

# COMPTE RENDU PPE 2.2

## Equipe 1 - Roche/Rodrigues



### Présentation et objectifs

Nous sommes l'**équipe 1**, composée de **Rodrigues Quentin** et de **Roche Sébastien**. Nous avons choisi la **mission 2**.

#### Nous avons 2 taches à accomplir :

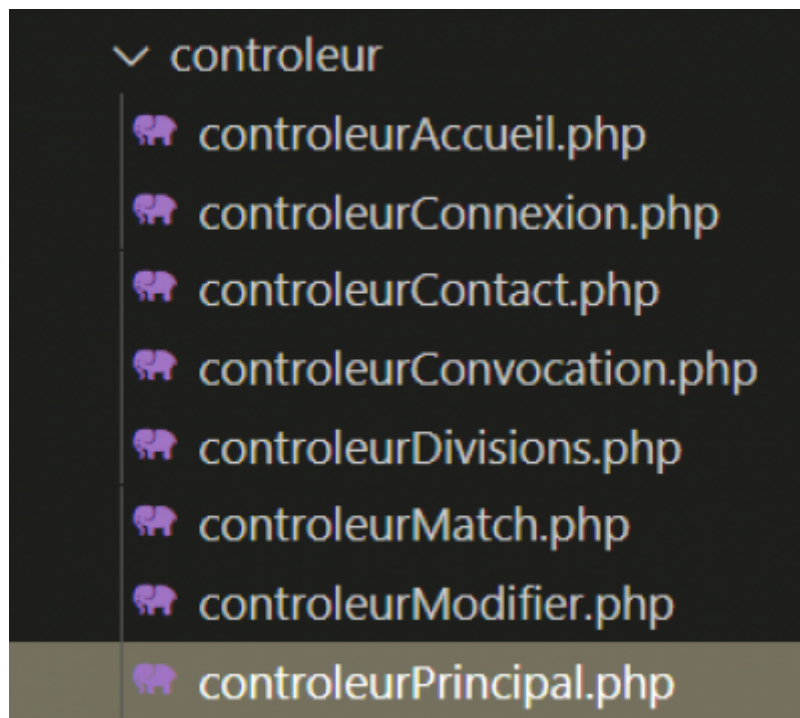
- **Gestion des divisions** : Cette tâche doit permettre la **création**, la **modification** et la **suppression d'une division**.
- **Envoi des convocations aux arbitres** : Pour un **match** donné vous devez générer sous forme d'un **fichier PDF** l'ensemble des convocations des **arbitres** . La convocation doit contenir : les informations sur le **championnat**, sur le **match** lui-même, indiquer qui est son **co-arbitre**, ainsi que le **montant de sa prestation**.

### Apparence et MVC

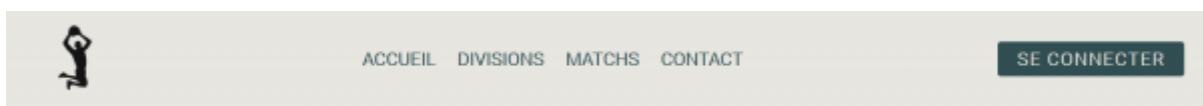
Voici notre page d'accueil :



Notre site est structuré comme demandé en **MVC**



Dans la **barre de navigation**, chaque bouton fait référence à un **?action** que le **contrôleur principal** récupère.



Voici le code html de la **barre de navigation** :

```
<header class="nav-bar">
  <a href="./?action=accueil"></a>
  <nav>
```

```

        <ul class="nav-area">
            <li><a href="./?action=accueil">Accueil</a></li>
            <li><a href="./?action=divisions">Divisions</a></li>
            <li><a href="./?action=match">Matches</a></li>
            <li><a href="./?action=contact">Contact</a></li>
        </ul>
    </nav>
    <a class="btn-area" href="?action=connexion">Se connecter</a>
</header>

