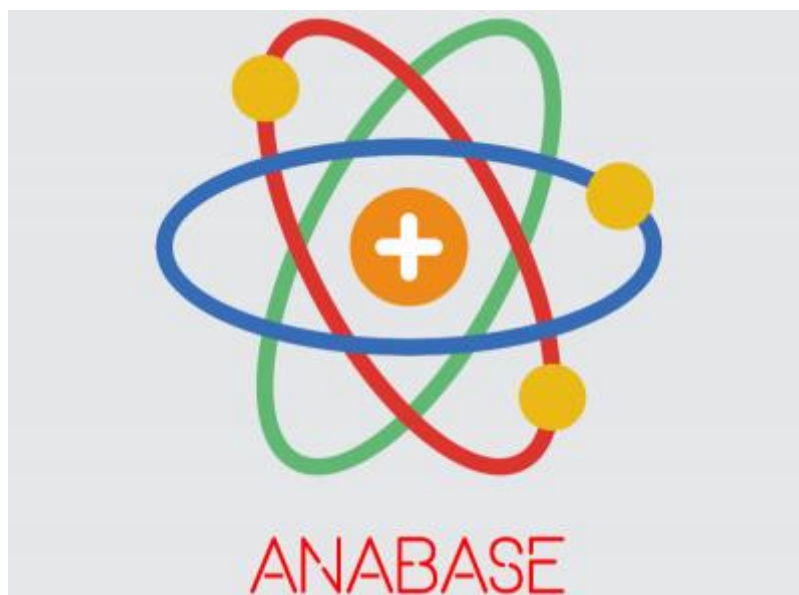


# AP3.1 ANABASE MVC-PHP OBJET

---



---

## PRÉSENTATION

Présentation de l'équipe et de la mission	
Durée	Du 10 septembre au 22 octobre 2021
N° mission	Mission 6
Qui sommes nous?	L'équipe est composée de deux élèves de 2ème année en BTS SIO option SLAM. Christian et Mélanie.

---

## OBJECTIFS

Afin de réaliser notre mission, nous avons deux tâches à accomplir :

- **Gestion des sessions** : Cette tâche doit permettre la **création**, la **modification**, la **suppression** d'une **session** à partir d'un formulaire. Il permet également d'obtenir la **liste des sessions sur un formulaire**. Il faudra réaliser les maquettes de ces formulaires, les faire valider par le chef de projet avant de commencer le codage.
- **Inscription à une session** : Cette tâche doit permettre d'**inscrire un congressiste à une session**, mais également de permettre l'**annulation de son inscription**. Pour chacun des cas la facture ne doit pas avoir été créée.

## Création d'une session

Nous commençons par créer une nouvelle **action** dans le **Contrôleur principal**.

```
$lesActions["session"] = "controleur.gosession.php";
```

Comme nous le voyons ci-dessus, nous avons appelé cette **action**, **session**.

### Affichage du tableau

Avant la création d'une **session**, nous avons commencé par l'affichage du tableau **Gestion des sessions** comprenant le **numéro de la session**, la **date**, l'**heure**, le **prix** et le **nom de la session**.

Nous nous aidons de l'architecture **MVC**. De ce fait, nous avons récupéré les données correspondant dans la base de données via le **Modèle**. Nous pouvons rappeler que ce dernier est le noyau de l'application qui gère les données, permet de récupérer les informations dans la base de données, de les organiser pour qu'elles puissent ensuite être traitées par le **Contrôleur**.

```
function getSession()
{
    try {

        $req = $this->conn->prepare("SELECT * FROM session");

        $req->execute();

        $resultat = $req->fetchAll(PDO::FETCH_OBJ);

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

    return $resultat;
}
```

Nous souhaitons faire afficher toutes les sessions dans un tableau. La requête qui se trouve ci-dessus est donc très simple (nous récupérons tous les éléments de la table session avec un **select \***).

Pour gérer les **Sessions**, nous créons un fichier **contrôleur.gosession**. Rappelons que ce dernier correspond à la principale partie de l'architecture, car c'est lui qui va faire le lien entre l'utilisateur, le **Modèle** et la **Vue**.

Nous codons en **PDO (PHP Data Objects)**, de ce fait, nous gérons le contrôleur **gosession** avec des fonctions nommées **todo** qui correspondent à l'action que nous voulons réaliser dans le **Contrôleur**.

```
public function todo_initialiser()

{

    $this->post["dateSession"] = "";

    $this->post["heureSession"] = "";

    $this->post["prixSession"] = "";

    $this->post["nomSession"] = "";

    $this->data["liste"] = $this->modele->getSession();

}
```

La première fonction **todo\_initialiser()** nous permet tout simplement d'afficher les données que nous voulons et que nous avons indiqué dans la fonction **getSession()** du **Modèle**.

Comme nous sommes passé au **modèle objet**, pour accéder à toutes les **Sessions** nous faisons tout d'abord **\$this->data["liste"]** qui nous permettra de faire le lien avec la **Vue** que nous verrons tout à l'heure, puis un **\$this->modele** avec la méthode : **getSession()** afin de faire le lien avec le **Modèle**.

À présent, nous pouvons voir ce que ça donne côté **Vue**.

```
<table>

    <tr>

        <th colspan="5">TABLEAU POUR CONSULTER LES SESSIONS</th>

    </tr>

    <tr>
```

```

        <td><B>Numéro de la session</B></td>

        <td><B>Date de la session</B></td>

        <td><B>Heure de la session</B></td>

        <td><B>Prix de la session</B></td>

        <td><B>Nom de la session</B></td>

    </tr>

    <?php foreach ($c->data["liste"] = $c->modele->getSession() as
$UneSession) { ?>

        <tr>

            <td><?php echo $UneSession->NUM_SESSION; ?> </td>

            <td><?php echo $UneSession->DATE_SESSION; ?></td>

            <td><?php echo $UneSession->HEURE_SESSION; ?></td>

            <td><?php echo $UneSession->PRIX_SESSION; ?>€</td>

            <td><?php echo $UneSession->NOM_SESSION; ?></td>

        </tr>

    <?php }

    ?>

</table>

```

Comme nous le voyons ci-dessus, pour récupérer toutes les sessions, nous faisons un **foreach**. Dedans notre boucle, le **\$c** représente le contrôleur. Nous voyons que la syntaxe suivante est quasiment identique à la syntaxe que nous avons vu tout à l'heure dans le **Contrôleur**.

