

# TPI - Chronoview

---

Code Source

Zoé Cugni

19/06/2017

## 1. Table des matières

1.	Table des matières.....	1
2.	Index.php .....	2
3.	Login.php .....	6
4.	Js/timeline.js .....	7
5.	Js/lightbox.js .....	15
6.	Functions/dao.php.....	25
7.	Functions/dbCoConfig.php .....	30
8.	Functions/interlocutor/event.php .....	31
9.	Functions/interlocutor/getCategories.php .....	31
10.	Functions/interlocutor/getSession.php.....	32
11.	Functions/interlocutor/loginCheck.php.....	32
12.	Functions/interlocutor/rangeYear.php.....	33
13.	Functions/interlocutor/updateEvent.php .....	34
14.	Functions/interlocutor/updateCategory.php .....	35
15.	Functions/interlocutor/insertCategory.php .....	35
16.	Functions/interlocutor/insertEvent.php .....	36
17.	Functions/interlocutor/deleteCategory.php .....	37
18.	Functions/interlocutor/deleteEvent.php .....	37
19.	Css/style.scss .....	38

## 2. Index.php

```
C:\xampp\htdocs\TPI\index.php                                         lundi 19 juin 2017 16:13
1  <?php
2      session_start();
3  ?>
4
5  <!DOCTYPE html>
6  <html>
7      <head>
8          <meta charset="utf-8">
9          <link href="css/nouislider.min.css" rel="stylesheet">
10         <title>Chronoview</title>
11         <link rel="stylesheet" href="css/nouislider.css">
12         <link rel="stylesheet" href="css/style.css">
13         <link href="https://fonts.googleapis.com/css?family=PT+Sans+Caption" rel=
14             "stylesheet">
15         <script src="https://code.jquery.com/jquery-3.1.1.min.js" integrity=
16             "sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLlv2b8=" crossorigin="anonymous"
17             ></script>
18         <script src="//cdn.jsdelivr.net/jquery.color-animation/1/mainfile"></script>
19         <script src="js/wNumb.js"></script>
20         <script src="js/nouislider.js"></script>
21         <script src="js/timeline.js"></script>
22         <script src='js/lightbox.js'></script>
23     </head>
24     <body>
25         <div id='blackCloak'>
26             <div id='sidebarBlock'>
27                 <h2>Catégories &#x25BE;</h2>
28                 <ul id='listCat'>
29                     <!--Generated in javascript -->
30                 </ul>
31             </div>
32             <h1>Chronoview</h1>
33             <!-- hidden insert lightbox -->
34             <div id="addLightbox">
35                 <h1>Ajouter un événement</h1>
36                 <form action='functions/interlocutor/insertEvent.php' method=
37                     'post'>
38                     <div id='left'>
39                         <div id='leftInnerBlock'>
40                             <label for="insertTitle">Titre</label>
41                             <input type="text" name="insertTitle" id=
42                                 "insertTitle" required>
43
44                             <label for="insertDescription">Description</label>
45                             <textarea id='insertDescription' name=
46                                 'insertDescription' rows='20' required></textarea>
47                         </div>
48                     </div>
49
50                     <div id='right'>
51                         <div id='rightInnerBlock'>
52                             <div id='startDateBlock'>
53                                 <label for="insertStartingDay">Date de début
54                                 </label>
55                                 <input type="number" min='1' max='31' name=
56                                     "insertStartingDay" id="insertStartingDay">
57                                 <select id='insertStartingMonth'>
```

