

```

1  /*
2  * SADEK Serena
3  * Juin 2017
4  * TripTracker
5  * insertInterface.js
6  *
7  * Toute les fonctions reliées à l'onglet permettant l'insertion de voyage
8  */
9
10 var InsertionElements = [];
11
12 //Compteur d'onglet générés
13 var count = 0;
14
15 //Indique si on est en train de fermer manuellement un panel
16 var coll = true;
17
18 //Valeur numérique ou nulle indiquant l'onglet qui est ouvert
19 var focus = null;
20
21 $(document).ready(function () {
22
23
24
25     /**
26     * Evenènement click du bouton "Ajouter une étape"
27     * Genère un onglet pré-fabriqué, mais unique grâce à la variable count
28     */
29     $("#AddState").click(function () {
30         //Crée un marqueur non défini
31         creationMarkers.push("none");
32
33         //Crée une route non défini
34         if (count > 0) { //Si ce n'est pas le premier marqueur
35             creationRoutes.push("none");
36         }
37
38         var nowDate = new Date();
39         var date = nowDate.toLocaleDateString();
40         var length = count;
41
42         //Incrémente le compteur
43         count++;
44
45         //Code du pannel
46         $("#InsertionContent").append(
47             '<div class="panel panel-default" id="insert' + length + '">'
48             + '<div class="panel-heading" role="tab" id="headInsert' + length + ' ">'
49             + '<h4 class="panel-title">'
50             + '<a role="button" data-toggle="collapse" id="TripName' + length + ' ">'
51             + ' href="#content' + length + ' " aria-expanded="true" '
52             + ' aria-controls="content' + length + ' " class="trigger '
53             + ' collapsed">[Insérer un titre]</a>'
54             + '<button type="button" id="Close' + length + ' " class="close '
55             + ' pull-right closeInsertionPanel" aria-label="Close" ' + length + ' '
56             + '>&times;</button></h4></div>'
57             + '<div id="content' + length + ' " class="panel-collapse collapse" '
58             + ' role="tabpanel" + length + ' " aria-labelledby="headInsert' + length + ' '
59             + '>'
60             + '<div class="panel-body">'
61             + '<ul class="list-group">'
62             + '<li class="list-group-item">' //\n\
63             + '<!-- Titre à modifier -->\n\
64             + ' <div class="form-group">\n\
65             + ' <i class="glyphicon glyphicon-flag col-lg-1"></i>\n\
66             + ' <div id="Ptitle' + length + ' " class="input-group '
67             + ' col-lg-11">\n\
68             + ' <input type="text" id="title' + length + ' " '
69             + ' class="form-control titleControl" placeholder="Titre '
70             + ' de L\'étape">\n\

```

```

61         </div>\n\
62     </div>\n\
63     <!-- Recherche d\'adresse -->\n\
64     <div class="form-group">\n\
65         <i class="glyphicon glyphicon-map-marker col-lg-1"></i>\n\
66         <div id="Paddress" + length + " class="input-group
67             col-lg-11">\n\
68             <input type="text" id="adress" + length + "
69                 class="form-control" placeholder="Adresse de
70                 l\'étape">\n\
71             <div class="input-group-btn">\n\
72                 <button class="btn btn-default searchLoc"
73                     id="secfind" + length + "><i id="findLoc" +
74                         length + " class="glyphicon glyphicon-search
75                         col-lg-1"></i></button>\n\
76                 <button class="btn btn-default searchMyLoc"
77                     id="secMyLc" + length + "><i id="MyLoc" +
78                         length + " class="glyphicon
79                         glyphicon-screenshot col-lg-1"></i></button>\n\
80             </div>\n\
81         </div>\n\
82     </div>\n\
83     <!-- Selection de la date -->\n\
84     <div class="form-group">\n\
85         <i class="glyphicon glyphicon-time col-lg-1"></i>\n\
86         <div id="Pdate" + length + " class="input-group
87             col-lg-11">\n\
88             <input id="date" + length + " type="text" value="
89                 + date + " readonly class="form_date col-lg-12
90                 form-control">\n\
91             </div>\n\
92         </div>'
93     + '</li>'
94     + '<li class="list-group-item">'
95     + '<input id="picSelect" + length + " accept="image/*"
96         name="picSelect" + length + "[]" type="file" multiple
97         class="file-loading">' //Ajout des images
98     + '</li>'
99     + '</ul>'
100     + '<div id="Pcomment" + length + " class="form-group"><textarea
