

**PROJET CERTIFIANT S1**

# **TUTORIELS**

**Sujet : Faire un CRUD avec PHP (PDO)**

ÉCRIT PAR

**Ugo RASTELL**

ANNÉE-2

**Développeur web**

# Sommaire

**I. COMMENT SE CONNECTER À BASE DE DONNÉES AVEC PHP ? (PDO) - PAGE 3**

**II. COMMENT AJOUTER DES DONNÉES DANS UNE BASE DE DONNÉES AVEC PHP ? (PDO) - PAGE 4**

**III. COMMENT SELECTIONNER LES DONNÉES DE LA BASE DE DONNÉES AVEC PHP ? (PDO) - PAGE 7**

**IV. COMMENT SUPPRIMER DES DONNÉES DANS UNE BASE DE DONNÉES AVEC PHP ? (PDO) - PAGE 9**

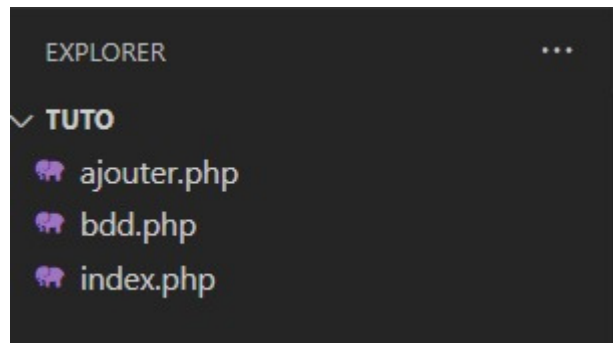
**V. COMMENT MODIFIER DES DONNÉES DANS UNE BASE DE DONNÉES AVEC PHP ? (PDO) - PAGE 12**

# **I. Comment se connecter à base de données avec PHP ? (PDO)**

Lien de la vidéo YouTube : **[https://www.youtube.com/watch?v=zmC4GGhjCSM&ab\\_channel=UgoRASTELL](https://www.youtube.com/watch?v=zmC4GGhjCSM&ab_channel=UgoRASTELL)**

## II. Comment ajouter des données dans une base de données avec PHP ? (PDO)

Ajouter dans votre dossier projet le fichier ajouter.php:



Dans le fichier ajouter.php, définir les informations de connexion à la base de données en déclarant les variables suivantes :

```
$host = 'localhost';  
$dbname = 'tuto';  
$username = 'root';  
$password = '';
```

Récupérer les valeurs du formulaire en utilisant les variables `$_POST` :

```
$prenom = $_POST['prenom'];  
$nom = $_POST['nom'];
```

Essayez de vous connecter à la base de données en utilisant la bibliothèque PDO :

```
try {  
    $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);  
    // définit le mode d'erreur pour PDO exception  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
}
```

Préparez une requête SQL pour insérer les données dans la table `etudiant` :

```
$sql = "INSERT INTO etudiant (prenom, nom)
VALUES ('$prenom', '$nom')";
```

Exécutez la requête en utilisant la méthode `exec()` de la connexion PDO :

```
$conn->exec($sql);
echo "Les données ont été ajoutées avec succès";
```

Gérez les exceptions potentielles en utilisant le bloc `catch` :

```
} catch(PDOException $e) {
    echo "L'ajout de données a échoué: " . $e->getMessage();
}
```

Remplacer le `<?php require_once "./bdd.php" ?>` dans le fichier index.php par :

```
<?php require_once "./ajouter.php" ?>
```

Si vous avez ces messages d'erreurs :

(!) Notice: Undefined index: prenom in C:\wamp64\www\tuto\ajouter.php on line 7				
Call Stack				
#	Time	Memory	Function	Location
1	0.0005	363192	{main}()	...\index.php:0
2	0.0005	363544	require_once( 'C:\wamp64\www\tuto\ajouter.php' )	...\index.php:1

(!) Notice: Undefined index: nom in C:\wamp64\www\tuto\ajouter.php on line 8				
Call Stack				
#	Time	Memory	Function	Location
1	0.0005	363192	{main}()	...\index.php:0
2	0.0005	363544	require_once( 'C:\wamp64\www\tuto\ajouter.php' )	...\index.php:1

Ignorer les et remplir les input et appuyer sur le bouton soumettre

Fin ! :

Les données ont été ajoutées avec succès

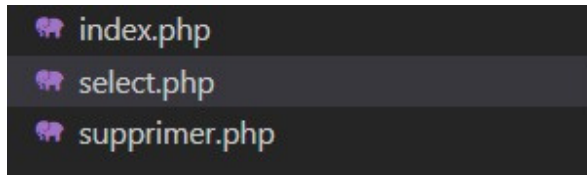
Prénom:

Nom de famille:

Voilà, vous savez maintenant comment ajouter des données en utilisant PDO en PHP. Je vous conseille de bien comprendre chacune des étapes et de tester le code pour mieux comprendre son fonctionnement.

