le panier (supprimer tous les items) et de confirmer l'achat de tous les items².

Lorsqu'un achat est effectué, la transaction doit évidemment être enregistrée et le nombre de places restantes doit être mis à jour pour le match et l'aréna en question. L'utilisateur peut accéder à une page qui liste l'historique de tous ses achats effectués par le passé.

Vous devez ajouter à la page d'inscription conçue au TP1 une liste de choix déroulante qui permet de choisir le thème préféré de l'utilisateur. Chaque thème est constitué d'un fichier de style CSS qui sera "intégré" dynamiquement aux pages du site. À cette fin, vous devez concevoir au minimum deux fichiers CSS différents correspondant à deux thèmes différents et donner la possibilité à l'utilisateur de choisir quel thème il désire utiliser parmi les thèmes disponibles. Le site **CSS Zen Garden**³ illustre très bien le concept de thèmes en CSS, sauf que l'exemple est un peu poussé à l'extrême... Dans le cadre de ce TP, quelques variations mineures démontrant que vous avez fait l'exercice suffiront amplement.

Enfin, le membre doit pouvoir mettre à jour ses renseignements personnels à n'importe quel moment avec une page conçue à cet effet. Les renseignements personnels modifiables sont les mêmes que ceux entrés à l'inscription (incluant le thème CSS souhaité).

Administrateur

Le rôle de l'administrateur est de voir au bon fonctionnement du site web. Ce dernier a la possibilité de gérer les arénas et les matchs :

- ajout et suppression d'arénas ; modification du nom et du nombre de places : ceci sous-entend que l'administrateur doit pouvoir visualiser la liste des arénas ainsi que le détail de chacun, qu'il peut modifier)
- ajout et suppression de matchs ; modification de la date, de l'aréna, des équipes en jeu, du prix

1. Dans le cas où le membre aurait placé plusieurs billets pour un même match dans son panier, il n'est pas nécessaire de pouvoir enlever seulement un nombre donné de billets, seule la fonction de retrait complet de tous les billets pour ce match est nécessaire

2. Bien que ça pourrait être très intéressant, nous ne manipulerons malheureusement pas les cartes de crédit dans le cadre de ce travail. Nous allons donc nous contenter d'un paiement sur l'honneur...

3. <http://www.csszengarden.com/>

Modèle de la base de données

Vous utiliserez MySQL pour ce travail. Nous vous fournissons le schéma (liste des tables) de la base de données que vous utiliserez. L'administrateur réseau du GIGL vous fournira un compte et une base de données SQL; vous recevrez les informations pertinentes par courriel. Nous vous fournissons également un fichier "backup" qui contient du code SQL permettant de préparer le schéma de votre base de données avec MySQL Administrator (le contenu de ce fichier se trouve en annexe).

Création du schéma

Lancer l'utilitaire MySQL Administrator situé dans le menu Programmation. Connectez-vous en suivant les étapes illustrées dans la figure 1. Procédez ensuite à la restauration du backup en suivant les étapes décrites dans la figure 2.



