

```
1 <?php
2 /*****
3 AUTEUR      : Constantin Herrmann
4 LIEU        : CFPT Informatique Genève
5 DATE        : Avril 2020
6 TITRE PROJET: RESA
7 DESCRIPTION : Ce script contient toutes les fonctions pour effectuer des READ sur la table
8 'user'
9 VERSION     : 1.0
10 *****/
11 // On inclu le connecteur de la base de données
12 include '../..../pdo.php';
13
14 // On inclu le fichier que génère le mot de passe pour l'utilisateur
15 include '../password/index.php';
16
17 include '../..../vars.php';
18
19 // Get all users permet de récupérer tous les users de la base de données
20 function GetAllUsers(){
21     static $query = null;
22
23     if ($query == null) {
24         $req = 'SELECT `user`.`id`, `user`.`first_name`, `user`.`last_name`, `user`.`phone`,
25 `user`.`email`, `user`.`username` FROM `user`';
26         $query = database()->prepare($req);
27     }
28
29     try {
30         $query->execute();
31         $res = $query->fetchAll(PDO::FETCH_ASSOC);
32     } catch (Exception $e) {
33         error_log($e->getMessage());
34         $res = false;
35     }
36
37     return $res;
38 }
39
40 /*
41 * Récupère les users d'après leur permission
42 * Params :
43 * - $idPermission : L'id de la permission recherchée
44 */
45 function GetUsersByPermission($idPermission){
46     static $query = null;
47
48     if ($query == null) {
49         $req = 'SELECT `user`.`first_name`, `user`.`last_name`, `user`.`phone`, `user`.`email`,
50 `user`.`username` FROM `user` WHERE `user`.`id` IN (SELECT `is_in_as`.`idUser` FROM
51 `is_in_as` WHERE `is_in_as`.`idPermission` = '.$idPermission.')';
52         $query = database()->prepare($req);
53     }
54
55     try {
56         $query->execute();
57         $res = $query->fetchAll(PDO::FETCH_ASSOC);
58     } catch (Exception $e) {
59         error_log($e->getMessage());
60         $res = false;
61     }
62
63     return $res;
64 }
65
66 /*
67 * Récupère les informations sur les permissions d'un user d'après son id
68 * Params :
69 * - $id : l'id de l'utilisateur recherché
70 */
71 function GetUserPermissionById($id){
72     static $query = null;
73
74     if ($query == null) {
75         $req = 'SELECT IFNULL(e.name, "-") establishment_name, IFNULL(p.name, "-") as
76 permission_name FROM user as u LEFT JOIN is_in_as as iia ON iia.idUser = u.id LEFT JOIN
77 establishment as e ON e.id = iia.idEtablissement LEFT JOIN permission as p ON p.id =
78 iia.idPermission WHERE u.id = '.$id;
79         $query = database()->prepare($req);
80     }
81 }
```

```

77
78     try {
79         $query->execute();
80         $res = $query->fetchAll(PDO::FETCH_ASSOC);
81     }
82     catch (Exception $e) {
83         error_log($e->getMessage());
84         $res = false;
85     }
86
87     return $res;
88 }
89
90 /*
91  * Récupère les données d'un utilisateur d'après son id
92  * Params :
93  *   - $id : l'id de l'utilisateur recherché
94  */
95 function GetUser($id){
96     static $query = null;
97
98     if ($query == null) {
99         $req = 'SELECT `id`, `first_name`, `last_name`, `phone`, `email`, `username` FROM `user`
100 WHERE `id` = '.$id;
101         $query = database()->prepare($req);
102     }
103
104     try {
105         $query->execute();
106         $res = $query->fetchAll(PDO::FETCH_ASSOC);
107     }
108     catch (Exception $e) {
109         error_log($e->getMessage());
110         $res = false;
111     }
112
113     return $res;
114 }
115
116 /*
117  * Récupère les données d'un utilisateur si celui-ci travail bien dans l'établissement en
118  * question
119  * Params :
120  *   - $username : le nom d'utilisateur
121  *   - $password : le mot de passe déjà hashé en sha256
122  *   - $idEstablishment : l'id de son établissement
123  */
124 function LoginEstablishment($username, $password, $idEstablishment){
125     global $key;
126     $pass = hash('sha256', hash('sha256', $key).$password);
127
128     static $query = null;
129
130     if ($query == null) {
131         $req = 'SELECT u.id as idUser, u.first_name as firstnameUser, u.last_name as
132 lastnameUser, u.phone as phoneUser, u.email as emailUser, p.name as namePermission, p.level
