

```

1  <?php
2  /*****
3  AUTEUR      : Constantin Herrmann
4  LIEU        : CFPT Informatique Genève
5  DATE        : 14.06.2018
6
7  TITRE PROJET: KidsGames Geneva Score
8
9  TITRE PAGE  : game.c
10 DESCRIPTION : Ce script contient toutes les fonctions concernant les matchs
11 VERSION     : 1.0
12 *****/
13
14 /**
15  * Création d'un nouveau MATCH
16  *
17  * @param int $Time -> L'id de l'heure à laquelle à lieu le match
18  * @param int $day -> L'id du jour ou à lieu le match
19  * @param int $field -> L'id du terrains
20  * @param int $arbitre -> L'id de l'Arbitres
21  * @param array $Teams -> un tableau avec les id des équipes qui jouent le match
22  */
23 function CreateNewGame($time, $day, $field, $arbitre, $teams, $sport){
24     static $query = null;
25     if ($query == null) {
26         $req = "INSERT INTO `Games`(`idArbitre`, `idTerrain`, `idJour`, `idTime`,
27         • `idSport`) VALUES (:arbitre,:terrain,:jour,:heure,:sport)";
28         $query = connecteur()->prepare($req);
29     }
30     try {
31         $query->bindParam(':arbitre', $arbitre, PDO::PARAM_INT);
32         $query->bindParam(':terrain', $field, PDO::PARAM_INT);
33         $query->bindParam(':jour', $day, PDO::PARAM_INT);
34         $query->bindParam(':heure', $time, PDO::PARAM_INT);
35         $query->bindParam(':sport', $sport, PDO::PARAM_INT);
36         $query->execute();
37         $query->fetch();
38     }
39     catch (Exception $e) {
40         error_log($e->getMessage());
41     }
42     $id = connecteur()->lastInsertId();
43
44     static $query2 = null;
45     if ($query2 == null) {
46         $req = "INSERT INTO `plays`(`idTeam`, `idGame`) VALUES (:team, :game)";
47         $query2 = connecteur()->prepare($req);
48     }
49
50     for ($team=0; $team < count($teams); $team++) {
51         try {
52             $query2->bindParam(':team', $teams[$team], PDO::PARAM_STR);
53             $query2->bindParam(':game', $id, PDO::PARAM_STR);
54             $query2->execute();
55             $query2->fetch();
56         }
57         catch (Exception $e) {

```

```

58         error_log($e->getMessage());
59     }
60 }
61 }
62
63 /**
64  * Récupère toutes infos du match
65  *
66  * @param string $id -> l'id du match
67  * @return array un tableau avec les infos du match
68  *     ['infos']
69  *         ['id'] -> l'id de l'équipe
70  *         ['played'] -> 1 si il as été joué, 0 si non
71  *         ['terrain'] -> le nom du terrain
72  *         ['arbitre'] -> le prenom de l'arbitre
73  *     ['teams']
74  *         [index]
75  *             ['numero'] -> le numéro de l'équipe
76  *             ['nom'] -> le nom de l'équipe
77  *             ['score'] -> le score de l'équipe
78  */
79 function GetAllMatchInfos($id){
80     $match = [];
81     $req = "SELECT `Games`.`id` as id, `Games`.`played` as played, `Field`.`nom` as
    • terrain, `Staff`.`prenom` as arbitre FROM `Games` INNER JOIN `Field` ON
    • `idTerrain` = `Field`.`id` INNER JOIN `Staff` ON `idArbitre` = `Staff`.`id`
    • WHERE `Games`.`id` = :id LIMIT 1";
82     $query = connecteur()->prepare($req);
83     $query->bindParam(':id', $id, PDO::PARAM_STR);
84     $query->execute();
85     $res = $query->fetch(PDO::FETCH_ASSOC);
86     $match['infos'] = $res;
87
88     $req = "SELECT `Teams`.`id` as numero, `Teams`.`nom` as nom, `score` FROM
    • `Teams`, `plays` WHERE `Teams`.`id` IN (SELECT `idTeam` FROM `plays` WHERE
    • `idGame` = :id) AND `plays`.`idGame` = :id AND `Teams`.`id` = `plays`.`idTeam`";
89     $query = connecteur()->prepare($req);
90     $query->bindParam(':id', $id, PDO::PARAM_STR);
91     $query->execute();
92     $res = $query->fetchAll(PDO::FETCH_ASSOC);
93     $match['teams'] = $res;
94
95     return $match;
96 }
97
98
99
100 /**
101  * Récupère toutes infos du match
102  *
103  * @param string $id -> l'id du match
104  * @return array un tableau avec le classement
105  *     [index]
106  *         ['id'] -> l'id de l'équipe
107  *         ['played'] -> 1 si il as été joué, 0 si non
108  *         ['terrain'] -> le nom du terrain
109  *         ['arbitre'] -> le prenom de l'arbitre
110  */
111 function GetAllMatchClassement($idGame, $idStaff, $idGame) {

