

```
1  <?php
2
3  /*
4   * SADEK Serena
5   * Juin 2017
6   * TripTracker
7   * navigationData.php
8   *
9   * Cette page regroupe toute les fonctionnalités retrouvant les informations des
10  * voyages contenu dans la base. Elle est l'aboutissement des call AJAX contenu
11  * dans le fichier JavaScript tripNavigation
12  */
13
14  if (!isset($_SESSION)) { //Le fichier étant lié à d'autre fichier utilisant une session
15      session_start();
16  }
17
18  //Connection PDO nécessaire
19  require_once '../connection.php';
20
21  /**
22   * Protocole retournant le nombre de pages de 5 voyage dont l'utilisateur a
23   * besoin pour afficher tout les siens
24   */
25  if (isset($_POST["getPages"])) {
26      try {
27          $pages = getPageNumber();
28          echo $pages;
29      } catch (Exception $exc) {
30          echo $exc->getTraceAsString();
31      }
32  }
33
34  /**
35   * Protocole retournant les voyages ainsi que les information primaires de leurs
36   * étapes. Le numéro de la page demandé est passé en paramètre dans $_POST
37   */
38  if (isset($_POST["getPageContent"])) {
39      try {
40          $pages = getPageNumber();
41          $pageNo = filter_input(INPUT_POST, "pageNo", FILTER_SANITIZE_NUMBER_INT);
42
43          //Empêcher la page d'être hors des limites
44          if ($pageNo > $pages) {
45              $pageNo = $pages;
46          }
47          if ($pageNo < 1) {
48              $pageNo = 1;
49          }
50
51          //Récupération des voyages de la page
52          $content = getPageContent($pageNo);
53
54          //Récupération des étapes des voyages de la page
55          for ($index = 0; $index < count($content); $index++) {
56              $content[$index]["pathConstructor"] =
57                  getPath($content[$index]["pathObject"]);
58              $content[$index]["waypoints"] =
59                  getTripContent($content[$index]["idTrip"]);
60          }
61
62          echo json_encode($content);
63      } catch (Exception $exc) {
64          echo $exc->getTraceAsString();
65      }
66  }
67
68  /**
69   * Protocole visant à récupérer l'intégralité des informations d'une étape,
```

```
68     * média compris.
69     */
70     if (isset($_POST["getWpDetails"])) {
71         try {
72             $wpId = filter_input(INPUT_POST, "wpId", FILTER_SANITIZE_NUMBER_INT);
73
74             $dbResult = getWpDetails($wpId);
75
76             if (count($dbResult) > 0) {
77                 //Récupération des détails
78                 $details = getWpDetails($wpId)[0];
79
80                 //Le format de la date de la base de donnée est aaaa-mm-jj
81                 //Le format des input date est jj/mm/aaaa
82                 //On le fait dont passer de l'un à l'autre
83                 $details["wpDate"] = implode("/", array_reverse(explode("-",
84                     $details["wpDate"])));
85
86                 //Récupération des média
87                 $details["media"] = getMediaOfWp($wpId);
88
89                 echo json_encode($details);
90             } else {
91                 echo 'noResult';
92             }
93         } catch (Exception $exc) {
94             echo $exc->getTraceAsString();
95         }
96     }
97 }
98
99 /**
100  * Protocole visant à charger toute les informations d'un voyage, du titre du
101  * voyage aux media.
102  */
103 if (isset($_POST["loadTripModif"])) {
104     try {
105         $data = array();
106         //Récupération du voyage
107         $data["trip"] = getTrip($_POST["tripId"]);
108
109         if (count($data["trip"]) > 0) {
110             //Récupération des étapes
111             $data["waypoints"] = getWps($_POST["tripId"]);
112
113             //Récupération des étapes
114             for ($index1 = 0; $index1 < count($data["waypoints"]); $index1++) {
115                 $media = getMediaOfWp($data["waypoints"][$index1]["idWaypoint"]);
116                 $mediaTitle = array();
117
118                 if (isset($_POST["getPreviewConfig"])) {
119                     $mediaConstruct = array();
120                     $_SESSION["picOnDelete"] = array();
121                 }
122
123                 //Récupération du nom des média
124                 for ($index2 = 0; $index2 < count($media); $index2++) {
125                     $filename = "usersRessources/image/" .