133 as levelPermission, IFNULL(e.id, "-") as idEstablishment, IFNULL(e.name, "-") as
134 nameEstablishment FROM `is_in_as` as iia INNER JOIN `permission` as p ON p.id =
135 iia.idPermission LEFT JOIN `establishment` as e ON e.id = iia.idEstablishment INNER JOIN
136 `user` as u ON u.id = iia.idUser WHERE iia.idUser IN (SELECT `id` FROM `user` WHERE
137 `username` = :u AND `password` = :p) AND (iia.idEstablishment = :e OR iia.idPermission = 1)';
138         $query = database()->prepare($req);
139     }
140
141     try {
142         $query->bindParam(":u", $username, PDO::PARAM_STR);
143         $query->bindParam(":p", $pass, PDO::PARAM_STR);
144         $query->bindParam(":e", $idEstablishment, PDO::PARAM_STR);
145         $query->execute();
146         $res = $query->fetch(PDO::FETCH_ASSOC);
147     }
148     catch (Exception $e) {
149         error_log($e->getMessage());
150         $res = false;
151     }
152
153     if($res == null || $res == false){
154         return false;
155     }else{
156         return $res;
157     }
158 }
159
160 }
161
162 }

```

```

152 /*
153 * Récupère les données d'un utilisateur d'après son id
154 * Params :
155 *   - $email : l'email du client
156 *   - $password : le mot de passe déjà hashé en sha256
157 */
158 function Login($email, $password){
159     global $key;
160     $pass = hash('sha256', hash('sha256', $key).$password);
161
162     static $query = null;
163
164     if ($query == null) {
165         $req = 'SELECT `id`, `first_name`, `last_name`, `phone`, `email` FROM `user` WHERE
`email` = :e AND `password` = :p';
166         $query = database()->prepare($req);
167     }
168
169     try {
170         $query->bindParam(":e", $email, PDO::PARAM_STR);
171         $query->bindParam(":p", $pass, PDO::PARAM_STR);
172         $query->execute();
173         $res = $query->fetch(PDO::FETCH_ASSOC);
174     }
175     catch (Exception $e) {
176         error_log($e->getMessage());
177         $res = false;
178     }
179
180     if($res == null || $res == false){
181         return false;
182     }else{
183         return $res;
184     }
185 }
186
187
188 function LoginEtablissementManager($n, $e, $p){
189     global $key;
190     $pass = hash('sha256', hash('sha256', $key).$p);
191
192     static $query = null;
193
194     if ($query == null) {
195         $req = 'SELECT e.id FROM establishment as e INNER JOIN is_in_as as iia ON
iia.idEtablissement = e.id WHERE e.name = :n AND iia.idUser IN (SELECT u.id FROM user AS u
WHERE email = :e AND password = :p) AND iia.idPermission = 2';
196         $query = database()->prepare($req);
197     }
198
199     try {
200         $query->bindParam(":e", $e, PDO::PARAM_STR);
201         $query->bindParam(":p", $pass, PDO::PARAM_STR);
202         $query->bindParam(":n", $n, PDO::PARAM_STR);
203         $query->execute();
204         $res = $query->fetch(PDO::FETCH_ASSOC);
205     }
206     catch (Exception $e) {
207         error_log($e->getMessage());
208         $res = false;
209     }
210
211     if($res == null || $res == false){
212         return false;
213     }else{
214         return $res;
215     }
216 }
217
218 /*
219 * Vérifie si il y à déjà un compte avec le mail
220 * Params :
221 *   - $email : l'email à vérifier
222 */
223 function CheckMailExists($email){
224     static $query = null;
225
226     if ($query == null) {
227         $req = 'SELECT `id` FROM `user` WHERE `email` = :email';
228         $query = database()->prepare($req);
229     }
230
231     try {

```

```

232     $query->bindParam(":email", $email, PDO::PARAM_STR);
233     $query->execute();
234     $res = $query->fetch(PDO::FETCH_ASSOC);
235 }
236 catch (Exception $e) {
237     error_log($e->getMessage());
238     $res = false;
239 }
240
241 if($res == null || $res == false){
242     return false;
243 }else{
244     return true;
245 }
246 }
247
248
249 function IsAllowedToBeHere($idUser, $idEtab){
250     static $query = null;
251
252     if ($query == null) {
253         $req = 'SELECT p.id as permission FROM `is_in_as` as iia INNER JOIN permission AS p ON
p.id = iia.idPermission WHERE (iia.idEtablissement = '.$idEtab.' AND iia.idUser = '.$idUser.')