```

Voici le code du **contrôleur Principal** :

```

<?php
function controleurPrincipal($action){
    $lesActions = array();
    $lesActions["default"] = "controleurAccueil.php";
    $lesActions["accueil"] = "controleurAccueil.php";
    $lesActions["divisions"] = "controleurDivisions.php";
    $lesActions["match"] = "controleurMatch.php";
    $lesActions["contact"] = "controleurContact.php";
    $lesActions["modifier"] = "controleurModifier.php";
    $lesActions["convocation"] = "controleurConvocation.php";
    $lesActions["connexion"] = "controleurConnexion.php";

    if (array_key_exists ( $action , $lesActions )){
        return $lesActions[$action];
    }
    else{
        return $lesActions["default"];
    }
}
?>

```

On peut voir que le **contrôleur principal** s'occupe de récupérer les **?actions** et de les rediriger vers les pages adéquates

A chaque nouvel **?action** cliquable (et qui , il faudra faire un nouvel **?action** dans le **contrôleur principal** (comme nous avons fait pour le ?action **Modifier** de la page **Division**)

```

$lesActions["modifier"] = "controleurModifier.php";

```

Le **contrôleur principal** renvoi donc les ?actions vers les **différents contrôleurs** qui s'occupe d'afficher la **vue** (c'est à dire ce qui sera vue par l'utilisateur) et effectuer les **fonctions** en rapport avec la page.

Voici par exemple le contrôleur contact qui affiche juste la page HTML contact

```

<?php
    include "./vue/contact.html.php";
?>

```

L'ensemble des **fonctions** qui seront utilisées dans notre site sont dans le fichier **./modele/bd.divisions.inc.php**.

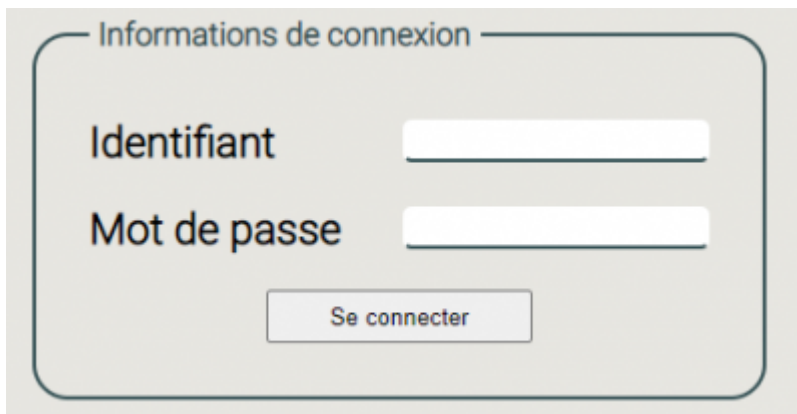
A chaque fois qu'il faudra utiliser une de ces fonctions dans un contrôleur, il faudra include le fichier **./modele/bd.divisions.inc.php**

```
include_once "./modele/bd.divisions.inc.php";  
$listeMatch = getAllMatch();
```

## Connexion et Authentification

Une authentification est disponible sur le site, il permet aux responsables d'accéder aux différentes interfaces. Mais également les arbitres, qui pourront voir les divisions et seulement celle ci. (Les identifiants sont uniques, donc il n'y a pas de système d'inscription)

L'interface de connexion directement accessible depuis le menu se présente ainsi :



Ainsi, nous pouvons voir la fonction de connexion qui est comme ceci :

```
session_start();  
  
if(!isset($_SESSION["username"])){  
    $isLogin = false;  
  
    if(isset($_POST['identifiant'])){  
        $cnx = connexionPDO();  
        $req = $cnx->prepare("SELECT * FROM user WHERE identifiant =  
?");  
        $req->bindValue(1,$_POST['identifiant']);  
        $req->execute();  
        $resultat = $req->fetch(PDO::FETCH_OBJ);  
        if(!empty($resultat)){  
            if(password_verify($_POST['motdepasse'], $resultat->password)){  
                $_SESSION["username"] = $_POST['identifiant'];  
                $isLogin = true;  
            }  
        }  
    }  
}
```

```

    }
} else{
    $isLogin = true;
}

```

Dans le langage PHP, le variable superglobal SESSION permet de gérer différentes sessions, nous avons utilisés ce système pour savoir qui est connecté. Le mot de passe est crypté avec la méthode de hashage BCRYPT de php.