126                         $media[$index2]["mediaName"];
127                     array_push($mediaTitle, $filename);
128
129                     //Génération d'un constructeur pour la prévisualisation de
130                     //l'input type file
131                     if (isset($_POST["getPreviewConfig"])) {
132                         $construct = array();
133                         $construct["type"] = "image";
134                         $construct["caption"] = "";
```

```

134         $construct["url"] = './AJAX/PictureInsertModif.php';
135         $construct["key"] = $media[$index2]["idMedia"];
136         array_push($mediaConstruct, $construct);
137     }
138 }
139 //Formatage de la date de aaaa-mm-jj à jj/mm/aaaa
140 $data["waypoints"][$index1]["wpDate"] = implode("/",
array_reverse(explode("-", $data["waypoints"][$index1]["wpDate"])));

141
142 $data["waypoints"][$index1]["media"] = $mediaTitle;
143 if (isset($_POST["getPreviewConfig"])) {
144     $data["waypoints"][$index1]["mediaConstruct"] = $mediaConstruct;
145 }
146 }
147
148     echo json_encode($data);
149 } else{
150     echo 'noResult';
151 }
152 } catch (Exception $exc) {
153     echo $exc->getTraceAsString();
154 }
155 }
156
157 /**
158  * Calcule le nombre de pages de cinq voyages dont l'utilisateur a besoin pour
159  * contenir tout ses voyages
160  * @return int : Nombre de pages
161  */
162 function getPageNumber() {
163     $co = getConnection();
164     $req = $co->prepare("SELECT count(`idTrip`) FROM `trip` where `idUser`=:idUser");
165     $req->bindParam(":idUser", $_SESSION["idUser"], PDO::PARAM_STR);
166     $req->execute();
167     $result = $req->fetch();
168     if ($result !== 0) {
169         if ($result["count(`idTrip`)"] % 5 == 0) { //La dernière page est remplie
170             return ($result["count(`idTrip`)"] / 5);
171         } else { //La dernière page n'est pas remplie
172             return ($result["count(`idTrip`)"] / 5) + 1;
173         }
174     } else {
175         return 0;
176     }
177 }
178
179 /**
180  * Retourne les cinq voyage contenu dans la page donnée
181  * @param int $page : index de la page
182  * @return array : détail des voyages
183  */
184 function getPageContent($page) {
185     $offset = ($page - 1) * 5;
186     $co = getConnection();
187     $req = $co->prepare("SELECT `idTrip`, `tpTitle`, `pathObject` FROM `trip` WHERE
`idUser`=:idUser LIMIT 5 OFFSET :pageStart");
188     $req->bindParam(":idUser", $_SESSION["idUser"], PDO::PARAM_INT);
189     $req->bindParam(":pageStart", $offset, PDO::PARAM_INT);
190     $req->execute();
191     $result = $req->fetchAll(PDO::FETCH_ASSOC);
192     return $result;
193 }
194
195 /**
196  * Retourne les informations minimales (titre et position) des étapes contenu
197  * dans le voyage donné
198  * @param type $idTrip
199  * @return array : détails des voyages
200  */

```

```

201 function getTripContent($idTrip) {
202     $co = getConnection();
203     $req = $co->prepare("SELECT `idWaypoint`, `wpTitle`, `lat`, `lng` FROM
204         `waypoint` where `idTrip`= :idTrip");
205     $req->bindParam(":idTrip", $idTrip, PDO::PARAM_INT);
206     $req->execute();
207     $result = $req->fetchAll(PDO::FETCH_ASSOC);
208     return $result;
209 }
210 /**
211  * Retourne les informations complètes de l'étape donnée
212  * @param type $idWp
213  * @return type
214  */
215 function getWpDetails($idWp) {
216     $co = getConnection();
217     $req = $co->prepare("SELECT `wpDate`, `wpComment`, `wpTitle`, `address`,
218         `user`.idUser FROM `waypoint`, trip, `user` where waypoint.idTrip = trip.idTrip
219         and trip.idUser = `user`.idUser and `idWaypoint`= :idWP and `user`.idUser =
220         :idUser");
221     $req->bindParam(":idWP", $idWp, PDO::PARAM_INT);
222     $req->bindParam(":idUser", $_SESSION["idUser"], PDO::PARAM_INT);
223     $req->execute();
224     $result = $req->fetchAll(PDO::FETCH_ASSOC);
225     return $result;
226 }
227 /**
228  * Retourne les informations des média associés à l'étape donnée
229  * @param type $idWp
230  * @return type
231  */
232 function getMediaOfWp($idWp) {
233     $co = getConnection();
234     $req = $co->prepare("SELECT `idMedia`, `mediaName` FROM `media` where
235         `idWaypoint`= :idWp");
236     $req->bindParam(":idWp", $idWp, PDO::PARAM_INT);
237     $req->execute();
238     $result = $req->fetchAll(PDO::FETCH_ASSOC);
239     return $result;
240 }
241 /**
242  * Retourne les informations du tracé contenu dans le fichier donné
243  * @param type $pathFileName
244  * @return type
245  */
246 function getPath($pathFileName) {
247     $fileFullName = "../usersRessources/path/" . $pathFileName;
248     $stream = fopen($fileFullName, 'r');
249     $content = fread($stream, filesize($fileFullName));
250     fclose($stream);
251     return $content;
252 }
253 /**
254  * Retourne toute les informations du voyage donné
255  * @param type $idTrip
256  * @return type
257  */
258 function getTrip($idTrip) {
259     $co = getConnection();
260     $req = $co->prepare("SELECT * FROM trip WHERE idTrip = :idTrip AND idUser =
261         :idUser");
262     $req->bindParam(":idTrip", $idTrip, PDO::PARAM_INT);
263     $req->bindParam(":idUser", $_SESSION["idUser"], PDO::PARAM_INT);
264     $req->execute();

```

```
264     $response = $req->fetch(PDO::FETCH_ASSOC);
265     return $response;
266 }
267
268 /**
269  * Retourne toute les informations des étapes du voyage donné
270  * @param type $idTrip
271  * @return type
272  */
273 function getWps($idTrip) {
274     $co = getConnection();
275     $req = $co->prepare("SELECT waypoint.idWaypoint, waypoint.`idTrip`,
        waypoint.wpDate, waypoint.wpComment, waypoint.wpTitle, waypoint.lat,
        waypoint.lng, waypoint.address FROM `waypoint`, `trip` WHERE
        `waypoint`.`idTrip`= :idTrip AND waypoint.idTrip = `trip`.idTrip AND
        `trip`.idUser = :idUser");
276     $req->bindParam(":idTrip", $idTrip, PDO::PARAM_INT);
277     $req->bindParam(":idUser", $_SESSION["idUser"], PDO::PARAM_INT);
278     $req->execute();
279     $response = $req->fetchAll(PDO::FETCH_ASSOC);
280     return $response;
281 }
282
```