C:\xampp\htdocs\TPI\index.php

lundi 19 juin 2017 16:13

```

50          <option value='null'> </option>
51          <option value='Janvier'>Janvier</option>
52          <option value='Février'>Février</option>
53          <option value='Mars'>Mars</option>
54          <option value='Avril'>Avril</option>
55          <option value='Mai'>Mai</option>
56          <option value='Juin'>Juin</option>
57          <option value='Juillet'>Juillet</option>
58          <option value='Août'>Août</option>
59          <option value='Septembre'>Septembre</option>
60          <option value='Octobre'>Octobre</option>
61          <option value='Novembre'>Novembre</option>
62          <option value='Décembre'>Décembre</option>
63      </select>
64      <input type="number" name="insertStartingYear" id=
65          "insertStartingYear" onfocusout=
66          "minInsertEndingYear()" required>
67  </div>
68
69  <div id='endingDateBlock'>
70      <label for="insertEndingDay">Date de fin</label>
71      <input type="number" min='1' max='31' name=
72          "insertEndingDay" id="insertEndingDay">
73          <select id='insertEndingMonth'>
74              <option value='null'> </option>
75              <option value='Janvier'>Janvier</option>
76              <option value='Février'>Février</option>
77              <option value='Mars'>Mars</option>
78              <option value='Avril'>Avril</option>
79              <option value='Mai'>Mai</option>
80              <option value='Juin'>Juin</option>
81              <option value='Juillet'>Juillet</option>
82              <option value='Août'>Août</option>
83              <option value='Septembre'>Septembre
84              </option>
85              <option value='Octobre'>Octobre</option>
86              <option value='Novembre'>Novembre</option>
87              <option value='Décembre'>Décembre</option>
88          </select>
89          <input type="number" name="insertEndingYear" id=
90              "insertEndingYear" required>
91  </div>
92
93  <label for="insertImgFile">Image</label>
94  <input type="file" name="insertImgFile" id=
95      "insertImgFile">
96
97  <label for="insertCategories">Catégories :</label>
98  <select multiple name="insertCategories" id=
99      "insertCategories" required>
    <!--Generated in javascript -->

```

```
C:\xampp\htdocs\TPI\index.php                                         lundi 19 juin 2017 16:13
100      </div>
101
102      <!-- hidden update lightbox -->
103      <div id='lightboxOverlay'>
104          <div id='updateLightbox'>
105              <form>
106                  <input type='text' id='updateTitle'>
107                  <input type='submit' id='modify' value="Confirmer">
108                  <button type='button' id='cancelUpdate' onclick=
109                      'cancelUpdate()'>Annuler</button>
110
111      <!-- Dates -->
112      <div id='dateSection'>
113          <input type='number' id='updateStartingDay' min='1'
114              max='31'>
115          <select id='updateStartingMonth'>
116              <option value='null'> </option>
117              <option value='Janvier'>Janvier</option>
118              <option value='Février'>Février</option>
119              <option value='Mars'>Mars</option>
120              <option value='Avril'>Avril</option>
121              <option value='Mai'>Mai</option>
122              <option value='Juin'>Juin</option>
123              <option value='Juillet'>Juillet</option>
124              <option value='Août'>Août</option>
125              <option value='Septembre'>Septembre</option>
126              <option value='Octobre'>Octobre</option>
127              <option value='Novembre'>Novembre</option>
128              <option value='Décembre'>Décembre</option>
129
130          <p> - </p>
131
132          <input type='number' id='updateEndingDay' min='1'max=
133              '31'>
134          <select id='updateEndingMonth'>
135              <option value='null'> </option>
136              <option value='Janvier'>Janvier</option>
137              <option value='Février'>Février</option>
138              <option value='Mars'>Mars</option>
139              <option value='Avril'>Avril</option>
140              <option value='Mai'>Mai</option>
141              <option value='Juin'>Juin</option>
142              <option value='Juillet'>Juillet</option>
143              <option value='Août'>Août</option>
144              <option value='Septembre'>Septembre</option>
145              <option value='Octobre'>Octobre</option>
146              <option value='Novembre'>Novembre</option>
147              <option value='Décembre'>Décembre</option>
148          <input type='number' id='updateEndingYear' required>
149      </div>
150
151      <textarea id='updateDescription' required></textarea>
152      <div id='rightBlock'>
```

C:\xampp\htdocs\TPI\index.php lundi 19 juin 2017 16:13

```
153             <input type='file' id='fileInput'>
154             <label for='updateCategories'>Catégories :</label>
155             <select multiple name="updateCategories" id=
156                 "updateCategories" size='7' required>
157                 <!--Generated in javascript -->
158             </select>
159         </form>
160     </div>
161 </div>
162 <main>
163     <?php
164         if(isset($_SESSION['admin']) && $_SESSION['admin'] == true)
165             echo '';
167         ?>
168         <!-- generated in javascript -->
169     </main>
170     <div id="slider">
171         <!-- generated in javascript -->
172     </div>
173 </div>
174 </body>
175 </html>
```

