

```

1  /*
2  * SADEK Serena
3  * Juin 2017
4  * TripTracker
5  * inertsDataTratments.js
6  *
7  * Les fonctions de cette page concernent le traitement, le formatage et l'envoi
8  * des informations rentrées par l'utilisateur dans le formulaire d'ajout de
9  * voyage.
10 * */
11
12 $(document).ready(function () {
13
14     /**
15     * Evènement déclenché par le click du bouton "Enregistrer le voyage"
16     */
17     $("#SubmitNewTrip").click(function () {
18         getInformations();
19     });
20
21     /**
22     * Récupère les informations rentrée par l'utilisateur
23     * dans le formulaire d'ajout de voyage
24     * @returns {undefined}
25     */
26     function getInformations() {
27
28         //Si le voyage est composé d'au moins deux étapes
29         if (count >= 1) {
30             var name = "#insert";
31             var content = [];
32
33             //Récupération du titre du voyage
34             var title = $('#titleTrip').val();
35
36             //Récupération du contenu des champs des panel, sous forme de tableau
37             //associatif
38             for (var flag = 0; flag < Number(count); flag++) {
39                 if ($("#insert" + flag).length !== 0) {
40                     content.push({});
41                     content[content.length - 1].ref = flag; //Numéro du panel
42                     content[content.length - 1].title = ($("#title" + flag).val());
43                     //Titre
44                     content[content.length - 1].date = ($("#date" + flag).val()); //Date
45                     content[content.length - 1].comment = ($("#comment" +
46                     flag).val()); //Commentaire
47                     if (typeof creationMarkers[flag] == "object") { //Si le marqueur
48                     est défini
49                         content[content.length - 1].lat =
50                         creationMarkers[flag].position.lat(); //Latitude
51                         content[content.length - 1].lng =
52                         creationMarkers[flag].position.lng(); //Longitude
53                         content[content.length - 1].address =
54                         creationMarkers[flag].address; //Adresse
55                         if (typeof creationMarkers[flag].id !== "undefined") {
56                             content[content.length - 1].id = creationMarkers[flag].id;
57                         }
58                     }
59                 }
60             }
61             if (count == flag + 1) {
62                 //Verification des informations
63                 checkInformations(content, title);
64                 break;
65             }
66         }
67     }
68
69     /**
70     * Vérifie la validité des informations contenues dans le paramètre
71     * @param {type} content

```

```

65     * @returns {undefined}
66     */
67     function checkInformations(content, title) {
68         //Expression régulière pour jj/mm/aaaa
69         var regExpDate = new
        RegExp(/^( ((0[1-9]|[12][0-9]|30)[-\/]? (0[13-9]|1[012])|31[-\/]? (0[13578]|1[02]
        )| (0[1-9]|1[0-9]|2[0-8]) [-\/]? 02) [-\/]? (0[0-9]{4}|29[-\/]? 02[-\/]? ([0-9]{2}([24
        68][048]| [02468][48])| [13579][26])| ([13579][26]| [02468][048]| 0[0-9]| 1[0-6]) 00)
        )$/);

70
71         var i = 0;
72
73         //Présence d'aucune erreur
74         var ok = true;
75
76         ///      VERIFICATION DU TITRE DU VOYAGE
77
78         if ($.trim(title).length == 0) { //Le titre n'est pas vide
79             ok = false;
80             $("#titleSection").addClass("has-error");
81         } else {
82             $("#titleSection").removeClass("has-error");
83         }
84
85         content.forEach(function (element) {
86             var hasError = false;
87
88             ///      VERIFICATION DE LA DATE
89
90             if (!regExpDate.test(element.date)) {
91                 //Si le format de la date n'est pas correcte;
92                 $("#Pdate" + element.ref).addClass("has-error");
93                 hasError = true;
94             } else if (i > 0 && regExpDate.test(content[i - 1].date)) {
95                 //Si la date de l'étape est antérieure à celle de la précédente
96                 var dateComponent1 = content[i - 1].date.split('/', 3);
97                 var dateComponent2 = element.date.split('/', 3);
98
99                 var date1 = new Date(dateComponent1[2], dateComponent1[1],
100                 dateComponent1[0]);
101                 var date2 = new Date(dateComponent2[2], dateComponent2[1],
102                 dateComponent2[0]);
103
104                 if (date1 > date2) {
105                     $("#Pdate" + element.ref).addClass("has-error");
106                     hasError = true;
107                 } else {
108                     $("#Pdate" + element.ref).removeClass("has-error");
109                 }
110             } else {
111                 $("#Pdate" + element.ref).removeClass("has-error");
112             }
113
114             ///      VERIFICATION DU COMMENTAIRE
115
116             if ($.trim(element.comment).length == 0) { //Le commentaire n'est pas vide
117                 hasError = true;
118                 $("#Pcomment" + element.ref).addClass("has-error");
119             } else {
120                 $("#Pcomment" + element.ref).removeClass("has-error");
121             }
122             content[i].comment = element.comment.replace(/</g, "&lt;").replace(/>/g,
123             "&gt;");
124
125             ///      VERIFICATION DU TITRE DE L'ETAPE
126
127             if ($.trim(element.title).length == 0) { //Le titre n'est pas vide
128                 hasError = true;
129                 $("#Ptitle" + element.ref).addClass("has-error");
130             } else {
131                 $("#Ptitle" + element.ref).removeClass("has-error");
132             }
133         });
134     }
135 }