FIGURE 1 – Connexion à MySQL Administrator

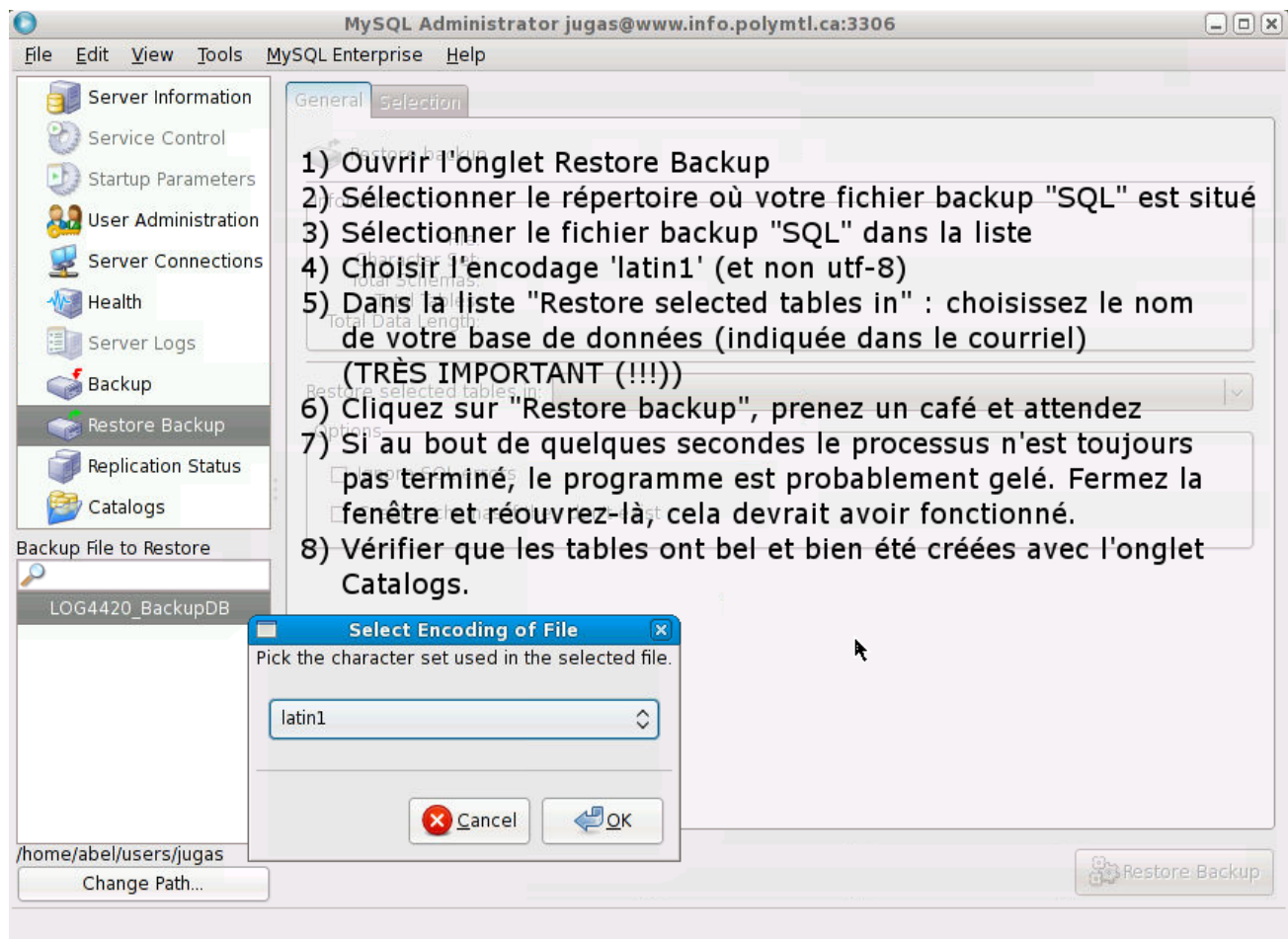


FIGURE 2 – Restauration du “backup” avec MySQL Administrator

Table arénas

Champ	Type	Description
id	entier	Index de la salle [clé primaire]
nom	texte	Nom de l'aréna
sieges	entier	Nombre de sièges au total dans cet aréna

Table matches

Champ	Type	Description
id	entier	Index du match [clé primaire]
description	texte	Equipe1 vs Equipe2
arena	entier	Id de l'aréna dans lequel le match se déroulera
date	date	Date du match
prix	entier	Prix des billets
places	entier	Nombre de places restantes pour ce match

Table reservations

Champ	Type	Description
id	entier	Index de la réservation [clé primaire]
utilisateur	entier	Id de l'utilisateur qui a effectué la réservation
match	entier	Id du match pour lequel porte la réservation
qte	entier	Quantité de billets réservés
expiration	date/heure	Moment où la réservation expirera

Noter qu'une entrée est ajoutée à la table réservation lorsqu'un membre place un match dans son panier. Tel que mentionné précédemment, pour éviter la situation où nous pourrions excéder accidentellement la capacité de l'aréna, il faut décrémenter le nombre de places restantes pour le match dès que la réservation est effectuée. Toutefois, si l'utilisateur décide de vider son panier ou si le délai de validité de la réservation expire, le nombre de places restantes sera "restauré" (donc incrémenté). Il pourrait être judicieux de définir une procédure de "nettoyage" des réservations expirées. Cette procédure pourrait être appelée dès qu'une nouvelle réservation survient ou est libérée, par exemple.

