

PDO : Sécurisation et validation des formulaires en PHP

voir cours en ligne : <https://www.pierre-giraud.com/php-mysql-apprendre-coder-cours/securiser-valider-formulaire/>

Ne jamais faire confiance aux données utilisateurs

Lorsqu'on crée des formulaires, c'est généralement pour demander aux utilisateurs de nous envoyer des données. Si on ne met pas en place des systèmes de filtre sur le type de données qui peuvent être envoyées pour chaque champ et de vérification ensuite de la qualité des données envoyées, les données récoltées vont alors pouvoir être aberrantes ou même potentiellement dangereuses.

Deux niveaux de vérifications (dans le navigateur / côté serveur) doivent être implémentés lors de la création de formulaires. En effet, n'utiliser qu'une validation dans le navigateur laisse de sérieuses failles de sécurité dans notre formulaire puisque les utilisateurs malveillants peuvent désactiver ces vérifications.

La validation des données du formulaire dans le navigateur

Les processus de validation des données que nous allons pouvoir mettre en place dans le navigateur vont s'effectuer avant ou au moment de la tentative d'envoi du formulaire. L'objectif va être ici de bloquer l'envoi du formulaire si certains champs ne sont pas correctement remplis et de demander aux utilisateurs de remplir correctement les champs invalides. Le HTML5 propose aujourd'hui des options de validation relativement puissantes et couvrant la majorité des besoins.

Attribut	Définition
size	Permet de spécifier le nombre de caractères dans un champ
minlength	Permet de spécifier le nombre minimum de caractères dans un champ
maxlength	Permet de spécifier le nombre maximum de caractères dans un champ
min	Permet de spécifier une valeur minimale pour un champ de type number ou date
max	Permet de spécifier une valeur maximale pour un champ de type number ou date
step	Permet de définir un multiple de validité pour un champ acceptant des données de type nombre ou date. En indiquant <code>step="4"</code> , les nombres valides seront -8, -4, 0, 4, 8, etc.
autocomplete	Permet d'activer l'autocomplétion pour un champ : si un utilisateur a déjà rempli un formulaire, des valeurs lui seront proposées automatiquement lorsqu'il va commencer à remplir le champ
required	Permet de forcer le remplissage d'un champ. Le formulaire ne pourra pas être envoyé si le champ est vide
pattern	Permet de préciser une expression régulière. La valeur du champ devra respecter la contrainte de la regex pour être valide

La validation des données du formulaire sur serveur

La validation dans le navigateur va nous éviter une immense majorité de données invalides et donc d'avoir des données à priori exploitables.

Cependant, elle n'est pas suffisante contre des utilisateurs malveillants puisque n'importe qui peut neutraliser les attributs HTML ou le JavaScript en les désactivant dans le navigateur avant d'envoyer le formulaire.

Une validation côté serveur, en PHP, va donc également s'imposer pour filtrer les données potentiellement dangereuses.

La vérification de base est de savoir si la variable de l'on souhaite récupérer existe (`isset`) et qu'elle n'est pas vide (`!empty`), exemple :

```
if ( isset($_GET['nom']) && !empty($_GET['nom'])){  
    $nom = $_GET['nom'] ;  
}
```

Ensuite PHP nous offre différentes options pour sécuriser nos formulaires en testant la validité des données envoyées : on va pouvoir utiliser des fonctions, des filtres, des expressions régulières, etc que vous découvrirez en fonction de vos besoins.

Exemple : formulaire pour ajouter un salarié

```
<form action="insertSalaries.php" method="get" >  
<fieldset>  
<legend> <i>Salariés</i></legend>  
Nom : <input type="text" name="nom" required placeholder="Nom" /> <br />  
Prénom : <input type="text" name="prenom" required placeholder="Prénom" /> <br />  
dates : <br />  
Date de naissance : <input type="date" name="naissance" /> <br />  
Date d'embauche : <input type="date" name="embauche" /> <br />  
Salaire : <input type="number" name="salaire" /> <br />  
Services :  
<select name="service">  
    <option value="commercial">Commercial</option>  
    <option value="comptable">Comptable</option>  
    <option value="informatique">informatique</option>  
</select>  
<fieldset>  
<input type="submit" value="Ajouter" />  
<input type="reset" value="Effacer" />  
</form>
```

le fichier : insertSalaries.php

```
<?php
require_once("connection.php");
//on récupère et on vérifie les données reçues par le formulaire
if ( isset($_GET['nom']) && !empty($_GET['nom'])){
    $nom = $_GET['nom'] ;
}
// à faire sur chaque donnée reçue
$prenom = $_GET['prenom'];
$naissance = $_GET['naissance'];
$embauche = $_GET['embauche'];
$salaire = $_GET['salaire'];
$service = $_GET['service'];

// on rédige la requête SQL
$sql = " INSERT into salaries (nom, prenom, date_naissance, date_embauche, salaire, service )
        values (:nom, :prenom, :naissance, :embauche, :salaire, :service)";

try {
    //on prépare la requête avec les données reçues
    $statement = $pdo->prepare($sql);
    $statement->bindParam(':nom', $nom, PDO::PARAM_STR);
    $statement->bindParam(':prenom', $prenom, PDO::PARAM_STR);
    $statement->bindParam(':naissance', $naissance, PDO::PARAM_STR);
    $statement->bindParam(':embauche', $embauche, PDO::PARAM_STR);
    $statement->bindParam(':salaire', $salaire, PDO::PARAM_INT);
    $statement->bindParam(':service', $service, PDO::PARAM_STR);
    $statement->execute();
    //On renvoie vers la liste des salaries
    header("Location:listeSalaries.php");
}
catch(PDOException $e){
    echo 'Erreur : '.$e->getMessage();
}
?>
```

Travail à faire :

- 1 – Ajouter un lien sur votre page listeSalaries.php pour ajouter un salarié dans votre table.
- 2 - Ajouter des liens pour supprimer et modifier un enregistrement :

id	nom	prenom	date-naissance	date-embauche	salaire	service	delete	update
1	Cuset	Jean	1966-06-02	1995-05-15	2400	informatique	delete	update
2	Dian	Charles	1972-08-30	2007-04-01	1800	commercial	delete	update
3	Paco	Alain	1958-04-25	2002-06-06	2100	commercial	delete	update
4	Perez	Amanda	1972-01-11	1999-03-11	1700	comptable	delete	update
5	Jeanu	Thierry	1970-07-11	1998-01-01	1600	comptable	delete	update