Nous n'avons plus qu'à ajouter du **PHP** dans notre tableau **HTML** et grâce au **foreach** nous allons exécuter la fonction donnée sur chaque élément du tableau.

Une fois allié au **CSS**, le tableau se trouvant dans notre **Vue**, est comme ci-dessous.

TABLEAU POUR CONSULTER LES SESSIONS				
Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
1	2021-09-12	16:00:00	12€	L'histoire de toto
2	2021-09-13	16:30:00	13€	Visite de totoville
3	2021-09-14	17:00:00	14€	Toto notre bien aimé
55	2021-11-01	20:00:00	5€	La teinture foirée
53	2021-10-25	15:00:00	16€	Les artistes incompris
54	2021-11-12	10:00:00	30€	Les reines de Valadon

## Création d'une session

Pour la création d'une **session**, nous allons commencer, comme tout à l'heure, par la description du fichier du **Modèle**.

Nous souhaitons que l'utilisateur ajoute **la date de la session**, **l'heure de la session**, **le prix de la session** et **le nom de la session**. Il nous faut donc seulement une requête d'insertion.

```
public function insertSession($dateSession, $heureSession, $prixSession,
    $nomSession)

{

    try {

        $req = $this->conn->prepare("INSERT INTO session VALUES (NULL,
        :dateSession, :heureSession, :prixSession, :nomSession)");

        $req->execute(array(

            'dateSession' => $dateSession,

            'heureSession' => $heureSession,

            'prixSession' => $prixSession,

            'nomSession' => $nomSession,

        ));

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();
    }
}
```

```
}  
  
}
```

Comme nous le voyons ci-dessus, le **numéro de session** est en **auto-incrémentation**. En effet, nous n'avons pas souhaité le gérer et de ce fait, il se définira automatiquement.

Nous pouvons passer ensuite au **Contrôleur**.

```
public function todo_enregistrer()  
  
{  
  
    $this->modele->insertSession($this->post["datesession"],  
$this->post["heuresession"], $this->post["prixsession"], $this->  
>post["nomsession"]);  
  
}
```

Pour gérer l'insertion d'une session, nous créons comme tout à l'heure, une fonction **todo** que nous nommons **todo\_enregistrer**. Puis, nous appelons à l'aide de la syntaxe **\$this→modele** la requête dont nous avons parlé précédemment, **insertSession** et nous passons en paramètre avec l'aide de **\$this→post** la **date de la session**, l'**heure de session**, le **prix de la session** et le **nom de la session**.

À présent, nous pouvons voir ce que ça donne côté **Vue**.

```
<div class="container">  
  
    <input type="hidden" name="numsession" value="" required>  
  
    <input type="date" name="datesession" value="" required>  
  
    <input type="time" name="heuresession" value="" required>  
  
    <input type="number" name="prixsession" placeholder="Saisir le prix de la  
session" value="" required></br>
```

```
<input type="text" name="nomsession" placeholder="Saisir le nom de la session" value="" required></br>

</div>
```

Après avoir créé un **input type submit** afin d'ajouter la nouvelle session, nous avons caché l'**input de numsession** grâce au **type hidden** car, puisqu'il est en **auto-incrément**, l'utilisateur ne l'ajoutera pas.

Ensuite, nous avons créé un **input type date** afin de rentrer la date de la nouvelle session, un **input type time** afin de rentrer l'heure de la nouvelle session, un **input type number** afin de rentrer le prix de la nouvelle session et un **input type text** afin de rentrer le nom de la nouvelle session.

Nous pouvons voir ce que ça donne une fois allié au **CSS**.



The image shows a web application interface. In the background, there are two tables. The left table is titled 'TABLE' and has columns 'Numéro de la session' and 'Date de la session'. It contains rows with numbers 1, 2, 3, 56, 55, and 54. The right table is titled 'SESSIONS' and has columns 'Num de la session' and 'Nom de la session'. It contains rows with session names like 'L'histoire de toto', 'Visite de toutoville', 'Toto notre bien aimé', 'Text compte rendu', 'La teinture fleurie', and 'Les reines de Valadon'. Overlaid on top of these tables is a white modal form with rounded corners and a purple border. The modal is titled 'AJOUTER UNE SESSION'. It contains four input fields: a date input with a calendar icon, a time input with a clock icon, a number input with the placeholder 'Saisir le prix de la session', and a text input with the placeholder 'Saisir le nom de la session'. At the bottom of the modal are two purple buttons: 'Enregistrer' and 'Fermer'.

🎮 Jeu d'essais

Ajout d'une session :

### AJOUTER UNE SESSION

25/10/2021

19:37

30

Test compte rendu

Enregistrer

Fermer

Vérification sur le site :

TABLEAU POUR CONSULTER LES SESSIONS				
Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
1	2021-09-12	16:00:00	12€	L'histoire de toto
2	2021-09-13	16:30:00	13€	Visite de totoville
3	2021-09-14	17:00:00	14€	Toto notre bien aimé
56	2021-10-25	19:37:00	30€	Test compte rendu
55	2021-11-01	20:00:00	5€	La teinture foirée
54	2021-11-12	10:00:00	30€	Les reines de Valadon

Vérification sur phpmyadmin :

NUM_SESSION	DATE_SESSION	HEURE_SESSION	PRIX_SESSION	NOM_SESSION
1	2021-09-12	16:00:00	12	L'histoire de toto
2	2021-09-13	16:30:00	13	Visite de totoville
3	2021-09-14	17:00:00	14	Toto notre bien aimé
56	2021-10-25	19:37:00	30	Test compte rendu
55	2021-11-01	20:00:00	5	La teinture foirée
54	2021-11-12	10:00:00	30	Les reines de Valadon

Nous voyons que ça **FUNCTIONNE !!**



## Modification d'une session ✂

Nous allons gérer la modification dans le même **Contrôleur** et dans le même **Modèle**.