Une fois le panier d'achat confirmé, les réservations seront converties en achats.

Table achats

Champ	Type	Description
id	entier	Index de l'achat [clé primaire]
utilisateur	entier	Id de l'utilisateur qui a effectué l'achat
match	entier	Id du match pour lequel porte la l'achat
qte	entier	Quantité de billets achetés
date	date/heure	Moment où la réservation a été effectuée

Table utilisateurs

Champ	Type	Description
id	entier	Index de l'utilisateur [clé primaire]
role	entier	Rôle de l'utilisateur (1=Membre,2=Admin)
utilisateur	texte	Nom d'utilisateur (login)
motdepasse	texte	Mot de passe (password)
prenom	texte	Prénom de l'utilisateur
nom	texte	Nom de l'utilisateur
courriel	texte	Courriel de l'utilisateur
jour	entier	Jour de naissance de l'utilisateur
mois	entier	Mois de naissance de l'utilisateur
annee	entier	Année de naissance (entre 1900 et 2010 :P)
sexe	entier	Sexe de l'utilisateur (1=Homme,2=Femme,3=Génie logiciel)
theme	texte	Thème choisi par l'utilisateur

Évaluation

Il est important de porter une attention particulière aux points suivants :

- Tel que vu en classe et en accord avec votre architecture réalisée au TP1, bien découpler l'architecture multi-niveaux.
- Créer des objets pour représenter les concepts.
- Ne pas intégrer les requêtes SQL directement dans la vue ou la logique applicative.
- Porter attention à l'expiration de la session PHP.
- Éviter le “directory browsing”

Remettez votre travail avant le 23 octobre 2009 à 23h55. Vous devez compresser votre arborescence (qui inclura tous les fichiers appropriés) sous un fichier ZIP. Nommez votre fichier ZIP selon la convention suivante : `TP2_Matricule1_Matricule2.zip`. Téléversez votre fichier ZIP sur Moodle. Une pénalité de 10% par jour de retard s'applique. Le barème est le suivant :

- 4pts : les fonctionnalités du rôle **Administrateur** ont été réalisées tel que décrit dans l'énoncé.
- 4pts : les fonctionnalités du rôle **Membre** ont été réalisées tel que décrit dans l'énoncé
- 2pts : les fonctionnalités du rôle **Visiteur** ont été réalisées tel que décrit dans l'énoncé, y compris la prise en charge de l'inscription.
- 5pts : l'application web est conçue selon les principes d'une architecture multi-niveaux à 3 couches avec une bonne dissociation des différents éléments propres à ces couches. Les éléments sont, dans la mesure du possible, réutilisables.
- 2pts : prises en charge d'au moins deux thèmes CSS selon la préférence du membre.
- 1pt : il n'y a pas de redondance. Lorsque possible, au lieu de copier-coller du code, des fichiers PHP sont “inclus”.
- 1pt : gestion appropriée de l'expiration de la session PHP.
- 1pt : pas de “directory browsing”

Total : 20 points.

Bon travail !

Julien Gascon-Samson, chargé de travaux dirigés

Annexe

Voici le contenu du fichier “backup” (LOG4420_BackupDB.sql) à restaurer à l’aide de MySQL Administrator :

```
-- phpMyAdmin SQL Dump
-- version 3.1.2deb1ubuntu0.1
-- http://www.phpmyadmin.net
--
-- Serveur: localhost
-- Généré le : Lun 21 Septembre 2009 à 11:44
-- Version du serveur: 5.0.75
-- Version de PHP: 5.2.6-3ubuntu4.2

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";

--
-- Base de données: 'log4420a2009'
--
-- -----

--
-- Structure de la table 'achats'
--

CREATE TABLE IF NOT EXISTS 'achats' (
  'id' int(11) NOT NULL auto_increment,
  'utilisateur' int(11) NOT NULL,
  'match' int(11) NOT NULL,
  'qte' int(11) NOT NULL,
  'date' date NOT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

--
-- Contenu de la table 'achats'
--
-- -----

--
-- Structure de la table 'arenas'
```



```
CREATE TABLE IF NOT EXISTS 'arenas' (  
  'id' int(11) NOT NULL auto_increment,  
  'nom' varchar(50) NOT NULL,  
  'sieges' int(11) NOT NULL,  
  PRIMARY KEY ('id')  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

