

```
1  /*
2  * SADEK Serena
3  * Juin 2017
4  * TripTracker
5  * tripNaviagtion.js
6  *
7  * Les fonctions de cette page concernent la nagigation parmis les voyages,
8  * de l'affichage au chargement des données
9  */
10
11 //Voyage actif lors de l'aperçu
12 var ActivePanelId = null;
13
14 //
15 var NavPaths = [];
16 var NavMarkers = [];
17
18 $(document).ready(function () {
19     //Chargement des lien de pagination
20     generatePageLinks();
21     //Chargement de la première page
22     loadPage(1);
23
24     $("#BackToNav").click(function() {
25         closeLeft(false);
26         openRight();
27     });
28
29     /*
30     * Évènement click sur un lien (numéro de page), généré automatiquement
31     */
32     $("body").on('click', ".noPage", function () {
33         var id = event.target.id;
34         var noPage = id.slice(8, id.length);
35         if (!isNaN(noPage)) {
36             loadPage(noPage);
37             generatePageLinks();
38         }
39     });
40
41     /*
42     * Évènement à l'ouverture des panneaux dynamiques de navigation
43     */
44     $("body").on('show.bs.collapse', '.collapse', function () {
45         if (!creating) {
46             if (typeof $(event.target).parent().attr("id") !== "undefined") {
47                 var id = $(event.target).parent().attr("id");
48                 var ref = id.slice(6, id.length);
49                 if (!isNaN(ref) && ref >= 0 && ref < 5) {
50                     selectTab(ref);
51                 }
52             }
53
54             if (coll) { //Si l'évènement n'est pas lui-même appelé par un évènement
collapse
55                 if (typeof ref !== "undefined") {
56                     ActivePanelId = ref;
57                 }
58                 coll = false;
59                 var id = "#" + (this.id);
60                 $('#navTrips .collapse').not(id).collapse("hide");
61                 coll = true;
62             }
63         }
64     });
65
66     /*
67     * Évènement à la fermeture des panneaux dynamiques de navigation
68     */
69     $("body").on('hide.bs.collapse', '.collapse', function () {
70         if (!creating) {
```

```
71         ActivePanelId = null;
72         showAllTrips();
73         panOnAllTrips();
74     }
75 });
76
77 /**
78  * Évènement lors de la sélection d'un lien désignant une étape
79  */
80 $("body").on("click", ".waypointLink", function () {
81     var id = event.target.id;
82     var ref = id.slice(6, id.length);
83     if (!isNaN(ref)) {
84         LoadDetails(ref);
85     }
86 });
87
88 /**
89  * Évènement click des icône de stylo présents à droite des panneau de navigation
90  */
91 $("body").on("click", ".EditTrip", function () {
92     var id = event.target.id;
93     var ref = id.slice(4, id.length);
94     if (!isNaN(ref)) {
95         if (ref >= 0 && ref < 5) {
96             if ((typeof NavPaths[ref]) !== "undefined") {
97                 OpenModif((NavPaths[ref].id));
98             }
99         }
100     }
101 });
102
103 /**
104  * Évènement click des icône de boubelle présents à droite des panneaux de
105  * navigation
106  */
107 $("body").on("click", ".DeleteTrip", function () {
108     var id = event.target.id;
109     var ref = id.slice(6, id.length);
110     if (!isNaN(ref)) {
111         if (ref >= 0 && ref < 5) {
112             if (typeof NavPaths[ref] !== "undefined") {
113                 var tripId = NavPaths[ref].id;
114                 var confirmation = confirm("Êtes-vous sûr de vouloir supprimer
115                 ce voyage ?");
116                 if (confirmation) {
117                     DeleteTrip(tripId);
118                 }
119             }
120         }
121     }
122 });
123
124 /**
125  * Charge et génère le contenu d'une nouvelle page
126  * @param {type} idPage
127  * @returns {undefined}
128  */
129 function loadPage(idPage) {
130     unsetPageDisplay();
131     $.ajax({
132         type: 'post',
133         url: './AJAX/navigationData.php',
134         data: {getPageContent: true, pageNo: idPage},
135         success: function (response) {
136             var data = JSON.parse(response);
137             generateTripPanels(data);
138             drawNavPath(data);
139             drawMarkers(data);
140             panOnAllTrips();
141         }
142     });
143 }
```

```

140     }
141     });
142 }
143 /**
144  * Charge et génère les liens menant aux différentes pages
145  * @returns {undefined}
146  */
147 function generatePageLinks() {
148     $("#pageNos").html("");
149     $.ajax({
150         type: 'post',
151         url: './AJAX/navigationData.php',
152         data: {getPages: true},
153         success: function (response) {
154             for (i = 1; i <= Number(response); i++) {