### 3. Login.php

```
C:\xampp\htdocs\TP\login.php                                         lundi 19 juin 2017 16:14
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <link href="css/nouislider.min.css" rel="stylesheet">
6          <title>Chronoview</title>
7          <link rel="stylesheet" href="css/nouislider.css">
8          <link rel="stylesheet" href="css/style.css">
9      </head>
10     <body id='login'>
11         <div id='blackCloak'>
12             <?php
13                 session_start();
14                 $toDisplay = "<form method='post' action='functions/interlocutor/loginCheck.php'>";
15
16                 if(!isset($_SESSION['admin']) || $_SESSION['admin'] != true) {
17                     $toDisplay .= "<label for='mail'>Email :</label>";
18                     $toDisplay .= "<input type='email' id='mail' name='mail'>";
19                     $toDisplay .= "<label for='pw'>Mot de passe :</label>";
20                     $toDisplay .= "<input type='password' id='pw' name='pw'>";
21                     $toDisplay .= "<input type='submit' value='Se connecter'>";
22
23                 if(isset($_GET['error']) && $_GET['error'] == 'true')
24                     $toDisplay .= "<p id='error'>Vous avez entré de mauvais
25                     paramètres ou vous n'êtes pas un administrateur.</p>";
26                 } else {
27                     $toDisplay .= '<input type="submit" value="Se déconnecter">';
28                 }
29
30                 echo $toDisplay;
31             ?>
32         </form>
33     </div>
34 </body>
35 </html>
```

## 4. Js/timeline.js

```
C:\xampp\htdocs\TPI\js\timeline.js                                         lundi 19 juin 2017 16:09
1  /*
2   * Description : Contains all functions regarding the display of the timeline and
3   * the document.ready() function
4   * Author : Zoé Cugni
5   * Date : June 2017
6   * Context : school last project (TPI)
7   */
8
9  var checkedCatIdArray = [];
10 var eventList = {};//actually an object because of js syntax
11 var nbPixelPerYear = 0;
12 var sliderValues = [];
13
14 //Prepare the page : create the slider, add the categories list and bind functions
15 //to element events
16 $(document).ready(function(){
17     //ask for years data and create the slider with it
18     $.ajax({
19         url : './functions/interlocutor/rangeYear.php',
20         type : 'GET',
21         dataType : 'json',
22         async: false, //the rest of the page can't work without this so there's no
need to make it asynchrone
23         success: function(rangeData){
24             addSlider(rangeData);
25
26             //Add the "+" button if the admin is connected
27             $.ajax({
28                 url : './functions/interlocutor/getSession.php',
29                 dataType : 'text',
30                 async: false, // I prefer to have the page completely ready when the
user start using it
31                 success: function(isAdmin){
32                     if(isAdmin)
33                         $('main').append('');
34
35                     sliderValues = slider.noUiSlider.get();
36                     nbPixelPerYear = parseInt($('main').css("width")) / (sliderValues
[1] - sliderValues[0]); //width of line / year represented by
the line
37                 }
38             });
39
40             displayCategories();
41
42             //Bind function to different event of different event
43
44             $('main').on("mouseenter", ".dot", displayMiniInfo);
45             $('main').on("mouseleave", ".dot", removeMiniInfo);
46
47             $('#addLightbox form').submit(function(e){
48                 e.preventDefault();
49                 insertData();
50             });