Si l'utilisateur veut modifier une session, il doit choisir via un **input type radio**, la session qu'il veut modifier.

```
<input value="<?php echo $UneSession->NUM_SESSION; ?>" name="session" type="radio" required>
```

Comme nous pouvons le voir ci-dessus, nous avons mis dans la **value** un **echo** du **foreach** avec pour paramètre le **numéro de la session** car c'est ce dernier qui sera transmis lorsque l'utilisateur fera son choix.

Puis, lorsqu'il aura choisi la session qu'il veut modifier, il devra valider son choix en cliquant sur le **submit** qui permettra d'afficher les données dans quatre **inputs**. Pour cela, nous avons besoin de deux requêtes : une pour modifier et une autre pour afficher.

**Nous allons commencer par l'affichage.**

```
function getSessionByNum($numsession)

{

    try {

        $req = $this->conn->prepare("SELECT * FROM session
WHERE NUM_SESSION = :NUM_SESSION");

        $req->bindValue(':NUM_SESSION', $numsession,
PDO::PARAM_INT);

        $req->execute();

        $resultat = $req->fetch(PDO::FETCH_ASSOC);

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

    return $resultat;
```

```
}
```

Cette requête est partiellement identique à celle concernant l'affichage du tableau **getSession**. En effet, nous avons rajouté un **Where** car nous souhaitons afficher toutes les sessions en fonction de la clé primaire **NUM\_SESSION**. Comme elle contient un paramètre, nous rajoutons la variable **\$numsession**, à côté du nom de la fonction que nous déclarons.

Nous devons maintenant gérer cette requête dans le **Contrôleur**.

```
public function todo_Afficher()  
  
    {  
  
        $numsession = $this->post['session'];  
  
        $this->data["afficher"] = $this->modele->getSessionByNum($numsession);  
  
    }
```

Pour gérer l'affichage, nous créons comme tout à l'heure, une fonction **todo** que nous nommons **todo\_Afficher**. Ensuite, nous récupérons le paramètre de la requête **getSessionByNum**, et à l'aide de **\$this->post['session']** et de **\$this->data["afficher"]**, nous faisons un lien avec la **Vue** (afin qu'on puisse l'utiliser tout à l'heure). Puis, à l'aide de la syntaxe **\$this->modele**, nous appelons la requête dont nous avons parlé à l'instant.

Nous pouvons passer à la **Vue**.

```
<table name="modif">  
  
    <tr>  
  
        <th colspan="6">VEUILLEZ SELECTIONNER LA SESSION QUE VOUS VOULEZ  
MODIFIER</th>  
  
    </tr>  
  
    <tr>  
  
        <td><B>Sélectionner</B></td>  
  
        <td><B>Numéro de la session</B></td>  
  
        <td><B>Date de la session</B></td>  
  
        <td><B>Heure de la session</B></td>
```

```

        <td><B>Prix de la session</B></td>

        <td><B>Nom de la session</B></td>

    </tr>

    <?php foreach ($c->data["liste"] = $c->modele->getSession() as
$UneSession) { ?>

        <tr>

            <td><input value="<?php echo $UneSession->NUM_SESSION; ?>"
name="session" type="radio" required></td>

            <td><?php echo $UneSession->NUM_SESSION; ?> </td>

            <td><?php echo $UneSession->DATE_SESSION; ?></td>

            <td><?php echo $UneSession->HEURE_SESSION; ?></td>

            <td><?php echo $UneSession->PRIX_SESSION; ?>€</td>

            <td><?php echo $UneSession->NOM_SESSION; ?></td>

        </tr>

    <?php }

    ?>

</table>

```

Comme nous le voyons ci-dessus, l'extrait de code est quasiment identique à celui de l'affichage du tableau que nous voyons au début car il nous permet de rajouter uniquement l'**input type radio**.

En effet, si l'utilisateur veut modifier une session, il doit cliquer sur un bouton que permettra l'ouverture du pop-up comprenant ce tableau ainsi que l'**input type radio** afin qu'il puisse procéder à son choix.

Nous pouvons voir la capture de ce pop-up ci-dessous.

**MODIFIER UNE SESSION**

**VEUILLEZ SELECTIONNER LA SESSION QUE VOUS VOULEZ MODIFIER**

Sélectionner	Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
<input type="radio"/>	1	2021-09-12	16:00:00	12€	L'histoire de toto
<input type="radio"/>	2	2021-09-13	16:30:00	13€	Visite de totoville
<input type="radio"/>	3	2021-09-14	17:00:00	14€	Toto notre bien aimé
<input type="radio"/>	56	2021-10-25	19:37:00	30€	Test compte rendu
<input type="radio"/>	55	2021-11-01	20:00:00	5€	La teinture foirée
<input type="radio"/>	54	2021-11-12	10:00:00	30€	Les reines de Valadon

```
<input type="submit" class="open-button" name="todo" value="Afficher" />
```

C'est à l'aide de ce **submit** que nous ouvrons un pop-up comprenant les informations de la session sélectionnée avec laquelle elle soit modifiée.