155                 $("#pageNos").append('<button class="btn btn-link noPage" id="goToPage' + i + '">' + i + '</button>');
156             }
157         }
158     });
159 }
160
161 /**
162  * Génère les panneaux de navigation avec un code de couleur, à partir des
163  * informations passées en paramètre
164  * @param {type} pageContent
165  * @returns {undefined}
166  */
167 function generateTripPanels(pageContent) {
168     //Gris, vert, bleu, jaune, rouge
169     var panelClass = ["panel-default", "panel-success", "panel-info",
170         "panel-warning", "panel-danger"];
171
172     $("#TripPanels").html("");
173     var count = 0;
174     pageContent.forEach(function (trip) {
175         var panel = '<div class="panel ' + panelClass[count] + ' navPanelTrip'
176             id="paneTrip' + count + '">\n\
177                 <div class="panel-heading" role="tab" id="headingTrip' +
178                 count + '">\n\
179                 <h4 class="panel-title" id="Parent' + count + '">\n\
180                 <a role="button" data-toggle="collapse"
181                 href="#collapseTrip' + count + '"
182                 aria-expanded="true" aria-controls="collapseTrip' +
183                 count + '" class="trigger collapsed">\n\
184                 ' + trip.tpTitle
185
186                 + '</a>\n\
187
188                 <button type="button" id="Edit' + count + '"
189                 class="close pull-right EditTrip" aria-label="Edit"
190                 ' + count + '"><span id="SEdi' + count + '"
191                 class="glyphicon glyphicon-pencil"
192                 aria-hidden="true"></span></button>\n\
193                 <button type="button" id="Delete' + count + '"
194                 class="close pull-right DeleteTrip"
195                 aria-label="Delete" ' + count + '"><span
196                 id="SDelet' + count + '" class="glyphicon
197                 glyphicon-trash"
198                 aria-hidden="true"></span></button>\n\
199                 </h4></div>\n\
200                 </div>\n\
201                 <div id="collapseTrip' + count + '" class="panel-collapse
202                 collapse navPanel" role="tabpanel"
203                 aria-labelledby="headingTrip' + count + '">\n\
204                 <div class="panel-body">';
205
206         trip.waypoints.forEach(function (wp) {
207             panel += '<button id="wpLink' + wp.idWaypoint + '"
208                 class="list-group-item col-lg-12 text-left waypoitLink">\n\
209                 <i class="fa fa-globe"></i> ' + wp.wpTitle + '\n\
210                 </button>';
211         });
212     });

```

```

191
192     panel += '</div></div></div>';
193     $("#TripPanels").append(panel);
194     $("#navTrips .collapse").collapse({toggle: false});
195     count++;
196 });
197 }
198
199 /*
200  * Génère les tracés des informations passées en paramètres, avec un code de couleur
201  */
202 function drawNavPath(pageFullData) {
203     //Gris, vert, bleu, jaune, rouge
204     var color = ["#333333", "#3c763d", "#31708f", "#8a6d3b", "#a94442"];
205     var inc = 0;
206     pageFullData.forEach(function (trip) {
207         var pathConstructor = JSON.parse(trip.pathConstructor);
208         var Thiscolor = color[inc];
209         NavPaths[inc] = new google.maps.Polyline(pathConstructor[0]);
210         NavPaths[inc].setMap(map);
211         NavPaths[inc].ref = inc;
212         NavPaths[inc].id = trip.idTrip;
213         NavPaths[inc].setOptions({strokeWeight: 7, strokeColor: Thiscolor});
214         createPathClickEvent(inc);
215         inc++;
216     });
217 }
218
219 /**
220  * Rend le tracé d'index donné clickable
221  * @param {type} inc
222  * @returns {undefined}
223  */
224 function createPathClickEvent(inc) {
225     google.maps.event.addListener(NavPaths[inc], 'click', function () {
226         selectTrip(inc);
227     });
228 }
229
230 /**
231  * Crée des marqueurs cachés pour chaque étape des informations données
232  * @param {type} pageFullData
233  * @returns {undefined}
234  */
235 function drawMarkers(pageFullData) {
236     var countA = 0;
237     pageFullData.forEach(function (trip) {
238         var countB = 0;
239         NavMarkers[countA] = [];
240         trip.waypoints.forEach(function (wp) {
241             var position = new google.maps.LatLng(wp.lat, wp.lng);
242             NavMarkers[countA][countB] = new google.maps.Marker({position: position});
243             NavMarkers[countA][countB].ref = wp.idWaypoint;
244             createMarkerClickEvent(wp.idWaypoint, countA, countB);
245             countB++;
246         });
247         countA++;
248     });
249 }
250
251 /**
252  * Rend les marqueur de l'étape donnée, du voyage donné, clickable.