Maintenant que la connexion est faite, il faut tout simplement ajouter des "if" pour vérifier l'utilisateur en question. Voici un exemple pour l'affichage du menu :

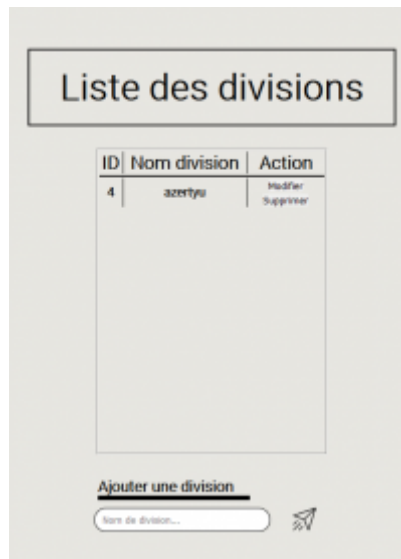
```

<header class="nav-bar">
    <a href="./?action=accueil"></a>
    <nav>
        <ul class="nav-area">
            <li><a href="./?action=accueil">Accueil</a></li>
            <?php
                if(isset($_SESSION['username'])){
                    if($_SESSION['username'] == "arbitre" ||
$_SESSION['username'] == "responsable"){
                        ?>
                        <li><a href="./?action=divisions">Divisions</a></li>
                    <?php
                }}
                ?>
            <?php
                if(isset($_SESSION['username'])){
                    if($_SESSION['username'] == "responsable"){
                        ?>
                        <li><a href="./?action=match">Matches</a></li>
                    <?php
                }
                ?>
            <li><a href="./?action=contact">Contact</a></li>
        </ul>
    </nav>
    <?php
    if(!isset($_SESSION['username'])){
        echo "<a class=\"btn-area\" href=\"?action=connexion\">Se
connecter</a>";
    } else{
        echo "<a class=\"btn-area\" href=\"controleur/logout.php\">Se
déconnecter</a>";
    }
    ?>
</header>

```

# Gestion des divisions

L'ensemble des divisions est regroupé sous la forme d'un tableau que voici :



Voici le code du tableau :

```
<?php
    if(!isset($_POST['get-id-div-modification'])) {

        ?>
        <fieldset id="fieldset-liste-division">

            <table id="division-table">
                <tr class="division-ligne-first">
                    <td class="division-cellule-first division-cellule-
id">ID</td>
                    <td class="division-cellule-first division-cellule-
nom">Nom division</td>
                    <td class="division-cellule-first division-cellule-
action">Action</td>
                </tr>
                <tr>
                    <td colspan='3' bgcolor='black' style='font-
size:1pt'>NBSP</td>
                </tr>

                <?php
                    foreach ($listeDivisions as $ligne) {
                        echo "<form method=\"post\"
action=\"../?action=divisions\">";
                        echo "<input type=\"hidden\" name=\"get-id-
div-modification\" value=\"\$ligne->num_division\">";
```

```

        echo "<tr class=\"division-ligne\">";
        echo "<td class=\"division-cellule-id-";
print\">$ligne->num_division</td>";
        echo "<td class=\"division-cellule-nom-";
print\">$ligne->nom_division</td>";
        echo "<td class=\"division-cellule-action-";
print\"><input type=\"submit\" name=\"modifier\" value=\"Modifier\"
id=\"form-modifier\"> </form>
        <form method=\"post\"
action=\"../?action=divisions\"><input type=\"submit\" name=\"supprimer\"
value=\"Supprimer\" id=\"form-supprimer\"></a></input>";
        echo "<input type=\"hidden\" name=\"get-id-";
div\" value=\"$ligne->num_division\">";
        echo "</form>";
        echo "</tr>";
        echo "<tr><td><br></td></tr>";

    }

?>

</table>

</fieldset>
<fieldset id="fieldset-ajout-division">
    <legend>Ajouter une division</legend>
    <form method = "post" action = "?action=divisions">
        <input type = "text" name = "new-variable-insertion"
id = "input-insertion-division" placeholder = "Nom de division...">
        <input type = "submit" name = "submit-insertion-division"
id = "submit-insertion-division-id">
    </form>
</fieldset>

<?php
} else{
?>

    <div id="div-modification-division">
        <form id="form-modification-division" method="post"
action="?action=divisions">
            <h4 id="h4-tilttitle-modification">Division numéro
<?= $_POST['get-id-div-modification'] ?></h4>
            <input id="input-modified-division" type="text"
name="input-modified-division-name" placeholder="<?=
$laDivision->nom_division ?>">
            <input type="hidden" name="hidden-num-division-
modification" value="<?= $_POST['get-id-div-modification'] ?>">