Nous pouvons voir ci-dessous le code de la **Vue** de ce pop-up.

```
<div class="container">

    <input type="hidden" name="numsession" value="<?php if (isset($c-
>data["afficher"])) {

                                echo $c-
>data["afficher"]['NUM_SESSION'];

                                } else { ?> <?php } ?>">

    <input type="date" name="datesession" value="<?php if (isset($c-
>data["afficher"])) {

                                echo $c-
>data["afficher"]['DATE_SESSION'];

                                } else { ?> <?php }
?>"></br>
```

```

        <input type="time" name="heuresession" value="<?php if (isset($c-
>data["afficher"])) {

                                                    echo $c-
>data["afficher"]['HEURE_SESSION'];

                                                    } else { ?> <?php }
?>"></br>

        <input type="number" name="prixsession" placeholder="Saisir le prix de la
session" value="<?php if (isset($c->data["afficher"])) {

echo $c->data["afficher"]['PRIX_SESSION'];

} else { ?> <?php } ?>"></br>

        <input type="text" name="nomsession" placeholder="Saisir le nom de la
session" value="<?php if (isset($c->data["afficher"])) {

echo $c->data["afficher"]['NOM_SESSION'];

} else { ?> <?php } ?>"></br>

</div>

```

Les **values** se trouvant dans chaque **input** contiennent chacune un **if** qui nous permet de faire : s'il y a quelque chose dans **\$c->data["afficher"]**, on affiche à l'aide d'un **echo** son contenu.

**Nous pouvons à présent passer à la modification.**

```

function updateSession($numSession, $dateSession, $heureSession, $prixSession,
$numSession)

{

    try {

```

```

        $req = $this->conn->prepare("UPDATE session SET
DATE_SESSION = ?,HEURE_SESSION = ?, PRIX_SESSION = ?, NOM_SESSION = ?

                                WHERE NUM_SESSION = ?");

        $req->bindValue(1, $dateSession);

        $req->bindValue(2, $heureSession);

        $req->bindValue(3, $prixSession);

        $req->bindValue(4, $nomSession);

        $req->bindValue(5, $numSession);

        $req->execute();

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

}

```

Ci-dessus, se trouve la requête **updateSession** qui possède cinq paramètres. Grâce à elle nous allons modifier la date et/ou l'heure et/ou le prix et/ou le nom de la session.

```

public function todo_Modifier()

{

    $numSession = $this->post['numsession'];

    $dateSession = $this->post['datesession'];

    $heureSession = $this->post['heuresession'];

    $prixSession = $this->post['prixsession'];

    $nomSession = $this->post['nomsession'];

    $this->modele->updateSession($numSession, $dateSession,
    $heureSession, $prixSession, $nomSession);
}

```

```
}
```

Pour gérer la modification, nous créons comme tout à l'heure, une fonction **todo** que nous nommons **todo\_Modifier**. Ensuite, nous récupérons les paramètres de la requête **updateSession** à l'aide de la syntaxe **\$this→post**. Puis, avec **\$this→modele**, nous appelons la requête dont nous avons parlé à l'instant.

```
<input type="submit" class="btn" name="todo" value="Modifier" />
```

Pour la **Vue**, il nous faut seulement rajouter l'**input** pour modifier.

Après y avoir ajouté du **CSS**, nous pouvons tester la **modification** et l'**affichage**.

The screenshot shows a web application interface for modifying a session. It features a main form titled 'MODIFIER UNE SESSION' with a table for selecting a session to modify. The table has columns for 'Sélectionner', 'Numéro de la session', 'Date de la session', 'Heure de la session', 'Prix de la session', and 'Nom de la session'. Below the table are 'Afficher' and 'Fermer' buttons. To the right, a sidebar contains input fields for '13/09/2021', '16:30', '13', and 'Visite de totoville', along with 'Modifier' and 'Fermer' buttons.

Sélectionner	Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
<input type="radio"/>	1	2021-09-12	16:00:00	12€	L'histoire de toto
<input type="radio"/>	2	2021-09-13	16:30:00	13€	Visite de totoville
<input type="radio"/>	3	2021-09-14	17:00:00	14€	Toto notre bien aimé
<input type="radio"/>	56	2021-10-25	19:37:00	30€	Test compte rendu
<input type="radio"/>	55	2021-11-01	20:00:00	5€	La teinture foirée
<input type="radio"/>	54	2021-11-12	10:00:00	30€	Les reines de Valadon

À gauche se trouve le tableau permettant à l'utilisateur de choisir la session qu'il veut modifier. Et, à droite se trouve le pop-up qui s'affiche uniquement lorsque l'utilisateur a sélectionné une session via les **inputs type radio** puis a cliqué sur le bouton Afficher.

## Jeu d'essais

**Modification d'une session :**

ACCUEIL

SESSION

INSCRIPTION SESSION

SESSIONS

de la session

12€

13€

14€

30€

5€

30€

MODIFIER UNE SESSION

VEUILLEZ SÉLECTIONNER LA SESSION QUE VOUS VOULEZ MODIFIER

Sélectionner	Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
<input type="radio"/>	1	2021-09-12	16:00:00	12€	L'histoire de toto
<input type="radio"/>	2	2021-09-13	16:30:00	13€	Visite de totoville
<input type="radio"/>	3	2021-09-14	17:00:00	14€	Toto notre bien aimé
<input type="radio"/>	56	2021-10-25	19:37:00	30€	Test compte rendu
<input type="radio"/>	55	2021-11-01	20:00:00	5€	La teinture foirée
<input type="radio"/>	54	2021-11-12	10:00:00	30€	Les reines de Valadon

Afficher

Fermer

MODIFIER UNE SESSION

25/10/2021

19:37

30

Test compte rendu

Modifier

Fermer

Nous souhaitons modifier l'heure de la session.