```

```
C:\xampp\htdocs\TPI\js\timeline.js                                         lundi 19 juin 2017 16:09
51
52     $('#updateLightbox form').submit(function(e) {
53         e.preventDefault();
54         saveUpdate();
55     });
56
57     //Show the list of categories when the user hover its title
58     $("#sidebarBlock h2").on('mouseenter', function(){
59         $('#listCat').show();
60     });
61
62     //Hide the list of categories when the user leave the list
63     $("#listCat").on('mouseleave', function(){
64         $('#listCat').hide();
65         $.ajax({
66             url : './functions/interlocutor/getSession.php',
67             dataType : 'text',
68             async: false, // I prefer to have the page completely ready when the
69             user start using it
70
71             //Reload the data in case the admin started updating or creation a
72             //category but didn't finish
73             success: function(isAdmin){
74                 if(isAdmin) {
75                     displayCategories();
76                 }
77             });
78
79             //important' is a class that override the hover style when the user unselect a
80             //category
81             //i need it because the hover and select style are the same, so when the user
82             //clicked to unselect,
83             // it did, but with the other style we didn't directly see a difference
84             //The rest of the time i still want the hover to work normally
85             $('#listCat li').on('mouseleave', function(){
86                 $(this).removeClass('important');
87             })
88
89             //hide every lightbox (the three aren't shown at the same time,
90             //but this click event works for the three of them, so i didn't separate it in
91             //different function)
92             $('#lightboxOverlay').click(function(e){
93                 if(e.target === this) { //check that just the overlay (and not its children)
94                     has been clicked
95                     $('#lightbox').remove();
96                     $('#updateLightbox form')[0].reset();
97                     $('#addLightbox form')[0].reset();
98                     $('#updateLightbox').hide();
99                     $('#addLightbox').hide();
100                    $('#lightboxOverlay').hide();
101                }
102            });
103
104            //Transform the 'Nouvelle Catégorie' element into a form element enabling to
105            //write the name of a new category
```

```
C:\xampp\htdocs\TPI\js\timeline.js                                         lundi 19 juin 2017 16:09
101    $('#listCat').on('click', '#newCat', function(e){
102        if(e.target === this) { //check that just the li (and not its children) has
103            been clicked
104            $("#newCat").html('');
105            $("#newCat").append('<input type="text">');
106            $("#newCat").append("<img src='img/tick.png' alt='Valider'>");
107        }
108    });
109
110    //Inser the new category name
111    $('#listCat').on('click', '#newCat img', function(e){
112        if(e.target === this) { //check that just the img has been clicked
113            var catName = $("#newCat input").val();
114            $("#newCat").empty();
115            $("#newCat").html('Nouvelle catégorie');
116
117            $.ajax({
118                url : './functions/interlocutor/insertCategory.php',
119                type : 'POST',
120                dataType : 'html',
121                data : { name : catName},
122                async: false, // I prefer to have the page completely ready when the
123                user start using it
124                success: function(msg){
125                    displayCategories();
126                    alert(msg);
127                }
128            });
129
130    //Delete the category from which the delete symbol has been clicked
131    $('#listCat').on('click', '.supp', function(e){
132        if(e.target === this) { //check that just the img has been clicked
133            var catId = $(this).parent().attr('id').split('t')[1];
134
135            $.ajax({
136                url : './functions/interlocutor/deleteCategory.php',
137                type : 'POST',
138                dataType : 'html',
139                data : { id : catId },
140                async: false,
141                success: function(msg){
142                    displayCategories();
143                    alert(msg);
144                }
145            });
146        }
147    });
148
149    $('#listCat').on('click', 'li', function(e){
150        if(e.target === this) { //check that just the li (and not its children) has
151            been clicked
152            selectCategory($(this));
153        }
154    });