### III. Comment sélectionner les données de la base de données avec PHP ? (PDO)

Ajouter dans votre dossier projet le fichier select.php:



Dans le fichier ajouter.php, définition des informations de connexion à la base de données :

```
$host = 'localhost';  
$dbname = 'tuto';  
$username = 'root';  
$password = '';
```

Initialisation de la connexion à la base de données via PDO :

```
try {  
    $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
```

Définition du mode d'erreur pour PDO exception :

```
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

Préparation de la requête SQL :

```
$sql = "SELECT * FROM etudiant";  
$stmt = $conn->prepare($sql);
```

Exécution de la requête préparée :

```
$stmt->execute();
```

Récupération des données sélectionnées :

```
$result = $stmt->fetchAll();
```

Affichage des données sélectionnées :

```
foreach($result as $row) {  
    echo "Prénom: " . $row['prenom'] . "<br>";  
    echo "Nom: " . $row['nom'] . "<br>";  
}
```

Gestion des erreurs en cas d'échec de la requête :

```
} catch(PDOException $e) {  
    echo "La requête a échoué: " . $e->getMessage();  
}
```

Remplacer le `<?php require_once "/edit.php" ?>` dans le fichier index.php par :

```
<?php require_once "/select.php" ?>
```

Fin ! :

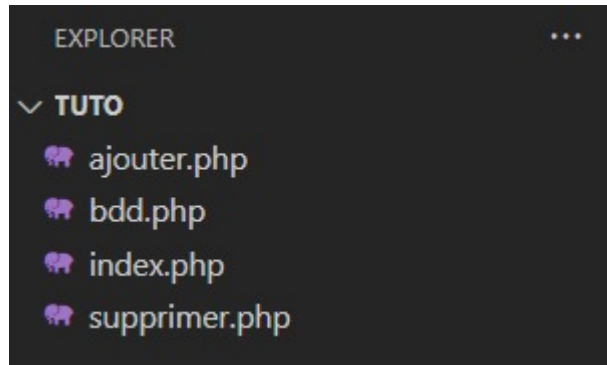
```
Prénom:  
Nom:  
Prénom:  
Nom:  
Prénom:  
Nom:  
Prénom:  
Nom:  
Prénom: ugo  
Nom: rastell
```

Voilà, vous savez maintenant comment sélectionner des données en utilisant PDO en PHP. Je vous conseille de bien comprendre chacune des étapes et de tester le code pour mieux comprendre son fonctionnement.



## IV. Comment supprimer des données dans une base de données avec PHP ? (PDO)

Ajouter dans votre dossier projet le fichier supprimer.php:



Dans le fichier supprimer.php, Il faut commencé par définir les informations de connexion à la base de données MySQL:

```
$host = 'localhost';  
$dbname = 'tuto';  
$username = 'root';  
$password = '';
```

Ensuite, une connexion à la base de données est établie à l'aide de la bibliothèque PDO:

```
$conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);
```

Le mode d'erreur pour PDO est défini pour que des exceptions soient générées en cas d'erreur:

```
$conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

Si un ID est fourni dans l'URL (**`$_GET['id']`**), une requête SQL est exécutée pour supprimer l'enregistrement correspondant:

```
if(isset($_GET['id'])) {  
    $id = $_GET['id'];  
    $sql = "DELETE FROM etudiant WHERE id = :id";  
    $stmt = $conn->prepare($sql);  
    $stmt->bindParam(':id', $id, PDO::PARAM_INT);  
    $stmt->execute();  
}
```

Enfin, une requête SQL est exécutée pour sélectionner tous les enregistrements dans la table **etudiant** et les afficher:

```
$sql = "SELECT * FROM etudiant";  
$stmt = $conn->prepare($sql);  
$stmt->execute();  
$result = $stmt->fetchAll();  
foreach($result as $row) {  
    echo "Prénom: " . $row['prenom'] . "<br>";  
    echo "Nom: " . $row['nom'] . "<br>";  
    echo "<a href='?id=" . $row['id'] . "'>Supprimer</a><br>";  
}
```

En cas d'erreur lors de la requête, un message d'erreur est affiché:

```
} catch(PDOException $e) {  
    echo "La requête a échoué: " . $e->getMessage();  
}
```

Remplacer le **`<?php require_once './ajouter.php' ?>`** dans le fichier index.php par :

```
<?php require_once './supprimer.php' ?>
```

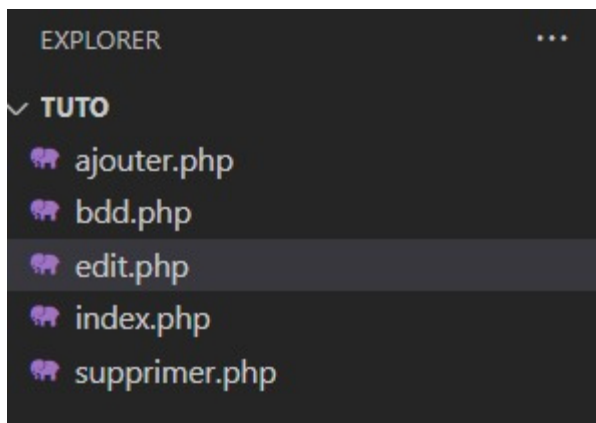
Fin ! :

Prénom:  
Nom:  
[Supprimer](#)  
Prénom:  
Nom:  
[Supprimer](#)  
Prénom:  
Nom:  
[Supprimer](#)  
Prénom:  
Nom:  
[Supprimer](#)  
Prénom:  
Nom:  
[Supprimer](#)  
Prénom:  
Nom:  
[Supprimer](#)  
Prénom: ugo  
Nom: rastell  
[Supprimer](#)  
Prénom:   
  
Nom de famille:

Voilà, vous savez maintenant comment supprimer par id des données en utilisant PDO en PHP. Je vous conseille de bien comprendre chacune des étapes et de tester le code pour mieux comprendre son fonctionnement.