MODIFIER UNE SESSION

VEUILLEZ SÉLECTIONNER LA SESSION QUE VOUS VOULEZ MODIFIER

Sélectionner	Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
<input type="radio"/>	1	2021-09-12	16:00:00	12€	L'histoire de toto
<input type="radio"/>	2	2021-09-13	16:30:00	13€	Visite de totoville
<input type="radio"/>	3	2021-09-14	17:00:00	14€	Toto notre bien aimé
<input type="radio"/>	56	2021-10-25	19:37:00	30€	Test compte rendu
<input type="radio"/>	55	2021-11-01	20:00:00	5€	La teinture foirée
<input type="radio"/>	54	2021-11-12	10:00:00	30€	Les reines de Valadon

Afficher

Fermer

MODIFIER UNE SESSION

25/10/2021

20:40

30

Test compte rendu

Modifier

Fermer

Vérification sur le site :

TABLEAU POUR CONSULTER LES SESSIONS				
Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
1	2021-09-12	16:00:00	12€	L'histoire de toto
2	2021-09-13	16:30:00	13€	Visite de totoville
3	2021-09-14	17:00:00	14€	Toto notre bien aimé
56	2021-10-25	20:40:00	30€	Test compte rendu
55	2021-11-01	20:00:00	5€	La teinture foirée
54	2021-11-12	10:00:00	30€	Les reines de Valadon

Vérification sur phpmyadmin :



NUM_SESSION	DATE_SESSION	HEURE_SESSION	PRIX_SESSION	NOM_SESSION
1	2021-09-12	16:00:00	12	L'histoire de toto
2	2021-09-13	16:30:00	13	Visite de totoville
3	2021-09-14	17:00:00	14	Toto notre bien aimé
56	2021-10-25	20:40:00	30	Test compte rendu
55	2021-11-01	20:00:00	5	La teinture foirée
54	2021-11-12	10:00:00	30	Les reines de Valadon

Nous voyons que ça **FUNCTIONNE !!**

## Suppression d'une session

Nous allons gérer la suppression dans le même **Contrôleur** et dans le même **Modèle**.

Si l'utilisateur veut supprimer une session, il doit choisir via un **input type radio**, la session qu'il veut supprimer.

Pour supprimer nous avons besoin d'une requête, **Delete** et cette dernière se place dans le **Modèle**.

```
function deleteSession($numsession)
{
    try {

        $req = $this->conn->prepare("DELETE FROM session

                                WHERE NUM_SESSION = ?");

        $req->bindValue(1, $numsession);

        $req->execute();

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

}
```

Comme la **suppression** d'une session comprend tous les champs de la table **Session** nous mettons dans le **Where** la clé primaire **NUM\_SESSION**.

```

public function todo_Supprimer()

{

    $numsession = $this->post['session'];

    $this->modele->deleteSession($numsession);

}

```

Puis, comme pour l'**ajout** et la **modification**, nous créons une fonction **todo** que nous nommons **todo\_Supprimer**. Ensuite, nous récupérons les paramètres de la requête **deleteSession** à l'aide de la syntaxe **\$this->post**. Puis, avec **\$this->modele**, nous appelons la requête dont nous avons parlé à l'instant.

```

<table name="delete">

    <tr>

        <th colspan="6">VEUILLEZ SELECTIONNER LA SESSION QUE VOUS VOULEZ
        SUPPRIMER</th>

    </tr>

    <tr>

        <td><B>Sélectionner</B></td>

        <td><B>Numéro de la session</B></td>

        <td><B>Date de la session</B></td>

        <td><B>Heure de la session</B></td>

        <td><B>Prix de la session</B></td>

        <td><B>Nom de la session</B></td>

    </tr>

    <?php foreach ($c->data["liste"] = $c->modele->getSession() as
    $UneSession) { ?>

        <tr>

            <td><input value="<?php echo $UneSession->NUM_SESSION; ?>"
            name="session" type="radio" required></td>

```

```

        <td><?php echo $UneSession->NUM_SESSION; ?> </td>

        <td><?php echo $UneSession->DATE_SESSION; ?></td>

        <td><?php echo $UneSession->HEURE_SESSION; ?></td>

        <td><?php echo $UneSession->PRIX_SESSION; ?>€</td>

        <td><?php echo $UneSession->NOM_SESSION; ?></td>

    </tr>

    <?php }

    ?>

</div>

</table>

```

Pour gérer la suppression, nous créons un autre pop-up comprenant le tableau avec toutes les sessions et un **input type radio** afin de sélectionner la session que nous voulons supprimer.

```

<input type="submit" class="btn" name="todo" value="Supprimer" />

```

Puis nous rajoutant le **submit** que nous nommons **todo** avec pour **value** **Supprimer** (lien avec la fonction du **Contrôleur**).

Une fois allié au **CSS**, nous pouvons tester la **suppression**.

SUPPRIMER UNE SESSION

VEUILLEZ SELECTIONNER LA SESSION QUE VOUS VOULEZ SUPPRIMER

Sélectionner	Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
<input type="radio"/>	1	2021-09-12	16:00:00	12€	L'histoire de toto
<input type="radio"/>	2	2021-09-13	16:30:00	13€	Visite de totoville
<input type="radio"/>	3	2021-09-14	17:00:00	14€	Toto notre bien aimé
<input type="radio"/>	56	2021-10-25	20:40:00	30€	Test compte rendu
<input type="radio"/>	55	2021-11-01	20:00:00	5€	La teinture foirée
<input type="radio"/>	54	2021-11-12	10:00:00	30€	Les reines de Valadon

Supprimer

Fermer

 Jeu d'essais

Suppression d'une session :

Nous souhaitons supprimer la session n°56.

SUPPRIMER UNE SESSION

VEUILLEZ SELECTIONNER LA SESSION QUE VOUS VOULEZ SUPPRIMER

Sélectionner	Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
<input type="radio"/>	1	2021-09-12	16:00:00	12€	L'histoire de toto
<input type="radio"/>	2	2021-09-13	16:30:00	13€	Visite de totoville
<input type="radio"/>	3	2021-09-14	17:00:00	14€	Toto notre bien aimé
<input checked="" type="radio"/>	56	2021-10-25	20:40:00	30€	Test compte rendu
<input type="radio"/>	55	2021-11-01	20:00:00	5€	La teinture foirée
<input type="radio"/>	54	2021-11-12	10:00:00	30€	Les reines de Valadon

Supprimer
Fermer

Vérification sur le site :