```

```
C:\xampp\htdocs\TPI\js\timeline.js                                         lundi 19 juin 2017 16:09
155      //replace the category name with an input enabling its edition
156      $('#listCat').on('click', '.edit', function(e) {
157          if(e.target === this) {
158              var catName = $(this).prev().text();
159              $(this).prev().remove();
160              $(this).parent().prepend("<input type='text' value='" + catName + "' id='editCat'>");
161          }
162      });
163
164      $('#listCat').on('mouseleave', '#editCat', modifyCat);
165  });
166
167  //Create the slider from a range of years
168  function addSlider(rangeData){
169      var slider = document.getElementById('slider');
170
171      // variable used to know where to place the two handles
172      var minYear = parseInt(rangeData[0].minYear);
173      var maxYear = parseInt(rangeData[0].maxYear);
174      var range = maxYear - minYear;
175      var halfOfSlider = minYear + range / 2;
176      var percentageOfRange = (range * 10) / 100;
177
178      noUiSlider.create(slider, {
179          range: { //value represented by the slider
180              'min': [minYear],
181              'max': [maxYear]
182          },
183          start: [halfOfSlider - percentageOfRange, halfOfSlider + percentageOfRange],
184          //position the two handles, at the half +/- 10% of the range
185          connect: true, //fill the area between the handles with a different color
186
187          //format the number above the handle to not show any decimals
188          // (wNumb being from an extern librairy)
189          tooltips: [ wNumb({ decimals: 0 }), wNumb({ decimals: 0 }) ],
190
191          behaviour: 'none' //just the normal drag of one handle, without any other
192          //possible behaviour
193      });
194
195      sliderValues = slider.noUiSlider.get();
196      nbPixelPerYear = parseInt($("#main").css("width")) / (sliderValues[1] -
197      sliderValues[0]); //width of line / year represented by the line
198
199      //detect when one of the handle has been moved and update the view if necessary
200      slider.noUiSlider.on('end', function() {
201          updateView();
202      });
203      updateView();
204      if(typeof checkedCatIdArray !== 'undefined' && checkedCatIdArray.length > 0) {
205          //if any category has been selected
206          $('#main').empty();
```

```
C:\xampp\htdocs\TPI\js\timeline.js                                         lundi 19 juin 2017 16:09
207      //add the "+" sign enabling the creation of events if the admin is connected
208      //(needed because of the .empty() )
209      $.ajax({
210          url : './functions/interlocutor/getSession.php',
211          dataType : 'text',
212          async: false, // I prefer to have the page completely ready when the
213          user start using it
214          success: function(isAdmin){
215              if(isAdmin)
216                  $('#main').append('');
218
219              sliderValues = slider.noUiSlider.get();
220              nbPixelPerYear = parseInt($("#main").css("width")) / (sliderValues[1]
221              - sliderValues[0]); //width of line / year represented by the line
222          }
223      });
224
225      //Display the timeline for each selected categories
226      $.each(checkedCatIdArray, function(){
227          getEvents(this);
228      });
229
230      //Modify a category
231      function modifyCat(){
232          //the id is cat1, with 1 being the id of the cat (and so different each time)
233          var catId = $(this).parent().attr('id').split('t')[1];
234          var newCatName = $(this).val();
235
236          $.ajax({
237              url : './functions/interlocutor/updateCategory.php',
238              type : 'POST',
239              dataType : 'html',
240              data : { newName : newCatName , id : catId},
241              async: false, // I prefer to have the page completely ready when the user
242              start using it
243              success: function(msg){
244                  displayCategories();
245                  alert(msg);
246              }
247          });
248
249      //Update with the 3 categories list (the one on the main page, the one in the update
250      //event form and the one in the insert event form)
251      function displayCategories(){
252          $.ajax({
253              url : './functions/interlocutor/getCategories.php',
254              type : 'GET',
255              dataType : 'json',
256              success: function(catData){
257                  //Clear all data
258                  $('#updateCategories').empty();
259                  $('#insertCategories').empty();
260                  $('#listCat').empty();

