

Projet 5 : POST

Utilisation de requêtes POST et de session

Enoncé

Implémenter les procédures de login, logout et enregistrement. Lors du login, une session doit être créée, qui sera entre autres utilisée pour limiter l'utilisation des RPC autres que login et enregistrement aux seuls utilisateurs enregistrés

Ressources

Les ressources suivantes sont disponibles sur le site http://cours.endor.be/2eme_ajax.

- common.js : librairie javascript de base
- img/* : différentes images pour le projet
- map.txt : carte à afficher
- db.sql, db.sqlite : source SQL de la DB et fichier SQLite associé

Ces fichiers de ressources ne doivent pas être modifiés (à l'exception de db.sqlite qui sera modifiée quand on travaillera sur la DB).

Cahier des tâches

JavaScript

L'appui sur login enverra une requête avec nom d'utilisateur et mot de passe. La valeur de retour sera testée pour décider si on affiche une erreur ou si la connexion est valide.

L'appui sur register affichera un formulaire d'enregistrement demandant nom d'utilisateur et mot de passe (en deux exemplaires). Lors de la validation, on enverra les données à un RPC et soit on effectuera la connexion sur le compte en question, soit on affichera un message d'erreur (si l'utilisateur existe déjà ou si les deux exemplaires de mot de passe sont différents).

L'appui sur logout enverra une requête à un RPC avant de revenir à la page d'accueil.

PHP

Les RPC d'affichage devront s'assurer que l'on est connecté avant de retourner quoi que ce soit.

Le RPC de login vérifiera l'utilisateur dans la DB, sauvera le uid dans la session et retournera OK si la connexion a réussi ou retournera ERR en cas de problème

Le RPC d'enregistrement créera l'utilisateur et sauvera le uid avant de retourner OK si tout est correct. Sinon, il retournera un code d'erreur selon la situation.

Le RPC de logout se contentera de vider la session.

Grille d'évaluation

Domaine	Critère	Indicateur	++ + +/- -	Commentaires
Forme	Respect des consignes	Le travail est remis dans les temps (avant le cours suivant)		
		Les contraintes sont respectées		Les consignes de l'énoncé sont respectées
Respect des standards	Code lisible	Le code est correctement indenté		
		Les noms des variables sont explicites et respectent les règles de nommage		
	Standard html et CSS récents respectés ?	La page n'utilise pas les balises ou de propriétés CSS dépréciées ou incorrectes		
Correction	Erreurs	Le programme tourne sans erreurs		Pas de messages d'erreurs ou d'erreurs visibles lors de l'utilisation
	Résultat	Les fonctionnalités implémentées effectuent bien les tâches demandées		Le résultat est bien celui attendu (pas de résultat incorrect)
	Complet	Toutes les fonctionnalités demandées sont présentes		
	Portabilité	Le programme peut tourner sans problème sur tous les navigateurs récents		
		Les chemins vers les ressources sont exprimés de façon relative et correspondent à la norme web		
Qualité de la programmation	Organisation	Les codes Javascript et les feuilles de styles sont dans des fichiers séparés		
		Une arborescence logique est utilisée		
	Code propre	Pas de répétition inutile de blocs de codes, de conditions, ...		
		Pas de variables ou paramètres inutiles		
		Les extensions des éléments statiques correspondent à leur contenu. Les extensions standards sont utilisées		
	Elégance	Les algorithmes utilisés sont efficaces et bien choisis		
Design	Affichages corrects	Les affichages sont corrects, disposés de façon logique		
		Absence d'artefacts de rendu (espacements, bordures,... dont la présence n'est pas voulue et nuit à l'affichage)		
Documentation	RPC Documentés	Les RPC AJAX sont documentés (noms, entrées, sorties, utilités, effets de bord, ...)		Prévoir un fichier TXT qui décrit les RPC : nom du fichier PHP, utilité (nom long), paramètres en entrée, valeurs en sortie, détails (ce que le RPC fait, pas comment il le fait)