OR (iia.idPermission = 1 AND iia.idUser = '.$idUser.')';
254         $query = database()->prepare($req);
255     }
256
257     try {
258         $query->execute();
259         $res = $query->fetch(PDO::FETCH_ASSOC);
260     }
261     catch (Exception $e) {
262         error_log($e->getMessage());
263         $res = false;
264     }
265
266     if($res == null || $res == false){
267         return false;
268     }else{
269         return true;
270     }
271 }
272
273
274
275 if(isset($_GET['byPermission']))){
276     $idPermission = $_GET['byPermission'];
277     // Vérification qu'il s'agit bien d'un int
278     if(is_numeric($idPermission)){
279         echo json_encode(GetUsersByPermission($idPermission));
280     }else{
281         echo json_encode("Valeur non integer");
282     }
283 }
284 else if(isset($_GET['permissions']) && isset($_GET['id'])){
285     $id = $_GET['id'];
286     // Vérification qu'il s'agit bien d'un int
287     if(is_numeric($id)){
288         echo json_encode(GetUserPermissionById($id));
289     }else{
290         echo json_encode("Valeur non integer");
291     }
292 }
293 else if(isset($_GET['user']) && isset($_GET['id'])){
294     $id = $_GET['id'];
295     // Vérification qu'il s'agit bien d'un int
296     if(is_numeric($id)){
297         echo json_encode(GetUser($id));
298     }else{
299         echo json_encode("Valeur non integer");
300     }
301 }
302 else if(isset($_GET['login_e']) && isset($_GET['username']) && isset($_GET['password'])){
303     $username = $_GET['username'];
304     // $password = hash('sha256', $_GET['password']);
305     $password = $_GET['password'];
306
307     $user = LoginEtablissement($username, $password, 1);
308     if($user != false){
309         echo json_encode($user);
310     }else{
311         echo json_encode("Utilisateur ou mot de passe incorrect");
312     }

```

```
313 }
314 else if(isset($_GET['login']) && isset($_GET['email']) && isset($_GET['password']))){
315     $email = $_GET['email'];
316     //$password = hash('sha256', $_GET['password']);
317     $password = $_GET['password'];
318
319     $user = Login($email, $password);
320     echo json_encode($user);
321 }
322 else if(isset($_GET['n']) && isset($_GET['e']) && isset($_GET['p']))){
323     echo json_encode(LoginEtablissementManager($_GET['n'], $_GET['e'], $_GET['p']));
324 }
325 else if(isset($_GET['id']))){
326     $id = $_GET['id'];
327     // Vérification qu'il s'agit bien d'un int
328     if(is_numeric($id)){
329         $user = GetUser($id);
330         $permissions = GetUserPermissionById($id);
331
332         $user['permissions'] = $permissions;
333         echo json_encode($user);
334     }else{
335         echo json_encode("Valeur non integer");
336     }
337 }
338 else if(isset($_GET['checkmail']) && isset($_GET['email']))){
339     echo json_encode(CheckMailExists($_GET['email']));
340 }
341 else if(isset($_GET['u']) && isset($_GET['e']))){
342     echo json_encode(IsAllowedToBeHere($_GET['u'], $_GET['e']));
343 }
344 else{
345     echo json_encode(GetAllUsers());
346 }
```