TABLEAU POUR CONSULTER LES SESSIONS				
Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
1	2021-09-12	16:00:00	12€	L'histoire de toto
2	2021-09-13	16:30:00	13€	Visite de totoville
3	2021-09-14	17:00:00	14€	Toto notre bien aimé
55	2021-11-01	20:00:00	5€	La teinture foirée
54	2021-11-12	10:00:00	30€	Les reines de Valadon

Vérification sur phpmyadmin :

NUM_SESSION	DATE_SESSION	HEURE_SESSION	PRIX_SESSION	NOM_SESSION
1	2021-09-12	16:00:00	12	L'histoire de toto
2	2021-09-13	16:30:00	13	Visite de totoville
3	2021-09-14	17:00:00	14	Toto notre bien aimé
55	2021-11-01	20:00:00	5	La teinture foirée
54	2021-11-12	10:00:00	30	Les reines de Valadon

Nous voyons que ça **FONCTIONNE !!**

## Gestion d'erreur X

Pour gérer les gestions d'erreur, nous nous sommes aidé de l'élément **required**. En effet, ce dernier permet de mettre en forme les éléments obligatoires pour remplir correctement un formulaire.

Nous pouvons voir plusieurs exemples de code ci-dessous.

1er exemple :

```
<input type="date" name="datesession" value="" required>
```

```
<input type="time" name="heuresession" value="" required>
```

2ème exemple :

```
<input value="<?php echo $UneSession->NUM_SESSION; ?>" name="session" type="radio" required>
```

Allons voir à présent ce que ça donne du côté du site.

**AJOUTER UNE SESSION**

jj/mm/aaaa

! Veillez renseigner ce champ.

...:--

Saisir le prix de la session

Saisir le nom de la session

Enregistrer

Fermer

**MODIFIER UNE SESSION**

**VEUILLEZ SÉLECTIONNER LA SESSION QUE VOUS VOULEZ MODIFIER**

Sélectionner	Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
<input type="radio"/>	1	2021-09-12	16:00:00	12€	L'histoire de toto
<input type="radio"/>	2	2021-09-13	16:30:00	13€	Visite de totoville
<input type="radio"/>	3	2021-09-14	17:00:00	14€	Toto notre bien aimé
<input type="radio"/>	55	2021-11-01	20:00:00	5€	La teinture foirée
<input type="radio"/>	54	2021-11-12	10:00:00	30€	Les reines de Valadon

! Veillez sélectionner l'une de ces options.

Afficher

Fermer

## Inscription à une session

Nous commençons par créer une nouvelle **action** dans le **Contrôleur principal**.

```
$lesActions["inscription"] = "contrôleur.inscription.php";
```

Comme nous le voyons ci-dessus, nous avons appelé cette **action**, **inscription**.

---

## Inscription d'un congressiste à une session

Afin de gérer l'**inscription** et la **désinscription**, nous avons décidé de faire en début, un tableau contenant toutes les sessions ainsi qu'un input type radio pour que ce dernier serve dans les deux parties.

```
<table>

  <tr>

    <th colspan="6">VEUILLEZ SELECTIONNER UNE SESSION</th>

  </tr>

  <tr>

    <td><B>Sélectionner</B></td>

    <td><B>Numéro de la session</B></td>

    <td><B>Date de la session</B></td>

    <td><B>Heure de la session</B></td>

    <td><B>Prix de la session</B></td>

    <td><B>Nom de la session</B></td>

  </tr>

  <?php foreach ($c->data["liste"] = $c->modele->getSession() as $UneSession) { ?>

    <tr>

      <td><input type=radio name="NUM_SESSION" value="<?= $UneSession->NUM_SESSION ?>" required></td>

      <td><?php echo $UneSession->NUM_SESSION; ?> </td>

      <td><?php echo $UneSession->DATE_SESSION; ?></td>

      <td><?php echo $UneSession->HEURE_SESSION; ?></td>

      <td><?php echo $UneSession->PRIX_SESSION; ?>€</td>
```

```

        <td><?php echo $UneSession->NOM_SESSION; ?></td>

    </tr>

<?php }

?>

</table>

```

Comme nous pouvons le voir ci-dessus, il s'agit de nouveau d'un tableau classique regroupant toutes les sessions.

Cela devient plus intéressant lorsque nous passons à l'affichage des congressistes.

En effet, nous souhaitons faire afficher seulement les congressistes qui ne sont pas encore inscrits à une session. Cela veut dire que nous devons gérer une requête **NOT IN !!** Ouais !!

```

function getSessionListeCongressiste($NUM_SESSION)

{

    try {

        $req = $this->conn->prepare("SELECT * FROM
congressiste WHERE NUM_CONGRESSISTE

                                NOT IN (SELECT NUM_CONGRESSISTE FROM
participation_session WHERE NUM_SESSION = ?)");

        $req->bindValue(1, $NUM_SESSION);

        $req->execute();

        $resultat = $req->fetchAll(PDO::FETCH_OBJ);

    } catch (PDOException $e) {

        print "Erreur !: " . $e->getMessage();

        die();

    }

    return $resultat;
}

```



```
}
```

Ci-dessus se trouve cette magnifique requête **NOT IN**. Expliquons là : Nous souhaitons récupérer tous les congressistes en fonction du **NUM\_CONGRESSISTE** qui ne sont pas dans la table **participation\_session** et en fonction du **NUM\_SESSION**.