```

```
        
        <input type="submit" name="submit-modifier-
division">
                </form>
        </div>
    <?php
    }
?>
<?php
    if(isset($_POST['supprimer'])){
        if($isPossible == false){
            echo "<div id=\"alert-suppression-impossible\">";
            echo "<div id=\"alert-suppression-texte\">";
            echo "La suppression n'est pas possible, un
championnat est en cours.";
            echo "</div>";
            echo "</div>";
        }
    }
?>
</div>
</div>
```

L'ensemble du tableau est affiché grace à la fonction **getAllDivisions** :

```
function getAllDivisions(){
    $resultat = array();
    try {
        $cnx = connexionPDO();
        $req = $cnx->prepare("select * from division");
        $req->execute();
        $resultat = $req->fetchAll(PDO::FETCH_OBJ);
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }
    return $resultat;
}
```

Mais le **tableau** est aussi composé de **form** pour chaque ligne.

Si une demande de **suppression** est faite alors le form renvoie en **\$\_POST** via un **?action** vers **controleurDivisions.php**.

```
include_once "./modele/bd.divisions.inc.php";

if(isset($_POST['supprimer'])){
    $isPossible = removeDivisionById($_POST['get-id-div']);
```



```
}

```

Si le `$_POST` est récupère alors la fonction `removeDivisionById()` est exécuté avec comme `ID`, l'`ID` de la **division du form** récupérée dans le tableau

C'est à peu prêt le même principe pour la **modification** sauf que c'est la fonction `getDivisionById()` qui est exécuté dans le **contrôleur** :

```
function getDivisionById($id){
    try{
        $cnx = connexionPDO();
        $req = $cnx->prepare("SELECT * FROM division WHERE num_division
= ?");
        $req->bindValue(1,$id);
        $req->execute();
        $resultat = $req->fetch(PDO::FETCH_OBJ);
    } catch (PDOException $e){
        print "Erreur ;; ".$e->getMessage();
        die();
    }
    return $resultat;
}
```

```
if(isset($_POST['get-id-div-modification'])){
    $laDivision = getDivisionById($_POST['get-id-div-modification']);
}
```

L'affichage de l'**interface de modification** se fait juste **avant** le tableau dans la vue avec un `if`. Si une demande de modification est faite alors l'interface s'affiche :

```
if(!isset($_POST['get-id-div-modification'])){
//tableau vue précédemment//
} else{
    ?>
    <div id="div-modification-division">
        <form id="form-modification-division" method="post"
action="?action=divisions">
            <h4 id="h4-tiltle-modification">Division numéro
<?= $_POST['get-id-div-modification'] ?></h4>
            <input id="input-modified-division" type="text"
name="input-modified-division-name" placeholder="<?=
$laDivision->nom_division ?>">
            <input type="hidden" name="hidden-num-division-
modification" value="<?= $_POST['get-id-div-modification'] ?>">
            
            <input type="submit" name="submit-modifier-
division">
        </form>
    </div>
```

```
<?php
```

```
}
```

L'interface de modification se présente comme ceci :



Si l'icône de confirmation de modification est pressée, la fonction `modifierDivision()` est effectuée dans le contrôleur :

```
function modifierDivision($newNom, $id){
    try{
        $cnx = connexionPDO();
        $req = $cnx->prepare("UPDATE division SET nom_division = ? WHERE
division.num_division = ?");
        $req->bindValue(1, $newNom);
        $req->bindValue(2, $id);
        $req->execute();
        return true;
    } catch (PDOException $e){
        print "Erreur : ".$e->getMessage();
        return false;
        die();
    }
}
```

```
if(isset($_POST['submit-modifier-division'])){
    $isModifier = modifierDivision($_POST['input-modified-division-
name'], $_POST['hidden-num-division-modification']);
    header("Refresh:0");
}
```