```

```
C:\xampp\htdocs\TPI\js\timeline.js                                         lundi 19 juin 2017 16:09
259
260          //add new data
261          $.each(catData, function(index, cat){
262              $('#updateCategories').append('<option id="option' + cat.idCategory +
263                  '" value="' + cat.idCategory + '">' + cat.name + '</option>');
264              $('#insertCategories').append('<option value="' + cat.idCategory +
265                  '">' + cat.name + '</option>');
266              $('#listCat').append('<li id="cat' + cat.idCategory + '"><p>' + cat.
267                  name + '</p></li>');
268          });
269
270          $.ajax({
271              url : './functions/interlocutor/getSession.php',
272              dataType : 'text',
273              async: false, // I prefer to have the page completely ready when the
274              user start using it
275
276              //Add the modify and delete symbols as well as the "Nouvelle
277              //Catégorie" element if the admin is connected
278              success: function(isAdmin){
279                  if(isAdmin) {
280                      $('#listCat li').append("<img src='img/edit.png'
281                         alt='Modifier la catégorie' class='edit'>");
282                      $('#listCat li').append("<img src='img/supp.png'
283                         alt='Supprimer la catégorie' class='supp'>");
284
285                      //The two img are in absolute position in order to have them
286                      //at the end of the line
287                      //So i need to find the right distance of the first one by
288                      //calculating the space taken by the second one
289                      var editRight = parseInt($(".supp").css('width')) + parseInt(
290                          $(".supp").css('right')) + 5;
291                      //Update the width of the li element in order to have their
292                      //border also goes under the symbols
293                      var newWidth = parseInt($("#listCat").css('width')) +
294                          parseInt($(".edit").css('width')) + editRight;
295                      $(".edit").css("right", editRight);
296                      $("#listCat li").css('width', newWidth);
297                      $("#listCat").prepend("<li id='newCat'>Nouvelle
298                          catégorie</li>");
299
300                  }
301              });
302          });
303      });
304  });
305
306  //Select a category and add its timeline to the page
307  function selectCategory(li){
308      //the id is cat1, with 1 being the id of the cat (and so different each time)
309      var idCat = $(li).attr('id').split('t')[1];
310
311      if (!$(li).hasClass('selected')){ //display new timeline
312          if(checkedCatIdArray.length < 8){ //check that there isn't too many cat
313              already selected
314              $(li).addClass('selected');
315              checkedCatIdArray.push(idCat)
316
317      }
318
319      
```

```
C:\xampp\htdocs\TPI\js\timeline.js                                         lundi 19 juin 2017 16:09
302         getEvents(idCat);
303     } else {
304         alert("Trop de catégories sélectionnées, désélectionner en une d'abord");
305     }
306
307 } else { //clear timeline of this category
308     $(li).removeClass('selected');
309
310     // 'important' is a class that override the hover style when the user
311     // unselect a category
312     // i need it because the hover and select style are the same, so when the
313     // user clicked to unselect,
314     // it did, but with the other style we didn't directly see a difference
315     // The rest of the time i still want the hover to work normally
316     $(li).addClass('important');
317     var index = checkedCatIdArray.indexOf(idCat);
318     checkedCatIdArray.splice(index, 1);
319     $('#timeline' + idCat).remove();
320 }
321
322 //Get the event for a certain period & category and call the function to display them
323 function getEvents(idCat){
324     $.ajax({
325         url : './functions/interlocutor/event.php',
326         type : 'GET',
327         dataType : 'json',
328         data : { idCategory: idCat, minYear: Math.trunc(sliderValues[0]), maxYear:
329             Math.trunc(sliderValues[1])},
330         async: false, //otherwise the line won't appear in the right order
331         success: function(eventData){
332             displayTimeline(eventData);
333         }
334     });
335
336 //display the new timeline with the received events
337 function displayTimeline(eventData){
338     //eventData has the id and name of the category at the beginning and then all
339     //the event object
340     var idCategory = eventData.splice(0,1);
341     var categoryName = eventData.splice(0,1);
342
343     //add the timeline
344     $('main').append('<div class="timeline" id="timeline' + idCategory + '"></div>');
345     $('#timeline' + idCategory).append('<p class="timelineName">' + categoryName +
346     '</p>');
347
348     //add the dots
349     $(eventData).each(function(){
350         //the id is used to retrieve this event, but i also need to retrieve this
351         //specific dot, which is why i use a unique class
352         var uniqueClass = 'dot' + this.idEvent + '_' + idCategory;
353
354         // -sliderValues[0] because we want the year difference between the start of
355         // the timeline (not the year 0) and the dot * nb pixel per 1 year
356         var dotLeftPos = (this.startingYear - sliderValues[0]) * nbPixelPerYear;

```

```
C:\xampp\htdocs\TPI\js\timeline.js                                         lundi 19 juin 2017 16:09
352
353     $("#" + idCategory).append('<div class="dot ' + uniqueClass + '"'
354     id="dot' + this.idEvent + '"></div>');
355     $("." + uniqueClass).css("left", dotLeftPos);
356
357     var lastDotLeftPos = parseInt($(".." + uniqueClass).prev().css("left"));
358
359     //if the new dot is on the last one (same position) -> put the new dot
360     //higher (vertically)
361     if(dotLeftPos >= lastDotLeftPos - 5 && dotLeftPos <= lastDotLeftPos + 5){
362         var lastDotBottomPos = parseFloat($(".." + uniqueClass).prev().css(
363             "bottom"));
364
365         // 2 * border of dot (= its height) + last dot bottom pos + small space
366         var dotBottomPos = parseFloat($(".dot").css("border-width")) * 2 + 2 +
367             lastDotBottomPos;
368         $("." + uniqueClass).css("bottom", dotBottomPos);
369     }
370
371     eventList['dot' + this.idEvent] = this;
372 });
373 }
```

