

# Welcome Training

## Extraits de code commentés

### 1. Création d'un utilisateur

#### 1.1 FetchAllClasses : classe récupérant la liste des classes existantes en base de données

```
class FetchAllClasses {
    private $pdo;

    public function __construct($pdo) {
        $this->pdo = $pdo;
    }

    public function fetchAllClasses() {
        $sql = "SELECT * FROM `class`";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();

        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }

    public function fetchAllClassesID() {
        $sql = "SELECT id, name FROM class";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

#### 1.2 UserForm : classe récupérant les données saisies dans le formulaire de création d'utilisateur

Si rien n'est sélectionné dans le champ « Classe » alors la colonne « class » de la table « user » prendra la valeur « null » (pour un utilisateur enseignant par exemple).

```
class UserForm {

    public $first_name;
    public $surname;
    public $email;
    public $password;
    public $role;
    public $class_id;

    public function __construct() {
        $this->fetchForm();
    }

    private function fetchForm() {
        $this->first_name = $_POST['first_name'];
        $this->surname = $_POST['surname'];
        $this->email = $_POST['email'];
        $this->password = $_POST['password'];
        $this->role = $_POST['role'];
        $this->class_id = !empty($_POST['class_id']) ? $_POST['class_id'] : null;
    }
}
```

## 1.3 CreateUser : classe qui crée l'utilisateur en base de données

Elle utilise les objets :

- \$fetchClass issu de la classe FetchAllClasses, qui permettra d'appeler la méthode retournant la liste des classes pour l'afficher en input select dans le formulaire de création d'utilisateur
- \$form issu de la classe UserForm, qui est un tableau clé-valeur permettant d'insérer dans la base de données les données utilisateur saisies dans le formulaire

```
class CreateUser {
    private $pdo;
    private $fetchClass;

    public function __construct($pdo) {
        $this->pdo = $pdo;
        $this->fetchClass = new FetchAllClasses($pdo);
    }

    private function pwdHash($pwd) {
        $options = [
            'cost' => 12,
        ];
        return password_hash($pwd, PASSWORD_BCRYPT, $options);
    }

    public function fetchClasses() {
        return $this->fetchClass->fetchAllClasses();
    }

    public function createUser() {
        $form = new UserForm;

        $hashedPassword = $this->pwdHash($form->password);

        $sql = "INSERT INTO user (first_name, surname, email, password, class_id, role)
            VALUES (:first_name, :surname, :email, :password, :class, :role)";

        $stmt = $this->pdo->prepare($sql);
        $stmt->bindParam(':first_name', $form->first_name, PDO::PARAM_STR);
        $stmt->bindParam(':surname', $form->surname, PDO::PARAM_STR);
        $stmt->bindParam(':email', $form->email, PDO::PARAM_STR);
        $stmt->bindParam(':password', $hashedPassword, PDO::PARAM_STR);
        $stmt->bindParam(':class', $form->class_id, PDO::PARAM_INT);
        $stmt->bindParam(':role', $form->role, PDO::PARAM_STR);

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

## 1.4 Traitement du formulaire de création d'utilisateur

Au chargement de la page du formulaire de création d'utilisateur :

- La classe CreateUser est instanciée avec la connexion PDO
- Sa méthode fetchClasses() est appelée pour récupérer la liste des classes disponibles depuis la base de données et la stocker dans l'array \$classes, utilisé dans le champ <select> du formulaire

À la soumission du formulaire (POST détecté) :

- La méthode createUser() est déclenchée pour récupérer les données saisies, les sécuriser (hash du mot de passe), et insérer un nouvel utilisateur en base de données.

```
<?php
include_once __DIR__ . "../classes/form/create-user-form-handler.php";

$user_form = new CreateUser($pdo);

$classes = $user_form->fetchClasses();

if (count($_POST) > 0) {
    $user_form->createUser();
}

?>
```