Maintenant que la requête est faite, allons l'adapter dans la **Vue**.

```
<form name="addCongressiste" class="form-container"
action="./?controleur=inscription" method="POST">

<input type="hidden" name="NUM_SESSION" value="<?= $c->post["NUM_SESSION"]
?>">

<h2>Inscrire un congressiste a une Session</h2>

</br>

<div class="container">

    <h1>Vous avez sélectionné la Session n°<?php echo $c->post["NUM_SESSION"]
?></h1>

    <table name="addCongressiste">

        <tr>

            <th colspan="2">VEUILLEZ SELECTIONNER UN CONGRESSISTE</th>

        </tr>

        <tr>

            <td><B>Nom du congressiste</B></td>

            <td><B>Sélectionner un congressiste</B></td>

        </tr>

        <?php

        ?>

        <?php foreach ($c->data["congressiste"] = $c->modele-
>getSessionListeCongressiste($c->post["NUM_SESSION"]) as $UnCongressiste) { ?>

            <tr>
```

```

        <td><?php echo $UnCongressiste->PRENOM_CONGRESSISTE; ?> <?php
echo $UnCongressiste->NOM_CONGRESSISTE; ?> </td>

        <td><input name="NUM_CONGRESSISTE[]" value="<?=$
$UnCongressiste->NUM_CONGRESSISTE ?>" type="checkbox"></td>

    </tr>

<?php }

?>

</table>

<?php

?>

</div>

</br></br>

<input type="submit" name="todo" value="Inscription" />

```

Tout d'abord dans notre **form**, nous mettons un champ caché dans lequel nous faisons un écho de **\$c→post["NUM\_SESSION"]**. Cela permettra de stocker le numéro de la session que nous aurons précédemment choisi.

Ensuite, pour indiquer qu'il s'agit d'un tableau et donc que nous pouvons choisir plusieurs congressistes nous mettons derrière le nom de l'**input (NUM\_CONGRESSISTE)** des crochets. Puis, dans le **value** nous mettons un **echo** du **NUM\_CONGRESSISTE** afin d'envoyer ce dernier lorsque l'utilisateur aura fait son choix puis cliquer sur le bouton **Inscription**.

De ce fait, dans ce pop-up nous aurons le **numéro de la session ainsi que celui du congressiste**.

Avant de passer à la suite, nous pouvons vérifier dans le code source si lorsque nous sélectionnons une session, ce dernier est bien passé en paramètre et sélectionné.

```

<div class="login-popup">
<div class="form-popup4" id="popupFormSuppression" style="">
    <form name="supCongressiste" class="form-container" action="./?controleur=inscription" method="POST">
        <input type="hidden" name="NUM_SESSION" value="1">

```

Puisque nous avons choisi la **session n° 1**, nous voyons qu'elle est bien passé en paramètre.

Nous pouvons voir ce que ça donne une fois allié au **CSS**.

**INSCRIRE UN CONGRESSISTE A UNE SESSION**

Vous avez sélectionné la Session n° 1

VEUILLEZ SELECTIONNER UN CONGRESSISTE	
Nom du congressiste	Sélectionner un congressiste
Thierry Bogusz	<input type="checkbox"/>
Claude Pasqualini	<input type="checkbox"/>
Alain Tournier	<input type="checkbox"/>

Inscription

Fermer

## 🎮 Jeu d'essais

Ajout d'un congressiste à un session :

Nous souhaitons ajouter le congressiste Thierry Bogusz à la session n°1.

**INSCRIRE UN CONGRESSISTE A UNE SESSION**

Vous avez sélectionné la Session n° 1

VEUILLEZ SELECTIONNER UN CONGRESSISTE	
Nom du congressiste	Sélectionner un congressiste
Thierry Bogusz	<input checked="" type="checkbox"/>
Claude Pasqualini	<input type="checkbox"/>
Alain Tournier	<input type="checkbox"/>

Inscription

Fermer

Vérification sur phpmyadmin :

NUM_SESSION	NUM_CONGRESSISTE
1	1
1	2
3	2
2	3
2	4
1	5
3	5
51	5

Nous voyons que ça **FONCTIONNE !!**

## Désinscription d'un congressiste à une session

Nous allons gérer la désinscription dans le même **Contrôleur** et dans le même **Modèle**.

Si nous devons procéder à une désinscription, tout comme pour l'inscription, nous devons sélectionner une session via un **input type radio** puis cliquer sur le bouton **SupprimerCongressiste** qui ouvrira un pop-up listant les congressistes qui ont déjà été inscrit à la session choisie.

Il nous donc créer une requête faisant l'inverse de tout à l'heure (afficher seulement les congressistes qui sont déjà inscrits à une session). Cela veut dire que nous devons gérer une requête **IN**.

```
function getSessionListeCongressiste2($NUM_SESSION)
{
    try {

        $req = $this->conn->prepare("SELECT * FROM
congressiste WHERE NUM_CONGRESSISTE

                                IN (SELECT NUM_CONGRESSISTE FROM
participation_session WHERE NUM_SESSION = ?)");

        $req->bindValue(1, $NUM_SESSION);

        $req->execute();

        $resultat = $req->fetchAll(PDO::FETCH_OBJ);

    } catch (PDOException $e) {
```

```

        print "Erreur !: " . $e->getMessage();

        die();

    }

    return $resultat;

}

```

Maintenant que la requête est faite, nous pouvons à présent l'adapter dans la **Vue**.

```

<form name="supCongressiste" class="form-container"
action="./?controleur=inscription" method="POST">

    <input type="hidden" name="NUM_SESSION" value="<?= $c->post["NUM_SESSION"]
?>">

    <h2>Supprimer un congressiste a une Session</h2>

    </br>

    <div class="container">

        <table name="supCongressiste">

            <tr>

                <th colspan="2">VEUILLEZ SELECTIONNER UN CONGRESSISTE</th>

            </tr>

            <tr>

                <td><B>Nom du congressiste</B></td>

                <td><B>Sélectionner un congressiste</B></td>

            </tr>

            <?php

                ?>

                <?php foreach ($c->data["congressiste"] = $c->modele-
>getSessionListeCongressiste2($c->post["NUM_SESSION"]) as $UnCongressiste) {
                ?>

```

```

        <tr>

            <td><?php echo $UnCongressiste->PRENOM_CONGRESSISTE; ?>
<?php echo $UnCongressiste->NOM_CONGRESSISTE; ?> </td>

            <td><input name="NUM_CONGRESSISTE[]" value="<?=$UnCongressiste->NUM_CONGRESSISTE ?>" type="checkbox"></td>

        </tr>

        <?php }

        ?>

    </table>

    <?php

        ?>

</div>

</br></br>

<input type="submit" name="todo" value="Suppression" /><br />

```

Tout d'abord dans notre **form**, comme précédemment, nous mettons un champ caché dans lequel nous faisons un **echo** de **\$c→post["NUM\_SESSION"]**. Cela permettra de stocker le numéro de la session que nous aurons précédemment choisi.