101         class="form-control col-lg-10" offset="" rows="15" id="comment" +
102         length + "></textarea></div>'
103     + '</div></div></div>'
104     );
105     var name = '#content' + length;
106     //Initialise la variable définissant l'ouverture du panel
107     $(name).collapse({toggle: false});
108
109     //Initialise le DateTimePicker avec la date du jour
110     $("#date" + length).datetimepicker({format: 'dd/mm/yyyy', startView: 'month',
111         minView: 'month',
112         autoclose: true});
113
114     //Initialise le fileINput
115     $("#picSelect" + length).fileinput({
116         previewFileType: "image",
117         browseClass: "btn btn-default custom",
118         browseLabel: "",
119         browseIcon: "<i class=\"glyphicon glyphicon-picture\"></i> ",
120         removeClass: "btn btn-default customRem",
121         removeLabel: "",
122         removeIcon: "<i class=\"glyphicon glyphicon-trash\"></i> ",
123         uploadUrl: './AJAX/PictureInsertModif.php',
124         uploadExtraData: {
125             idState: length,
126             insert: true
127         }, // server upload action ?idState=' + length+"&insert=true"
128         uploadAsync: false,
129         showUpload: false,
130         maxFileSize: 200.0,
131         maxFileCount: 10,

```

```

116         msgFilesTooMany: "Vous ne pouvez pas insérer plus de {n} images par
117         étape",
118         msgSizeTooLarge: "L'image {name} dépasse la limite de taille autorisée
119         ({maxSize})",
120         allowedFileTypes: ["image"]
121     });
122
123     $('#insert' + length + " .collapse").collapse("show");
124
125     //Défini l'évènement de suppression commandé par la croix
126     $('#QuitCreation').click(function () {
127         var confirmation = confirm("Voulez-vous vraiment quitter l'interface de
128         création ?");
129         if (confirmation) {
130             closeInsertInterface();
131         }
132     });
133
134     /**
135     * Evènement déclenché par la croix d'un panel dynamique
136     * Commande la suppression de ce panel
137     */
138     $('body').on('click', '.closeInsertionPanel', function () {
139         var confirmation = confirm("Voulez-vous vraiment supprimer cette étape ?");
140         if (confirmation) {
141             //Récupération de l'id de la cible
142             var id = event.target.id;
143             //Récupération du numéro du panel
144             var position = id.substring(5, id.length);
145
146             if (!isNaN(position)) {
147                 if (position >= 0 && position < count) {
148                     //Génération de l'id du panel
149                     var DeleteItemID = "#insert" + position;
150                     //Suppression du panel
151                     $(DeleteItemID).remove();
152                     if (position == focus) {
153                         //Le focus ne doit pas rester sur un panel supprimé
154                         focus = null;
155                     }
156                     //Masquer le marqueur si il est défini
157                     if (creationMarkers[position] !== "none" &&
158                         creationMarkers[position] !== null) {
159                         creationMarkers[position].setMap(null);
160                     }
161                     //Supprimer le marqueur
162                     creationMarkers[position] = null;
163
164                     //Supprimer les routes liées à cette étape
165                     suppressRoadsOfDot(position);
166
167                     //Centre la map sur le nouveau tracé
168                     PanOnCreationTrip();
169                 }
170             }
171         }
172     });
173
174     /**
175     * Événement généré par l'évènement click du bouton loupe, à droite du champ
176     * d'adresse.
177     * Cherche la position correspondant à l'adresse, et si elle existe, génère
178     * un marqueur, corrige l'adresse et trace les éventuelles route qui le
179     * relie aux autres marqueurs.
180     */
181     $('body').on('click', '.searchLoc', function () {
182         var id = event.target.id;
183         var ref = id.substring(7, id.length);
184         if (!isNaN(ref)) {
185             if (ref >= 0 && ref < count) {

```

```

183         var source = "#adress" + ref;
184         var address = $(source).val();
185         if (address.length > 0) {
186             getPositionFromAdresse(address);
187         }
188     }
189 }
190 });
191
192 /**
193  * Événement généré par l'évènement click du bouton cible, à droite du champ
194  * d'adresse.