```
--  
-- Contenu de la table 'arenas'  
--
```

```
INSERT INTO 'arenas' ('id', 'nom', 'sieges') VALUES  
(1, 'Air Canada Centre', 85),  
(2, 'Centre Bell', 120),  
(3, 'Scotiabank Place', 180),  
(4, 'Verizon Center', 40),  
(5, 'HSBC Arena', 220),  
(6, 'Madison Square Garden', 70);
```

```
-- -----  
  
--  
-- Structure de la table 'matchs'  
--
```

```
CREATE TABLE IF NOT EXISTS 'matchs' (  
  'id' int(11) NOT NULL auto_increment,  
  'description' varchar(50) NOT NULL,  
  'arena' int(11) NOT NULL,  
  'date' date NOT NULL,  
  'prix' int(11) NOT NULL,  
  'places' int(11) NOT NULL,  
  PRIMARY KEY ('id')  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;
```

```
--  
-- Contenu de la table 'matchs'  
--
```

```
INSERT INTO 'matchs' ('id', 'description', 'arena', 'date', 'prix', 'places') VALUES  
(1, 'Canadiens de Montréal vs New Jersey Devils', 1, '2009-12-07', 100, 85),  
(2, 'Canadiens de Montréal vs New York Islanders', 3, '2009-12-08', 1850, 180),  
(3, 'Canadiens de Montréal vs Pittsburgh Penguins', 5, '2009-12-09', 5, 220),  
(4, 'Canadiens de Montréal vs Philadelphia Flyers', 3, '2009-12-10', 70, 180),  
(5, 'Canadiens de Montréal vs New York Rangers', 2, '2009-12-11', 450, 120),
```

```
(6, 'Canadiens de Montréal vs New York Islanders', 4, '2009-12-12', 222, 40),
(7, 'Canadiens de Montréal vs New Jersey Devils', 6, '2009-12-13', 199, 70),
(8, 'Canadiens de Montréal vs New Jersey Devils', 6, '2009-12-14', 110, 70),
(9, 'Canadiens de Montréal vs Pittsburgh Penguins', 1, '2009-12-15', 514, 85),
(10, 'Canadiens de Montréal vs New York Rangers', 5, '2009-12-16', 300, 220);
```

```
-- -----
```

```
--
```

```
-- Structure de la table 'reservations'
```

```
--
```

```
CREATE TABLE IF NOT EXISTS 'reservations' (
  'id' int(11) NOT NULL auto_increment,
  'utilisateur' int(11) NOT NULL,
  'match' int(11) NOT NULL,
  'qte' int(11) NOT NULL,
  'expiration' date NOT NULL,
  PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
--
```

```
-- Contenu de la table 'reservations'
```

```
--
```

```
-- -----
```

```
--
```

```
-- Structure de la table 'utilisateurs'
```

```
--
```

```
CREATE TABLE IF NOT EXISTS 'utilisateurs' (
  'id' int(11) NOT NULL auto_increment,
  'role' int(11) NOT NULL,
  'utilisateur' varchar(50) NOT NULL,
  'motdepasse' varchar(50) NOT NULL,
  'prenom' varchar(50) NOT NULL,
  'nom' varchar(50) NOT NULL,
  'courriel' varchar(50) NOT NULL,
  'jour' int(11) NOT NULL,
  'mois' int(11) NOT NULL,
  'annee' int(11) NOT NULL,
  'sexe' int(11) NOT NULL,
  'theme' varchar(50) NOT NULL,
```

```
PRIMARY KEY ('id')
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

--
-- Contenu de la table 'utilisateurs'
--

INSERT INTO 'utilisateurs' ('id', 'role', 'utilisateur', 'motdepasse', 'prenom',
'nom', 'courriel', 'jour', 'mois', 'annee', 'sexe', 'theme') VALUES
(1, 2, 'admin', 'admin', 'Yvon', 'Gagné', 'fireball33482738829@hotmail.com',
1, 1, 1901, 1, 'Standard');
```