Mais il y a également une interface d'ajout de division :

```
<fieldset id="fieldset-ajout-division">
    <legend>Ajouter une division</legend>
    <form method = "post" action="?action=divisions">
        <input type="text" name="new-variable-insertion"
id="input-insertion-division" placeholder="Nom de division...">
        <input type="submit" name="submit-insertion-division"
id="submit-insertion-division-id">
    </form>
</fieldset>
```

Si le submit d'ajout est pressé alors la fonction `ajoutDivision()` est exécuté dans contrôleur :



```
function ajoutDivision($nom){
    try{
        $cnx = connexionPDO();
        $req = $cnx->prepare("INSERT INTO division VALUES(NULL, ?)");
        $req->bindValue(1, $nom);
        $req->execute();
        return true;
    } catch (PDOException $e){
        print "Erreur : ".$e->getMessage();
        return false;
        die();
    }
}
```

```
if(isset($_POST['submit-insertion-division'])){
    $isInsert = ajoutDivision($_POST['new-variable-insertion']);
    header("Refresh:0");
}
```

## Match et Convocations aux arbitres

Voici la page **Match**, on y retrouve un **tableau** qui récupère l'ensemble des informations de notre **base de donnée** et qui les affiche.

# Match du championnat

N° Match	Date Match	Equipe 1	Equipe 2	N° Arbitre Principal	Convocation
1	2021-03-16	Equipe de Raimon	Equipe de Valadon	1	
2	2021-03-11	Equipe de Gueret	Spurs	3	

Cette affichage se fait grace à la fonction **getAllMatch()** que voici :

```
function getAllMatch(){
    $resultat = array();
    try {

        $conn = connexionPDO();
        $req = $conn->prepare("SELECT num_match, date_match,
principal.nom_arbitre as nom_arbitre_principal, secondaire.nom_arbitre as
nom_arbitre_secondaire, equipe1.nom_equipe as nom_equipe_1,
equipe2.nom_equipe as nom_equipe_2, match.num_arbitre_p as
num_arbitre_principal FROM `match` INNER JOIN arbitre principal ON
principal.num_arbitre = match.num_arbitre_p INNER JOIN arbitre secondaire ON
secondaire.num_arbitre = match.num_arbitre_s INNER JOIN equipe equipe1 ON
equipe1.num_equipe = match.num_equipe_1 INNER JOIN equipe equipe2 ON
equipe2.num_equipe = match.num_equipe_2");

        $req->execute();

        $resultat = $req->fetchAll(PDO::FETCH_OBJ);
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }
    return $resultat;
}
```

Et Voici le **controleurMatch.php** :

```
<?php
    include_once "../modele/bd.divisions.inc.php";
    $listeMatch = getAllMatch();
    include "../vue/match.html.php";
?>
```

Ensuite, apres avoir récupéré les informations nécessaires grâce à la **requête SQL**, on les affiche dans la vue que voici :

```
<div id="match_area">

    <div id="title_div_match">
        <p id="title_match">Match du championnat</p>
    </div>



    <fieldset id="match_div_table">
        <table id="match_table">
            <form method="post" action="./?action=match">
                <tr class="table_top_match">
                    <td class="cellule_match">N° Match</td>
                    <td class="cellule_match">Date
Match</td>
                    <td class="cellule_match">Equipe 1</td>
```

```

                <td class="cellule_match">Equipe 2</td>
                <td class="cellule_match">N° Arbitre
Principal</td>
                <td
class="cellule_match">Convocation</td>
            </tr>
            <tr>
            <td colspan='6' bgcolor='black' style='font-
size:1pt'>NBSP</td>
            </tr>
            <?php
            foreach ($listeMatch as $ligne) {
            ?>
                <tr class="match_row">
                <td class="cellule_match"><?=
"$ligne->num_match"?> </td>
                <td class="cellule_match"><?=
"$ligne->date_match"?> </td>
                <td class="cellule_match"><?=
"$ligne->nom_equipe_1"?> </td>
                <td class="cellule_match"><?=
"$ligne->nom_equipe_2"?> </td>
                <td class="cellule_match"><?=
"$ligne->num_arbitre_principal"?> </td>
                <td class="cellule_match"><a
href="./?action=convocation&numero=<?= "$ligne->num_match"?>" download> </img></a></
                <td>
            </tr>
            <td colspan='6' bgcolor='black' style='font-
size:1pt'>NBSP</td>
            </tr>
            <?php }
            ?>
            </form>
            </table>
            </fieldset>
        </div>