```

```

111 function IsArbitreUrAdmin($idStaff, $idGame){
112     static $query = null;
113
114     if ($query == null) {
115         $req = "SELECT `Staff`.`prenom` FROM `Staff` WHERE `Staff`.`id` = (SELECT
        • `Games`.`idArbitre` FROM `Games` WHERE `Games`.`id` = :idGame LIMIT 1) AND
        • `Staff`.`id` = :idStaff";
116         $query = connecteur()->prepare($req);
117     }
118     try {
119         $query->bindParam(':idStaff', $idStaff, PDO::PARAM_STR);
120         $query->bindParam(':idGame', $idGame, PDO::PARAM_STR);
121         $query->execute();
122         $res = $query->fetch(PDO::FETCH_ASSOC);
123     }
124     catch (Exception $e) {
125         error_log($e->getMessage());
126         $res = false;
127     }
128
129     if ($res != false)
130         return true;
131     else{
132         return IsAdmin($idStaff);
133     }
134 }
135
136 /**
137  * Récupère le nombre de matchs qui ont lieu au jour indiqué
138  *
139  * @param string $id -> l'id du jour
140  * @return array un tableau avec le nombre de matchs
141  *             ['matches'] -> le nombre de matchs
142  */
143 function CountGamesOnDay($idDay){
144     static $query = null;
145
146     if ($query == null) {
147         $req = "SELECT COUNT(*) as matches FROM `Games` WHERE `idJour` = :idDay";
148         $query = connecteur()->prepare($req);
149     }
150     try {
151         $query->bindParam(':idDay', $idDay, PDO::PARAM_STR);
152         $query->execute();
153         $res = $query->fetch(PDO::FETCH_ASSOC);
154     }
155     catch (Exception $e) {
156         error_log($e->getMessage());
157         $res = false;
158     }
159
160     return $res;
161 }
162
163 /**
164  * Supprime un match
165  *
166  * @param string $id -> l'id du match
167  */

```

```

167 */
168 function DeleteGame($idGame){
169     static $query = null;
170
171     if ($query == null) {
172         $req = "DELETE FROM `Games` WHERE `id` = :id";
173         $query = connecteur()->prepare($req);
174     }
175     try {
176         $query->bindParam(':id', $idGame, PDO::PARAM_STR);
177         $query->execute();
178         $query->fetch();
179     }
180     catch (Exception $e) {
181         error_log($e->getMessage());
182     }
183 }
184
185 /**
186  * Récupère le match pour l'arbitre à une heure et un jour
187  *
188  * @param string $idStaff -> l'id de l'arbitre
189  * @param string $idDay -> l'id du jour
190  * @param string $idTime -> l'id du créneau horaire
191  * @return array un tableau avec le nombre de matchs
192  *         [index]
193  *         ['id'] -> l'id du match
194  *         ['terrain'] -> le nom du terrain
195  *         ['played'] -> si le match a été joué ou non
196  */
197 function GetMatchesForArbitreDayAndTime($idStaff, $idDay, $idTime){
198     static $query = null;
199
200     if ($query == null) {
201         $req = "SELECT `Games`.`id` as id, `Field`.`Nom` as terrain, `Games`.`played`
202             • as played FROM `Games` INNER JOIN `Field` ON `Field`.`id` =
203             • `Games`.`idTerrain` WHERE `Games`.`idArbitre` = :idStaff AND `Games`.`idJour`
204             • = :idDay AND `Games`.`idTime` = :idTime LIMIT 1";
205         $query = connecteur()->prepare($req);
206     }
207     try {
208         $query->bindParam(':idStaff', $idStaff, PDO::PARAM_STR);
209         $query->bindParam(':idDay', $idDay, PDO::PARAM_STR);
210         $query->bindParam(':idTime', $idTime, PDO::PARAM_STR);
211         $query->execute();
212         $res = $query->fetch(PDO::FETCH_ASSOC);
213     }
214     catch (Exception $e) {
215         error_log($e->getMessage());
216         $res = false;
217     }
218
219     return $res;
220 }

```