## 1.5 Formulaire de création d'utilisateur

L'instanciation préalable de \$classes permet d'afficher en input select la liste des classes à laquelle un utilisateur peut appartenir

```
1  <?php
2  include_once "../common/header.php";
3  include_once "../database/db-connection.php";
4  include_once "../classes/form/create-user-form-handler.php";
5  include_once "../database/create/db-create-user.php";
6  ?>
7
8  <main class="bg-light d-flex align-items-center vh-100">
9      <div class="container">
10
11         <div class="row justify-content-center">
12             <div class="col-md-8 col-lg-6">
13                 <div class="card bg-white border-0 shadow p-4 rounded">
14
15                     <h3 class="text-center text-secondary mb-4">Créer un utilisateur</h3>
16
17                     <form action="create-user.php" method="POST">
18
19                         <!-- First name -->
20                         <div class="mb-3">
21                             <label for="prenom" class="form-label">Prénom :</label>
22                             <input type="text" class="form-control" id="prenom" name="first_name" placeholder="Entrez le prénom" required>
23                         </div>
24
25                         <!-- Surname -->
26                         <div class="mb-3">
27                             <label for="nom" class="form-label">Nom :</label>
28                             <input type="text" class="form-control" id="nom" name="surname" placeholder="Entrez le nom" required>
29                         </div>
30
31                         <!-- Email -->
32                         <div class="mb-3">
33                             <label for="email" class="form-label">Email :</label>
34                             <input type="email" class="form-control" id="email" name="email" placeholder="exemple@campus-espl.org" required>
35                         </div>
36
37                         <!-- Password -->
38                         <div class="mb-3">
39                             <label for="password" class="form-label">Mot de passe provisoire :</label>
40                             <input type="password" class="form-control" id="password" name="password" placeholder="Entrez un mot de passe provisoire" required>
41                         </div>
42
43                         <!-- Role -->
44                         <div class="mb-3">
45                             <label for="role" class="form-label">Rôle :</label>
46                             <select class="form-control custom-select form-select" id="role" name="role" required>
47                                 <option value="" selected disabled></option>
48                                 <option value="etudiant">Étudiant</option>
49                                 <option value="enseignant">Enseignant</option>
50                                 <option value="administrateur">Administrateur</option>
51                             </select>
52                         </div>
53
54                         <!-- Class -->
55                         <div class="mb-4">
56                             <label for="classe" class="form-label">Classe :</label>
57                             <select class="form-control custom-select form-select mb-4" id="class_id" name="class_id">
58                                 <option value=""></option>
59                                 <?php foreach($classes as $class): ?>
60                                     <option value=<?php echo($class["id"]); ?>>
61                                         <?php echo($class["name"]); ?>
62                                 </option>
63                             <?php endforeach; ?>
64                             </select>
65                         </div>
66
67                         <!-- Submit Button -->
68                         <div class="d-flex justify-content-center">
69                             <button type="submit" class="btn rounded-pill w-50">Créer l'utilisateur</button>
70                         </div>
71                     </form>
72
73                 </div>
74             </div>
75         </div>
76     </div>
77 </main>
78
```

## 2. Création d'un cours

### 2.1 FetchAllSubjects : classe récupérant la liste des matières existantes en base de données

```
class FetchAllSubjects {
    private $pdo;

    public function __construct($pdo) {
        $this->pdo = $pdo;
    }

    public function fetchAllSubjects() {
        $sql = "SELECT * FROM `subject`";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();

        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }

    public function fetchAllSubjectsID() {
        $sql = "SELECT id FROM subject";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

### 2.2 FetchAllTeachers : classe récupérant la liste des utilisateurs ayant le rôle « enseignant » en base de données

```
class FetchAllTeachers {
    private $pdo;

    public function __construct($pdo) {
        $this->pdo = $pdo;
    }

    public function fetchAllTeachers() {
        $sql = "SELECT id, first_name, surname, email FROM user WHERE role = 'enseignant'";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();

        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

## 2.3 FetchAllClasses : classe récupérant la liste des classes existantes en base de données

```
class FetchAllClasses {
    private $pdo;

    public function __construct($pdo) {
        $this->pdo = $pdo;
    }

    public function fetchAllClasses() {
        $sql = "SELECT * FROM `class`";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();

        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }

    public function fetchAllClassesID() {
        $sql = "SELECT id, name FROM class";
        $stmt = $this->pdo->prepare($sql);
        $stmt->execute();
        return $stmt->fetchAll(PDO::FETCH_ASSOC);
    }
}
```