```

```

129         $("#Ptitle" + element.ref).removeClass("has-error");
130     }
131     content[i].title = element.title.replace(/</g, "&lt;").replace(/>/g,
132         "&gt;");
133     /// Vérification du type des images
134
135     var reg2 = new RegExp(/danger/);
136     var htmlContent = $($($("#picSelect" +
137         element.ref).parent().parent().parent().find(".file-caption-name").html(
138     ));
139     if (reg2.test(htmlContent)) {
140         hasError = true;
141     }
142
143     var regexp = new RegExp(/image*/);
144     var files = $("#picSelect" + element.ref).fileinput('getFileStack');
145     if (files.length > 10) {
146         hasError = true;
147     }
148     files.forEach(function (file) {
149         if (!regexp.test(file.type)) {
150             hasError = true;
151         }
152         if (Number(file.size) / 1024 > 600) {
153             hasError = true;
154         }
155     });
156
157     console.log()
158     //Met le panneau en rouge s'il possède au moins une erreur
159     if (hasError) {
160         $("#insert" + element.ref).removeClass("panel-default");
161         $("#insert" + element.ref).addClass("panel-danger");
162         ok = false;
163     } else {
164         $("#insert" + element.ref).addClass("panel-default");
165         $("#insert" + element.ref).removeClass("panel-danger");
166     }
167     i++;
168 });
169 //Le chemin est-il complet
170 areAllPointSet(content, title, ok, 0);
171 }
172
173 /**
174  * Vérifie si un tracé a pu être trouvé entre tout les points
175  * @returns {undefined}
176  */
177 function areAllPointSet(content, title, ok, inc) {
178     /* Si un marqueur n'est pas défini, c'est dans le champs d'adresse du
179     * panel correspondant que l'erreur sera indiquée */
180
181     creationMarkers.forEach(function (element) {
182         if ($("#insert" + inc).length !== 0) {
183             if (element == "none") {
184                 $("#Padress" + inc).addClass("has-error");
185                 ok = false;
186             } else {
187                 $("#Padress" + inc).removeClass("has-error");
188             }
189         }
190     });
191     if (inc == creationMarkers.length - 1) {
192         if (ok) {
193             renderDirectionsPolylines(content, title);
194         } else {
195             $('#InsertionErrorSection').addClass('alert');
196             $('#InsertionErrorSection').addClass('alert-danger');
197             $('#InsertionErrorSection').html('<span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span> <span