253  * @param {type} id
254  * @param {type} positionA : Index du voyage
255  * @param {type} positionB : Index de l'étape
256  * @returns {undefined}
257  */
258 function createMarkerClickEvent(id, positionA, positionB) {
259     google.maps.event.addListener(NavMarkers[positionA][positionB], 'click',
260         function () {
261             LoadDetails(id);

```

```

261         panOnMarker(NavMarkers[positionA][positionB]);
262     });
263 }
264
265 /**
266  * Réinitialise l'affichage de la navigation
267  * @returns {undefined}
268  */
269 function unsetPageDisplay() {
270     NavMarkers.forEach(function (markerGroup) {
271         markerGroup.forEach(function (marker) {
272             marker.setMap(null);
273         });
274     });
275     NavPaths.forEach(function (path) {
276         path.setMap(null);
277     });
278     NavMarkers = [];
279     NavPaths = [];
280     $("#TripPanels").html("");
281     $("#pageNos").html("");
282 }
283
284 /**
285  * Charge le contenu détaillé de l'étape dans le panneau de droite. Génère
286  * également le carousel si le voyage contient une image
287  * @param {type} tripId
288  * @returns {undefined}
289  */
290 function LoadDetails(tripId) {
291     $.ajax({
292         type: 'post',
293         url: './AJAX/navigationData.php',
294         data: {getWpDetails: true, wpId: tripId},
295         success: function (response) {
296             if (response !== "noResult") {
297                 var currentZoom = map.getZoom();
298                 if (currentZoom >= 10) {
299                     map.setZoom(17);
300                 } else {
301                     map.setZoom(10);
302                 }
303
304                 closeRight(true);
305                 openLeft();
306                 findMarker(tripId);
307                 var result = JSON.parse(response);
308                 $("#wpTitle").html(result.wpTitle);
309                 $("#wpDate").html(result.wpDate);
310                 $("#wpComment").html(result.wpComment);
311                 $("#wpAddress").html(result.address);
312
313                 $("#carouselSection .carousel-indicators").html("");
314                 $("#carouselSection .carousel-inner").html("");
315
316                 if (result.media.length == 0) {
317                     $("#carouselSection").hide();
318                 } else {
319                     $("#carouselSection").show();
320
321                     $("#carouselSection").html(
322                         '<div id="carouselWP" class="carousel slide"
323                         data-ride="carousel">\n\
324                         <!-- Indicators -->\n\
325                         <ol class="carousel-indicators">\n\
326                         </ol>\n\
327                         <!-- Wrapper for slides -->\n\
328                         <div class="carousel-inner">\n\
329                         \n\
330                         </div>\n\
331                         <!-- Left and right controls -->\n\

```

```

331     <a class="left carousel-control" href="#carouselWP"
332         data-slide="prev">\n\
333         <span class="glyphicon
334             glyphicon-chevron-left"></span>\n\
335         <span class="sr-only">Previous</span>\n\
336     </a>\n\
337     <a class="right carousel-control" href="#carouselWP"
338         data-slide="next">\n\
339         <span class="glyphicon
340             glyphicon-chevron-right"></span>\n\
341         <span class="sr-only">Next</span>\n\
342     </a>\n\
343 </div>'
344 );
345
346 var inc = 0;
347 result.media.forEach(function (media) {
348     if (inc == 0) {
349         $("#carouselSection .carousel-indicators").html('<li
350             data-target="#carouselWP" data-slide-to="' + inc + "'
351             class="active"></li>');
352         $("#carouselSection .carousel-inner").html('<div
353             class="item active"></div>');
355     } else {
356         $("#carouselSection .carousel-indicators").append('<li
357             data-target="#carouselWP" data-slide-to="' + inc + "'
358             ></li>');
359         $("#carouselSection .carousel-inner").append('<div