## V. Comment modifier des données dans une base de données avec PHP ? (PDO)

Ajouter dans votre dossier projet le fichier edit.php:



Voici un tutoriel étape par étape pour éditer des données dans une base de données MySQL en utilisant PDO et PHP :

Dans le fichier edit.php, définissez les informations de connexion à votre base de données MySQL, telles que le nom d'hôte, le nom de la base de données, le nom d'utilisateur et le mot de passe.

```
$host = 'localhost';  
$dbname = 'tuto';  
$username = 'root';  
$password = '';
```

Utilisez la classe PDO pour vous connecter à votre base de données et définir le mode d'erreur sur **PDOException**.

```
try {  
    $conn = new PDO("mysql:host=$host;dbname=$dbname", $username, $password);  
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);  
} catch(PDOException $e) {  
    echo "La connexion à la base de données a échoué: " . $e->getMessage();  
}
```

Vérifiez si l'ID de l'enregistrement à éditer a été transmis en utilisant la méthode GET. Si c'est le cas, préparez une requête SQL pour récupérer les données de l'enregistrement à l'aide de l'ID.

```
phpCopy code
if(isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql = "SELECT * FROM etudiant WHERE id = :id";
    $stmt = $conn->prepare($sql);
    $stmt->bindParam(':id', $id, PDO::PARAM_INT);
    $stmt->execute();
    $result = $stmt->fetch();
}
```

Affichez le formulaire avec les valeurs pré-remplies de l'enregistrement à éditer.

```
echo '<form action="" method="post">';
    echo '<input type="text" name="prenom" value="'.$result['prenom'].'">';
echo '<input type="text" name="nom" value="'.$result['nom'].'">';
echo '<input type="hidden" name="id" value="'.$result['id'].'">';
echo '<input type="submit" value="Enregistrer">';
echo '</form>';
}
```

Vérification de l'existence des données POST :

```
phpCopy code
if(isset($_POST['prenom']) && isset($_POST['nom'])) {
```

Cette vérification s'assure que les données nécessaires pour la mise à jour sont présentes dans la requête POST.

Récupération des données :

```
shellCopy code
$prenom = $_POST['prenom'];
$nom = $_POST['nom'];
$id = $_POST['id'];
```

Nous récupérons les valeurs du formulaire pour le prénom, le nom et l'ID.

## Préparation de la requête SQL :

```
bashCopy code
$sql = "UPDATE etudiant SET prenom = :prenom, nom = :nom WHERE id = :id";
$stmt = $conn->prepare($sql);
```

Nous préparons la requête SQL pour la mise à jour des données.

## Liage des valeurs :

```
phpCopy code
$stmt->bindParam(':prenom', $prenom, PDO::PARAM_STR);
$stmt->bindParam(':nom', $nom, PDO::PARAM_STR);
$stmt->bindParam(':id', $id, PDO::PARAM_INT);
```

Nous liions les valeurs récupérées à la requête préparée.

## Exécution de la requête :

```
scssCopy code
$stmt->execute();
```

Nous exécutons la requête pour mettre à jour les données.

## Affichage du message de succès :

```
bashCopy code
echo "Données mises à jour avec succès";
```

Nous affichons un message de succès pour indiquer que la mise à jour des données a réussi.

## Traitement des erreurs :

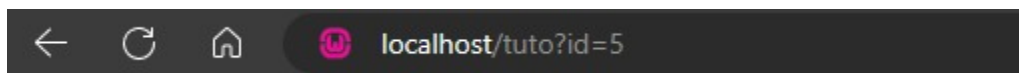
```
phpCopy code
} catch(PDOException $e) {
    echo "La requête a échoué: " . $e->getMessage();
}
```

Enfin, nous gérons les erreurs potentielles en les attrapant à l'aide de PDOException et en les affichant.

Remplacer le `<?php require_once "./supprimer.php" ?>` dans le fichier index.php par :

```
<?php require_once "./edit.php" ?>
```

Ensuite, dans l'URL, ajoutez '?id=' et insérez l'identifiant du correspondant pour modifier ses informations:



Fin ! :

ugo2	rastell	Enregistrer
------	---------	-------------

Données mises à jour avec succès

Prénom:

Nom de famille:

Voilà, vous savez maintenant comment mettre à jour des données en utilisant PHP et PDO !

**Merci d'avoir pris le temps de suivre mes  
tutoriels en entier.**