## 2.4 ScheduleForm : classe récupérant les données saisies dans le formulaire de création de cours

```
class ScheduleForm {

    public $subject_id;
    public $class_id;
    public $teacher_id;
    public $start_datetime_str;
    public $end_datetime_str;

    private $date;
    private $time;
    private $length;

    public function __construct() {
        $this->fetchForm();
        $this->setLength();
        $this->dateHandler();
    }

    private function fetchForm() {
        $this->subject_id = $_POST['subject_id'];
        $this->class_id = $_POST['class_id'];
        $this->teacher_id = empty($_POST['enseignant']) ? NULL : $_POST['enseignant'];
        $this->date = $_POST['date'];
        $this->time = $_POST['time'];
    }

    private function setLength() {
        $this->length = isset($_POST['duree']) ? (int)$_POST['duree'] : 0;
    }

    private function dateHandler() {
        // Combines date and time to form start_datetime using DateTime
        $start_datetime = new DateTime("$this->date $this->time");

        // Clones start_datetime to calculate end_datetime and add duration in hours
        $end_datetime = clone $start_datetime;
        $end_datetime->modify("+$this->length hours");

        // Converts DateTime objects to string format for database insertion
        $this->start_datetime_str = $start_datetime->format('Y-m-d H:i:s');
        $this->end_datetime_str = $end_datetime->format('Y-m-d H:i:s');
    }
}
```

## 2.5 CreateSchedule : classe qui crée un cours en base de données

Elle utilise les objets :

- \$fetchSubjects issu de la classe FetchAllSubjects, qui permettra d'appeler la méthode retournant la liste des matières pour l'afficher en input select dans le formulaire de création de cours
- \$fetchTeachers issu de la classe FetchAllTeachers, qui permettra d'appeler la méthode retournant la liste des enseignants pour l'afficher en input select dans le formulaire de création de cours
- \$fetchClasses issu de la classe FetchAllClasses, qui permettra d'appeler la méthode retournant la liste des classes pour l'afficher en input select dans le formulaire de création de cours
- \$form issu de la classe ScheduleForm, qui est un tableau clé-valeur permettant d'insérer dans la base de données les données cours saisies dans le formulaire

```
class CreateSchedule {  
  
    private $pdo;  
    private $fetchSubjects;  
    private $fetchTeachers;  
    private $fetchClasses;  
  
    public function __construct($pdo) {  
        $this->pdo = $pdo;  
        $this->fetchSubjects = new FetchAllSubjects($pdo);  
        $this->fetchTeachers = new FetchAllTeachers($pdo);  
        $this->fetchClasses = new FetchAllClasses($pdo);  
    }  
  
    public function fetchSubjects() {  
        return $this->fetchSubjects->fetchAllSubjects();  
    }  
  
    public function fetchTeachers() {  
        return $this->fetchTeachers->fetchAllTeachers();  
    }  
  
    public function fetchClasses() {  
        return $this->fetchClasses->fetchAllClasses();  
    }  
  
    public function createSchedule() {  
        $form = new ScheduleForm();  
  
        $sql_schedule = "INSERT INTO schedule (subject_id, class_id, teacher_id, start_datetime, end_datetime)  
                        VALUES (:subject_id, :class_id, :teacher_id, :start_datetime, :end_datetime)";  
  
        $stmt_schedule = $this->pdo->prepare($sql_schedule);  
        $stmt_schedule->bindParam(':subject_id', $form->subject_id, PDO::PARAM_INT);  
        $stmt_schedule->bindParam(':class_id', $form->class_id, PDO::PARAM_INT);  
        $stmt_schedule->bindParam(':teacher_id', $form->teacher_id, PDO::PARAM_INT);  
        $stmt_schedule->bindParam(':start_datetime', $form->start_datetime_str, PDO::PARAM_STR);  
        $stmt_schedule->bindParam(':end_datetime', $form->end_datetime_str, PDO::PARAM_STR);  
  
        return $stmt_schedule->execute();  
    }  
}
```



## 2.6 Traitement du formulaire de création de cours

Au chargement de la page du formulaire de création d'utilisateur :

- La classe CreateSchedule est instanciée avec la connexion PDO
- Sa méthode fetchSubjects() est appelée pour récupérer la liste des matières disponibles depuis la base de données et la stocker dans l'array \$subjects, utilisé dans un champ <select> du formulaire
- Sa méthode fetchTeachers() est appelée pour récupérer la liste des enseignants disponibles depuis la base de données et la stocker dans l'array \$teachers, utilisé dans le champ <select> du formulaire
- Sa méthode fetchClasses() est appelée pour récupérer la liste des classes disponibles depuis la base de données et la stocker dans l'array \$classes, utilisé dans le champ <select> du formulaire