```

Le **foreach (\$listeMatch as \$ligne)** permet de le répéter pour une ligne du tableau un résultat de la **fonction**. Et **\$listeMatch** correspond au **\$listeMatch** du **controleurMatch.php**

N° Match	Date Match	Equipe 1	Equipe 2	N° Arbitre Principal	Convocation
1	2021-03-16	Equipe de Raimon	Equipe de Valadon	1	
2	2021-03-11	Equipe de Gueret	Spurs	3	

- L'**icon PDF** renvoie une action vers le **contrôleur Principal** qui le renvoi à son tour vers le **controleurConvocation.php**
- L'**affichage de la convocation** se fait pour le **match sélectionné** grace au **numéro du match** qui passe dans l'**URL** et qui est récupéré par **\$\_GET**

```
href="./?action=convocation&numero=<?=" $ligne->num_match"?>
```

```
<?php
include_once "./modele/bd.divisions.inc.php";
$id = $_GET['numero'];
$convoc = getMatchById($id);

include ("./modele/createpdf.php");
?>
```

Comme on peut le voir dans ce code, il est récupéré et utilisé avec la fonction **getMatchById()** que voici :

```
function getMatchById($id){
    try{
        $cnx = connexionPDO();
        $req1 = $cnx->prepare("SELECT num_match, date_match,
championnat.num_championnat as numero_champio, nom_championnat,
principal.nom_arbitre as nom_arbitre_principal, secondaire.nom_arbitre as
nom_arbitre_secondaire, equipe1.nom_equipe as nom_equipe_1,
equipe2.nom_equipe as nom_equipe_2,
montant_deplt_p, principal.prenom_arbitre as
prenom_arbitre_principal, secondaire.prenom_arbitre as
prenom_arbitre_secondaire, num_arbitre_s, num_arbitre_p FROM `match` INNER
JOIN arbitre principal ON principal.num_arbitre = match.num_arbitre_p INNER
JOIN arbitre secondaire ON secondaire.num_arbitre = match.num_arbitre_s
INNER JOIN equipe equipe1 ON equipe1.num_equipe = match.num_equipe_1 INNER
JOIN equipe equipe2 ON equipe2.num_equipe = match.num_equipe_2 INNER JOIN
championnat ON equipe2.num_championnat = championnat.num_championnat WHERE
match.num_match = ?");

        $req1 -> bindValue(1,$id);
        $req1->execute();
        $resultat = $req1->fetchAll(PDO::FETCH_OBJ);
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage();
        die();
    }
}
```

```

return $resultat;
}

```

Dans cette requête, nous sommes obligé de passer par des **alias** car on récupère **plusieurs informations avec le même** nom de table et de colonne. Si on ne fait pas ceci, on va se retrouver avec les memes **equipes** et les mêmes **arbitres**, comme ceci :

Arbitre Principal	
Nom :	SINGH
Prenom :	Liberty
Montant Prestation :	12

Arbitre Secondaire	
Nom :	SINGH
Prenom:	Liberty

Ensuite, l'ensemble de ces informations vers la page **createpdf.php**, ou elles sont affichées sous forme d'une **convocation** et d'un tableau grâce à la librairie **FPDF**(<http://www.fpdf.org/>) Voici le code du PDF :

```

<?php
require('./librairies/fpdf.php');

foreach ($convoc as $ligne) {

$pdf = new FPDF();
$pdf -> AddPage();
//image
$pdf->Image('./images/convoc_icon.png',6,6,50);
// police
$pdf->SetFont('Arial','B',20);
// Centre le titre
$pdf->Ln(10);
$pdf->SetX((190-40)/2);
// Couleur du texte et du fond du titre
$pdf->SetDrawColor(221,221,221,1);
$pdf->SetFillColor(49, 78, 82);
$pdf->SetTextColor(242, 161, 84);
// Taille du contour
$pdf->SetLineWidth(1);

// Titre
// Cellule (Longueur, Hauteur, Contenu, Bordure, nextline parametters,
C=CENTER , Remplissage)
$title = 'Convocation';
$pdf->SetTitle($title);
$pdf->Cell(60, 19, $title, 1, 1, 'C', true);
// Saut de ligne
$pdf->Ln(15);
$pdf->SetFont('Arial','B',15);

