

```
1  <?php
2
3  /*
4   * SADEK Serena
5   * Juin 2017
6   * TripTracker
7   * UserDbRelation.php
8   *
9   * Cette page contient toute les fonction interagissant avec la base de donnée,
10  * en ce qui concerne la table des utilisateur
11  * Elle est utilisée pour la connexion et l'inscription
12  */
13
14  session_start();
15
16  require_once '../connection.php';
17
18  if (isset($_POST["register"])) {
19      $result = [];
20      $username = filter_input(INPUT_POST, "userName", FILTER_SANITIZE_STRING);
21      $pwd = filter_input(INPUT_POST, "password", FILTER_SANITIZE_STRING);
22
23      if (userExist($username)) {
24          $result["ok"] = false;
25          $result["message"] = "Le nom d'utilisateur que vous avez choisi existe déjà.";
26      } else {
27          try {
28              $_SESSION["idUser"] = registerUser($username, $pwd);
29              $result["ok"] = true;
30              $result["message"] = "Vous avez été inscrit avec succès.";
31          } catch (Exception $exc) {
32              $result["ok"] = false;
33              $result["message"] = $exc->getTraceAsString();
34          }
35      }
36      //var_dump($result);
37      echo json_encode($result);
38  }
39
40  if (isset($_POST["login"])) {
41      $username = filter_input(INPUT_POST, "userName", FILTER_SANITIZE_STRING);
42      $pwd = filter_input(INPUT_POST, "password", FILTER_SANITIZE_STRING);
43
44      $result = connectUser($username, $pwd);
45      if (is_numeric($result)) {
46          $_SESSION["idUser"] = $result;
47      }
48      echo json_encode($result);
49  }
50
51  function userExist($username) {
52      $co = getConnection();
53      $req = $co->prepare("SELECT * FROM user WHERE `userName` LIKE :userName");
54      $req->bindParam(":userName", $username, PDO::PARAM_STR);
55      $req->execute();
56      $result = $req->fetchAll(PDO::FETCH_ASSOC);
57      if (count($result) > 0) {
58          return TRUE;
59      } else {
60          return FALSE;
61      }
62  }
63
64  /**
65   *
66   * @param string $username : Nom d'utilisateur
67   * @param string $pwd : Mot de passe non crypté
68   */
69  function registerUser($username, $pwd) {
```

```
70     $crypt = sha1($pwd);
71     $co = getConnection();
72     $req = $co->prepare("INSERT INTO user(`userName`, `userPwd`) values (:userName,
73     :password)");
74     $req->bindParam(":userName", $username, PDO::PARAM_STR);
75     $req->bindParam(":password", $crypt, PDO::PARAM_STR);
76     $req->execute();
77     return $co->lastInsertId();
78 }
79 /**
80  * Vérifie les informations que l'utilisateur a entré lors de sa tentative de
81  * connexion
82  * @param string $username : Nom d'utilisateur
83  * @param string $pwd : Mot de passe non crypté
84  * @return int idUser si les informations sont correctes
85  * @return TRUE si l'utilisateur existe mais que le mot de passe est incorrecte
86  * @return FALSE si l'utilisateur n'existe pas
87  */
88 function connectUser($username, $pwd) {
89     $co = getConnection();
90     $req = $co->prepare("SELECT * FROM user WHERE `userName` LIKE :userName");
91     $req->bindParam(":userName", $username, PDO::PARAM_STR);
92     $req->execute();
93     $result = $req->fetchAll(PDO::FETCH_ASSOC);
94     if (count($result) > 0) {
95         if ($result[0]["userPwd"] === sha1($pwd)) {
96             return $result[0]["idUser"];
97         } else {
98             return TRUE;
99         }
100     } else {
101         return FALSE;
102     }
103 }
```