```

```
class="sr-only">Error:</span>Toute vos informations ne sont pas
correctes.');
```

```

196         }
197     }
198     inc++;
199 });
200 }
201
202 /*
203  * Genère le constructeur d'une Polyline devant correspondre au tracé de
204  * l'utilisateur
205  */
206 function renderDirectionsPolylines(content, title) {
207     var points = [];
208     var inc = 0;
209     creationRoutes.forEach(function (response) {
210         if (typeof response == "object" && response !== null) {
211             if (points.length != 0) {
212                 if (typeof response.route == "string") {
213                     console.log(points);
214                     points = points.concat({path: response.display.getPath().b});
215                 } else {
216                     points =
217                         points.concat(response.route.routes[0].legs[0].steps);
218                 }
219                 if (inc == creationRoutes.length - 1) {
220                     serializePath(points, content, title);
221                 }
222             } else {
223                 if (typeof response.route == "string") {
224                     points = [{path: response.display.getPath().b}];
225                 } else {
226                     points = response.route.routes[0].legs[0].steps;
227                 }
228                 if (inc == creationRoutes.length - 1) {
229                     serializePath(points, content, title);
230                 }
231             }
232         } else {
233             if (inc == creationRoutes.length - 1) {
234                 serializePath(points, content, title);
235             }
236         }
237         inc++;
238     });
239 }
240
241 /**
242  * Genère le constructeur de la polyline sous forme de string
243  * @param {array} Polylines
244  * @param {array} content
245  * @param {string} title
246  * @returns {undefined}
247  */
248 function serializePath(Polylines, content, title) {
249     var PathString = "[";
250     PathString += "{\"map\": null, \"geodesic\": true, \"path\": [";
251     var countA = 0;
252     Polylines.forEach(function (polyline) {
253         if (countA !== 0) {
254             PathString += ",";
255         }
256
257         countA++;
258
259         var countB = 0;
260         var path = polyline.path;
261         path.forEach(function (point) {

```

```

264
265         if (countB !== 0) {
266             PathString += ",";
267         }
268
269         countB++;
270
271         PathString += "{\\"lat\\": " + point.lat() + ",\\"lng\\": " +
272             point.lng() + "}";
273     });
274
275     PathString += "]}";
276     PathString += "];";
277
278     SaveInformations(PathString, content, title);
279 }
280
281 /*
282  * Enregistre les informations dans la base de donnée et enregistre le chemin
283  * dans un fichier texte, stocké sur le serveur. Toute ses opérations se font
284  * sous forme de transaction : si une étape n'est pas effectuée correctement,
285  * rien ne sera enregistré ni sur le serveur, ni sur la base de donnée
286  */
287 function SaveInformations(path, content, title) {
288     var data;
289     var action;
290     var condition;
291     if (editing !== null) {
292         data = {path: path, content: JSON.stringify(content), title: title,
293             edit: true, tripId: editing};
294         action = "mis à jour";
295     } else {
296         data = {path: path, content: JSON.stringify(content), title: title,
297             insert: true};
298         action = "ajouté";
299     }
300     $.ajax({//On demande à la base de donnée de vérifier les informations de
301         l'utilisateur
302         type: 'post', //La methode poste empêche l'utilisateur d'accéder
303         lui-même au contenu de la base de donnée
304         url: './AJAX/DataInsertModif.php',
305         data: data,
306         success: function (response) {
307             try
308             {
309                 if (editing == null) {
310                     var wpIds = JSON.parse(response);
311                     condition = !isNaN(wpIds[0]);
312                 } else {
313                     condition = response == "OK";
314                 }
315
316                 if (condition) {
317                     var inc = 0;
318                     content.forEach(function (element) {
319                         var id = element.ref;
320                         $('#picSelect' + id).fileinput('upload');
321
322                     if (inc == content.length - 1) {
323                         $('#InsertionErrorSection').addClass('alert');
324
325                         $('#InsertionErrorSection').removeClass('alert-danger')
326                         );
327                         $('#InsertionErrorSection').addClass('alert-success');
328                         $('#InsertionErrorSection').html('<span
329                             class="glyphicon glyphicon-exclamation-sign"
330                             aria-hidden="true"></span> <span
331                             class="sr-only">Error:</span>Votre voyage a été ' +
332                             action);

```

```
324
325         setTimeout(function () {
326             closeInsertInterface();
327         }, 1500);
328     }
329     inc++;
330 });
331
332     } else {
333         $('#InsertionErrorSection').addClass('alert');
334         $('#InsertionErrorSection').addClass('alert-danger');
335         $('#InsertionErrorSection').html('<span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span> <span class="sr-only">Error:</span>' + response + '');
336     }
337 } catch (e)
338 {
339     $('#InsertionErrorSection').addClass('alert');
340     $('#InsertionErrorSection').addClass('alert-danger');
341     $('#InsertionErrorSection').html('<span class="glyphicon glyphicon-exclamation-sign" aria-hidden="true"></span> <span class="sr-only">Error:</span>' + response + '');
342
343     return;
344 }
345 }
346 });
347 }
348 });
349
350
351
```