À la soumission du formulaire (POST détecté) :

- La méthode createSchedule() est déclenchée pour récupérer les données saisies et insérer un nouveau cours en base de données.

```
<?php
include_once __DIR__ . "/../../classes/form/create-schedule-form-handler.php";

$schedule_form = new CreateSchedule($pdo);

$subjects = $schedule_form->fetchSubjects();
$teachers = $schedule_form->fetchTeachers();
$classes = $schedule_form->fetchClasses();

if(count($_POST) > 0) {
    $schedule_form->createSchedule();
}

?>
```

## 2.7 Formulaire de création de cours

L'instanciation préalable de \$subjects, \$teachers et \$classes permet d'afficher en input select les listes des matières, des enseignants et des classes relatives à un cours

```
1 <?php
2 include_once "../common/header.php";
3 include_once "../database/db-connection.php";
4 include_once "../classes/form/create-schedule-form-handler.php";
5 include_once "../database/create/db-create-schedule.php";
6 ?>
7
8 <main class="bg-light d-flex align-items-center vh-100">
9   <div class="container">
10     <div class="row justify-content-center">
11       <div class="col-md-8 col-lg-6">
12         <div class="card bg-white border-0 shadow p-4 rounded">
13
14           <h3 class="text-center text-secondary mb-4">Créer un cours</h3>
15
16           <form action="create-schedule.php" method="POST">
17
18             <!-- Matière -->
19             <div class="mb-3">
20               <label for="matiere" class="form-label">Matière :</label>
21               <select class="form-control custom-select form-select mb-4" id="subject_id" name="subject_id" required>
22                 <option value=""></option>
23                 <?php foreach($subjects as $s): ?>
24                   <option value=<?php echo($s["id"]); ?>>
25                     <?php echo($s["name"]); ?>
26                   </option>
27                 <?php endforeach; ?>
28               </select>
29             </div>
30
31             <!-- Date -->
32             <div class="mb-3">
33               <label for="date" class="form-label">Date :</label>
34               <input type="date" class="form-control" id="date" name="date" required>
35             </div>
36
37             <!-- Horaire -->
38             <div class="mb-3">
39               <label for="horaire" class="form-label">Horaire :</label>
40               <input type="time" class="form-control" id="time" name="time" required>
41             </div>
42
43             <!-- Durée -->
44             <div class="mb-3">
45               <label for="duree" class="form-label">Durée :</label>
46               <select class="form-control custom-select form-select" name="duree" id="duree">
47                 <option value=""></option>
48                 <option value="1">1h</option>
49                 <option value="2">2h</option>
50                 <option value="3">3h</option>
51                 <option value="4">4h</option>
52               </select>
53             </div>
54
55             <!-- Enseignant -->
56             <div class="mb-3">
57               <label for="enseignant" class="form-label">Enseignant :</label>
58               <select class="form-control custom-select form-select" id="enseignant" name="enseignant">
59                 <option value=""></option>
60                 <?php foreach($teachers as $t): ?>
61                   <option value = <?php echo($t["id"]); ?>>
62                     <?php echo($t["first_name"] . " " . $t["surname"]); ?>
63                   </option>
64                 <?php endforeach; ?>
65               </select>
66             </div>
67
68             <!-- Classe -->
69             <div class="mb-4">
70               <label for="classe" class="form-label">Classe :</label>
71               <select class="form-control custom-select form-select mb-4" id="class_id" name="class_id" required>
72                 <option value=""></option>
73                 <?php foreach($classes as $class): ?>
74                   <option value = <?php echo($class["id"]); ?>>
75                     <?php echo($class["name"]); ?>
76                   </option>
77                 <?php endforeach; ?>
78               </select>
79             </div>
80
81             <!-- Submit Button -->
82             <div class="d-flex justify-content-center">
83               <button type="submit" class="btn rounded-pill w-50">Créer le cours</button>
84             </div>
85           </form>
86
87         </div>
88       </div>
89     </div>
90   </div>
```