360             class="item"></div>');
362     }
363     inc++;
364 });
365 }
366
367 /**
368  * Met le focus le voyage à l'index donné lorsque c'est l'évènement click du tracé
369  * qui le déclenche
370  * @param {type} tripPosition
371  * @returns {undefined}
372  */
373 function selectTrip(tripPosition) {
374     panOnTrip(tripPosition);
375     ShowOnlyTrip(tripPosition);
376     $("#collapseTrip" + tripPosition).collapse("show");
377 }
378
379 /**
380  * Met le focus le voyage à l'index donné lorsque c'est l'évènement click de
381  * l'onglet qui le déclenche
382  * @param {type} tripPosition
383  * @returns {undefined}
384  */
385 function selectTab(tripPosition) {
386     panOnTrip(tripPosition);
387     ShowOnlyTrip(tripPosition);
388 }
389
390 /**
391  * Règle le zoom et le centre de la map de telle sorte à ce qu'elle contienne tout
392  * juste tout les marqueurs de la map
393  * @param {type} tripPosition
394  * @returns {undefined}
395  */
396 function panOnTrip(tripPosition) {
397     var bounds = new google.maps.LatLngBounds();

```

```
389
390     NavMarkers[tripPosition].forEach(function (marker) {
391         bounds.extend(marker.getPosition());
392     });
393     map.fitBounds(bounds);
394
395     var inc = 0;
396
397     NavMarkers.forEach(function (markerGroup) {
398         markerGroup.forEach(function (marker) {
399             if (inc !== Number(tripPosition)) {
400                 marker.setMap(null);
401             } else {
402                 marker.setMap(map);
403             }
404
405             });
406         inc++;
407     });
408 }
409
410 /**
411  * Centre la map sur le marque donné
412  * @param {type} marker
413  * @returns {undefined}
414  */
415 function panOnMarker(marker) {
416     map.panTo(marker.position);
417 }
418
419 /**
420  * Cache les tracés de tout les voyage sauf celui du voyage à l'index donné
421  * @param {type} tripPosition
422  * @returns {undefined}
423  */
424 function ShowOnlyTrip(tripPosition) {
425     var inc = 0;
426     NavPaths.forEach(function (path) {
427         if (inc == Number(tripPosition)) {
428
429             path.setMap(map);
430         } else {
431             path.setMap(null);
432         }
433         inc++;
434     });
435 }
436 /**
437  * Montre tout les tracés et cache tout les marqueurs
438  * @returns {undefined}
439  */
440 function showAllTrips() {
441     NavMarkers.forEach(function (markerGroup) {
442         markerGroup.forEach(function (marker) {
443             marker.setMap(null);
444         });
445     });
446
447     NavPaths.forEach(function (path) {
448         path.setMap(map);
449     });
450 }
451
452 /**
453  * Règle le zoom et le centre de la map de telle sorte à ce qu'elle puisse tout
454  * juste contenir tout les voyage de la page
455  * @param {type} tripId
456  * @returns {undefined}
457  */
458 function panOnAllTrips(tripId) {
459     var bounds = new google.maps.LatLngBounds();
```

```
460     NavMarkers.forEach(function (markerGroup) {
461         markerGroup.forEach(function (marker) {
462             bounds.extend(marker.getPosition());
463         });
464     });
465     map.fitBounds(bounds);
466 }
467
468 /**
469  * Cherche parmi tout les marqueur de tout les voyages de la page celui qui
470  * correspond à l'id sélectionné.
471  * @param {type} markerId
472  * @returns {undefined}
473  */
474 function findMarker(markerId) {
475     NavMarkers.forEach(function (markerGroup) {
476         markerGroup.forEach(function (marker) {
477             if (Number(markerId) == Number(marker.ref)) {
478                 panOnMarker(marker);
479             }
480         });
481     });
482 }
483
484 /**
485  * Supprime le voyage de l'id sélectionné.
486  * @param {type} tripId
487  * @returns {undefined}
488  */
489 function DeleteTrip(tripId) {
490     $.ajax({
491         type: 'post',
492         url: './AJAX/DataInsertModif.php',
493         data: {deleteTrip: true, idToDelete: tripId},
494         success: function (response) {
495             generatePageLinks();
496             loadPage(1);
497         }
498     });
499 }
```