Ensuite, pour indiquer qu'il s'agit d'un tableau et donc que nous pouvons choisir plusieurs congressistes nous mettons derrière le nom de l'**input (NUM\_CONGRESSISTE)** des crochets. Puis, dans le **value** nous mettons un **echo** du **NUM\_CONGRESSISTE** afin d'envoyer ce dernier lorsque l'utilisateur aura fait son choix puis cliquer sur le bouton **SupprimerCongressiste**.

À présent, nous pouvons voir ce que ça donne une fois allié au **CSS**.

**SUPPRIMER UN CONGRESSISTE A UNE SESSION**

Vous avez sélectionné la Session n° 3

VEUILLEZ SELECTIONNER UN CONGRESSISTE	
Nom du congressiste	Sélectionner un congressiste
Thierry Bogusz	<input type="checkbox"/>
Salah El Mahtaoui	<input type="checkbox"/>

Suppression

Fermer

## 🎮 Jeu d'essais

Annulation de l'inscription d'un congressiste à un session :

Nous souhaitons désinscrire la congressiste Agnès Bourgeois de la session n°1.

**SUPPRIMER UN CONGRESSISTE A UNE SESSION**

Vous avez sélectionné la Session n° 1

VEUILLEZ SELECTIONNER UN CONGRESSISTE	
Nom du congressiste	Sélectionner un congressiste
Agnès Bourgeois	<input checked="" type="checkbox"/>
Salah El Mahtaoui	<input type="checkbox"/>
Thierry Bogusz	<input type="checkbox"/>

Suppression

Fermer

Vérification sur phpmyadmin :

Le NUM\_CONGRESSISTE d'Agnès Bourgeois est le 1 et nous voyons qu'il ne se situe pas dans la capture.

NUM_SESSION	NUM_CONGRESSISTE
1	5
1	2
2	3
2	4
3	2
3	5
51	5

Nous voyons que ça **FUNCTIONNE !!**

## Gestion d'erreur **X**

Pour gérer les gestions d'erreur, comme pour la première partie, nous nous sommes aidé de l'élément **required**. En effet, ce dernier permet de mettre en forme les éléments obligatoires pour remplir correctement un formulaire.

Nous pouvons voir un exemple de code ci-dessous.

Exemple :

```
<input type=radio name="NUM_SESSION" value="<?= $UneSession->NUM_SESSION ?>" required>
```

Allons voir ce que ça donne côté site.

Sélectionner	Numéro de la session	Date de la session	Heure de la session	Prix de la session	Nom de la session
<input type="radio"/>	1	2021-09-12	16:00:00	12€	L'histoire de toto
<input type="radio"/>	2	2021-09-13	16:30:00	13€	Visite de totoville
<input type="radio"/>	3	2021-09-14	17:00:00	14€	Toto notre bien aimé
<input type="radio"/>	55	2021-11-01	20:00:00	5€	La teinture foirée
<input type="radio"/>	54	2021-11-12	10:00:00	30€	Les reines de Valadon

Cependant, l'élément **required** ne fonctionnait pas comme il fallait pour le reste des gestions d'erreur. Nous avons dû donc les gérer autrement.

Dans la fonction **todo\_inscription** qui concerne l'inscription d'un congressiste à une session, nous avons rajouté cet élément de code.

```
if (isset($_POST['NUM_CONGRESSISTE'])) {
```



```

        $this->modele->insertCongressiste($this-
>post["NUM_CONGRESSISTE"], $this->post["NUM_SESSION"]);

    } else {

        echo '<p class="style_css">⚠ Vous devez sélectionner
au moins un congressiste ! ⚠</p>';
    }
}

```

Dans l'extrait de code ci-dessus, nous testons si nous sélectionnons un congressiste. Dans ce cas, nous procédons à l'insertion, mais dans le cas inverse, nous affichons un message d'erreur.

Une fois alliées au **CSS**, nous pouvons aller voir ce que ça donne sur le site.

Nous avons fait la même chose pour gérer l'erreur dans l'annulation de l'inscription d'un congressiste à une session.

### Bonus : Gérer la modification du nom du bouton todo

Nous voulions modifier le nom de deux **button** afin de les rendre plus clairs, mais comme ils étaient liés aux fonctions **todo** que nous avons créé dans le **Contrôleur** cela poser un problème.

C'est pourquoi avec l'aide de Mr Bogusz, nous avons modifié notre code **PHP** puis créé **deux fonctions JavaScript** afin qu'elles prennent en charge ces changements.

```

<div class="open-btn">

    <input type="hidden" id="todo" name="todo" value="">

    <button onclick="validinscrire()" class="open-button">Inscrire à un
Congressiste à une Session </button>

</div>

```

```

<div class="open-btn">

    <button onclick="validSupprimer()" class="open-button">Supprimer un
    Congressiste à une Session </button>

</div>

```

Comme nous le voyons ci-dessus, nous avons un **input type hidden** qui prend en charge les deux **todo**, puis deux **button** qui appellent les fonctions **JavaScript** dont nous parlerons tout à l'heure.

```

function validinscrire() {

    document.getElementById("todo").value = "Inscrire";

    document.forms[0].submit();

}

function validSupprimer() {

    document.getElementById("todo").value = "SupprimerCongressiste";

    document.forms[0].submit();

}

```

Ces deux fonctions font le lien avec les fonctions todo créé dans le contrôleur. Puis elles prennent en charge l'envoi des données vers le formulaire.

De cette façon, nous pouvons mettre le nom que l'on veut dans nos button car la prise en charge du todo est détournée.

Nous pouvons voir le résultat du côté du navigateur.