195  * Géolocalise l'utilisateur et génère le marqueur, l'adresse et les
196  * éventuelles routes liés à cette localisation.
197  */
198 $('body').on('click', '.searchMyLoc', function () {
199     geoLocation();
200 });
201
202 /**
203  * Empêche l'utilisateur d'avoir plus d'un onglet ouvert à la fois
204  */
205 $('body').on('show.bs.collapse', '#InsertionContent .collapse', function () {
206     /*Vu que l'évènement se déclenche lui-même, il a le potentiel de
207     * Déclencher une boucle infini. La variable coll est donc nécessaire
208     * pour que l'évènement agisse différemment s'il est appelé par une
209     * action de l'utilisateur ou s'il est appelé par lui même*/
210
211     if (creating) {
212         if (coll) { //L'évènement est-il appelé par une action ?
213             //Le prochain ne le sera pas
214             coll = false;
215             var id = "#" + (this.id);
216             //Fermer tout les onglets qui ne sont pas la target.
217             $('#InsertionContent .collapse').not(id).collapse("hide");
218
219             //Les prochain evènement seront une action
220             coll = true;
221
222             var ref = id.substring(8, id.length);
223
224             if (!isNaN(ref)) {
225                 //Changement de focus
226                 focus = ref;
227             }
228         }
229     }
230 });
231
232 /**
233  * met le focus à null si une autre fenêtre n'est pas ouverte à la volée.
234  */
235 $('body').on('hide.bs.collapse', '#InsertionContent .collapse', function () {
236     if (coll) {
237         focus = null;
238     }
239 });
240
241 /**
242  * Met à jour l'entête du panneau par rapport à son champ de titre
243  */
244 $('body').on('change', '.titleControl', function () {
245     var id = event.target.id;
246     var ref = id.substring(5, id.length);
247     var target = "#TripName" + ref;
248     var value = event.target.value;
249     if (ref >= 0 && ref < count) {
250         if ($.trim(value).length > 1 && !isNaN(ref)) {
251             $(target).html(value);
252         } else {
253             //Si le champ titre est vide, on remet l'entête par défaut

```

```
254         $(target).html("[Insérer un titre]");
255     }
256 }
257 }
258 );
259 });
260
261
262 /**
263  * Ferme correctement l'interface de création, en supprimant son contenu et en
264  * réinitialisant les variables qui lui sont liés.
265  * @returns {undefined}
266  */
267 function closeInsertInterface() {
268     //Réinitialisation du focus
269     focus = null;
270     var countA = 0;
271
272     //Suppression et annulation de l'affichage de chaque marqueur
273     creationMarkers.forEach(function (element) {
274         if (typeof element == "object" && element != null) {
275             element.setMap(null);
276         }
277         if (countA == creationMarkers.length - 1) {
278             creationMarkers = [];
279         }
280         countA++;
281     });
282     var countB = 0;
283     //Suppression et annulation de l'affichage de chaque route
284     creationRoutes.forEach(function (element) {
285         if (typeof element == "object" && element != null) {
286             element.display.setMap(null);
287         }
288         if (countB == creationRoutes.length - 1) {
289             creationRoutes = [];
290         }
291         countB++;
292     });
293
294     //Vides les champs qui ne sont pas supprimés
295     $('#titleTrip').val('');
296     $('#InsertionErrorSection').html('');
297     $('#InsertionContent').html('');
298     $('#InsertionErrorSection').removeClass('alert');
299     $('#InsertionErrorSection').removeClass('alert-success');
300
301     //Fermeture de l'interface de création
302     closeAdd(false);
303     creating = false;
304     editing = null;
305
306     //Rechargement des numéro de page
307     generatePageLinks();
308
309     //Rechargement de la première page
310     loadPage(1);
311
312     //Réouverture de la navigation
313     openRight();
314 }
315
316 function PanOnCreationTrip() {
317     var bounds = new google.maps.LatLngBounds();
318     creationMarkers.forEach(function (marker) {
319         if (typeof marker == "object" && marker != null) {
320             bounds.extend(marker.getPosition());
321         }
322     });
323     map.fitBounds(bounds);
324 }
```