```

```
//Tableau 1
  //Ligne 1
$pdf->Ln(10);
$pdf->SetX(55);
$pdf->SetText(0, 0, 0);
$pdf->SetX((150-60)/2);
$pdf->Cell(120, 9, 'Informations generales', 1, 1, 'C');
$pdf->SetX((150-60)/2);
//Ligne 2
$pdf->Cell(60, 10, 'Numero Match :', 1, 0, 'C');
$pdf->Cell(60, 10, $id, 1, 1, 'C');

//Ligne 3
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Equipe 1 :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->nom_equipe_1, 1, 1, 'C');
//Ligne 4
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Equipe 2 :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->nom_equipe_2, 1, 1, 'C');

//Tableau 2
  //Ligne 1
$pdf->Ln(10);
$pdf->SetX(55);
$pdf->SetX((150-60)/2);
$pdf->Cell(120, 9, 'Arbitre Principal', 1, 1, 'C');
//Ligne 2
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Numero Arbitre :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->num_arbitre_p, 1, 1, 'C');
//Ligne 3
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Nom :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->nom_arbitre_principal, 1, 1, 'C');
//Ligne 4
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Prenom :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->prenom_arbitre_principal, 1, 1, 'C');
//Ligne 5
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Montant Prestation :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->montant_deplt_p, 1, 1, 'C');

//Tableau 3
  //Ligne 1
$pdf->Ln(10);
$pdf->SetX(55);
$pdf->SetX((150-60)/2);
```



```
$pdf->Cell(120, 9, 'Arbitre Secondaire', 1, 1, 'C');
//Ligne 2
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Numero Arbitre :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->num_arbitre_s, 1, 1, 'C');
//Ligne 3
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Nom :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->nom_arbitre_secondaire, 1, 1, 'C');
//Ligne 4
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Prenom:', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->prenom_arbitre_secondaire, 1, 1, 'C');

//Tableau 4
  //Ligne 1
$pdf->Ln(10);
$pdf->SetX((150-60)/2);
$pdf->Cell(120, 9, 'Informations sur le championnat', 1, 1, 'C');
$pdf->SetX((150-60)/2);
//Ligne 2
$pdf->Cell(60, 10, 'Numero :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->numero_champio, 1, 1, 'C');
//Ligne 3
$pdf->SetX((150-60)/2);
$pdf->Cell(60, 10, 'Nom :', 1, 0, 'C');
$pdf->Cell(60, 10, $ligne->nom_championnat, 1, 1, 'C');

// Positionnement à 1,5 cm du bas
$pdf->SetY(266);
// Police Arial italique 8
$pdf->SetFont('Arial', 'I', 8);
// Numéro de page
$pdf->Cell(0, 10, 'Page ' . $pdf->PageNo(), 0, 0, 'C');

//Nom du fichier (I= valeur par default, Nom, UTF8 BOOLEAN )
$pdf->Output('I', 'Convocation Match ' . $id , false);
}
?>
```

Voici à quoi ressemble ce **PDF** :



## Convocation

Informations generales	
Numero Match :	1
Equipe 1 :	Equipe de Raimon
Equipe 2 :	Equipe de Valadon

Arbitre Principal	
Numero Arbitre :	1
Nom :	SINGH
Prenom :	Liberty
Montant Prestation :	12

Arbitre Secondaire	
Numero Arbitre :	2
Nom :	Parker
Prenom:	Tony

Informations sur le championnat	
Numero :	1
Nom :	RLCS Saison 1

Page 1

From:

<http://ppe.boonum.fr/> - **AP.SIO**

Permanent link:

<http://ppe.boonum.fr/doku.php?id=slam:ws:2021:sio:ap2.2:equipe1:accueil>

Last update: **2021/04/27 00:52**

