
COMPTE RENDU TP06 – DEVWEB

Christopher Marie-Angélique – Groupe 2B

Table des matières

Introduction	2
1 – Création de la page AjoutType.php	2
2 – Maintien du choix effectué dans les formulaires	Erreur ! Signet non défini.
Conclusion	3

Introduction

Dans ce 6^{ème} TP de DevWeb, nous utiliserons une fois de plus phpmyadmin, cette fois ci pour insérer de nouveau éléments dans notre base de donnée. Pour insérer ces nouveaux éléments dans notre base de données, nous utiliserons des REGEX pour vérifier la conformité des données.

1 – Création de la page AjoutType.php

Veuillez entrer les informations du nouveau type de biens :

ID type :

Nom du type :

Valider

Cette nouvelle page va permettre aux Utilisateurs d'insérer un nouveau type de biens dans notre base de données. Elle contient deux « input » de type « text » ainsi qu'un bouton contenu dans une balise « form » comme suit :

```
<h4>Veuillez entrer les informations du nouveau type de biens :</h4>
<br>
<form method="POST">
  <label for="id">ID type : </label>
  <input type="text" name="id" id="id" required>
  <br><br>
  <label for="name">Nom du type : </label>
  <input type="text" name="name" id="name" required>
  <br>
  <br>
  <input type="submit" value="Valider">
</form>
```

Par la suite nous créons un script qui va donc insérer dans la base de données le nouveau type de biens et nous redirigera vers la page ConsultType.php. Nous vérifions dans notre script la validité des données avec des REGEX :

```

<?php
    $isValid = true;

    if (isset($_POST['id']) && isset($_POST['name'])) {
        if (!preg_match('#^F([1-8]|[A-Z])$#', $_POST['id'])) {
            echo '<p>L\'ID type est invalide.</p>';
            $isValid = false;
        }

        if (!preg_match('#^[a-zA-Z ]{6,25}$#', $_POST['name'])) {
            echo '<p>Le nom de type est invalide.</p>';
            $isValid = false;
        }

        if ($isValid == true) {
            $pdostat = $conn->prepare('INSERT INTO Types VALUES (:idType, :nomType)');
            try {
                $pdostat->execute(['idType' => $_POST['id'], 'nomType' => $_POST['name']]);
            } catch (PDOException $e) {
                echo '<br><p>Erreur: '.$e->getMessage().'</p>';
                die();
            }
            $pdostat->closeCursor();
            header('location: ConsultType.php');
        }
    }
?>

```

Nous vérifions dans un premier temps si l'utilisateur à bien saisi l'ID et le nom du nouveau type à insérer. Si c'est le cas, nous vérifions avec des REGEX plusieurs conditions :

- idType contient seulement 2 caractères, le 1er étant un 'F' et le 2ème un chiffre entre 1 et 8 ou un caractère de l'alphabet en majuscule
- nomType contient seulement 6 à 25 lettres ou espace

Si les conditions sont respectées nous insérons ce type de biens dans la base de données grâce à une requête préparée et nous redirigeons l'utilisateur sur la page consultType.php.

Conclusion

En conclusion, nous avons vu dans ce TP comment faire appel à la base de données pour insérer de nouveaux éléments dans celle-ci grâce à une requête préparée. Par la même occasion, nous avons utilisé des REGEX pour vérifier la conformité des données.