## 5. Js/lightbox.js

```
C:\xampp\htdocs\TPI\js\lightbox.js                                         lundi 19 juin 2017 16:13
1  /*
2   * Description : Contains all functions regarding the three lightbox and the mini
3   * Info box
4   * Author : Zoé Cugni
5   * Date : June 2017
6   * Context : school last project (TPI)
7   */
8
9
10 //Display a lightbox with the content of the hovered event
11 function displayLightbox(eventData){
12     $('#lightboxOverlay').show();
13
14     //add the lightbox and its element
15     $('#lightboxOverlay').append('<div id="lightbox"></div>'); //I don't append it
16     //to the overlay so the opacity doesn't affect it
17
18     //displayLightbox is call by different functions, once with element and once
19     //directly with the id
20     if(typeof eventData.data !== 'undefined')
21         idEvent = eventData.data.idEvent;
22     else
23         idEvent = eventData;
24
25     $.ajax({
26         url : './functions/interlocutor/getSession.php',
27         dataType : 'text',
28         async: false, //otherwise the line won't re-appear in the right order
29         success: function(isAdmin){
30             if(isAdmin){
31                 $('#lightbox').append('<button type="button" id="supp"'
32                     + 'onclick="deleteEvent(' + idEvent.split('t')[1] +
33                     ')">Supprimer</button>');
34                 $('#lightbox').append('<button type="button" id="modify"'
35                     + 'onclick="startUpdate()">Modifier</button>');
36             }
37         }
38     });
39
40     var event = eventList[idEvent];
41     $('#lightbox').addClass('event' + event.idEvent);
42     $('#lightbox').append('<h1>' + event.title + '</h1>');
43
44     var dates = treatEventDate(event.startingDay, event.startingMonth, event.
45     startingYear, event.endingDay, event.endingMonth, event.endingYear)
46
47     //if the two dates are the same, only the first one will be shown
48     //otherwise the two will be
49     var dateString = '<h3>' + dates[0];
50
51     if(dates[0] != dates[1])
52         dateString += ' - ' + dates[1];
53
54     dateString += '</h3>';
55
56     //add the date to the lightbox aswell as other element
```

```
C:\xampp\htdocs\TPI\js\lightbox.js                                         lundi 19 juin 2017 16:13
51      $('#lightbox').append(dateString);
52
53      if(event.imgFileName != null)
54          $('#lightbox').append('');
55
56      $('#lightbox').append('<p>' + event.description + '</p>');
57  }
58
59 //Delete an event from its id
60 function deleteEvent(idEvent){
61     $.ajax({
62         url : './functions/interlocutor/deleteEvent.php',
63         type : 'POST',
64         dataType : 'html',
65         data : {id : idEvent},
66         async: false,
67         success: function(msg){
68             alert(msg);
69
70             $('#lightbox').hide();
71             $('#lightboxOverlay').hide();
72
73             //reload data so the change is taken into account
74             $('#main').empty();
75             $('#main').append('');
76
77             $.each(checkedCatIdArray, function(){
78                 getEvents(this);
79             });
80         }
81     });
82 }
83
84 //Change all the data of the lightbox with form input containing said data
85 //Add two buttons, 'Cancel' and 'Confirm' as well as a select multiple with the
86 //categories
87 function startUpdate(){
88     var boxClass = $('#lightbox').attr('class');
89     var event = eventList['dot' + boxClass.split('t')[1]];
90
91     $('#lightbox').remove();
92     $('#lightboxOverlay').show();
93     $('#updateLightbox').show();
94     $('#updateLightbox').addClass(boxClass);
95
96     // Title
97     $('#updateTitle').val(event.title);
98     $('#updateTitle').attr('size', $('#updateTitle').val().length);
99
100    //starting date
101    $('#updateStartingDay').val(event.startingDay);
102    $('#updateStartingMonth option[value=' + event.startingMonth + ']').attr(
103        'selected', 'selected');
104    $('#updateStartingYear').val(event.startingYear);
```

```
C:\xampp\htdocs\TPI\js\lightbox.js
lundi 19 juin 2017 16:13

104    //ending date
105    $("#updateEndingDay").val(event.endingDay);
106    $("#updateEndingMonth option[value=" + event.endingMonth + "]).attr('selected',
107    'selected');
108    $("#updateEndingYear").val(event.endingYear);
109
110    // Description
111    $('#updateDescription').val(event.description);
112
113    // Image + description width & height
114    if(event.imgFileName != null){ //If there was already an image, add it
115        $("#rightBlock").prepend('');
116        //space taken by the image, with some space around it (used to calculate the new width of the textarea)
117        var imgAndSpaceWidth = parseInt($('#updateLightbox img').css('width')) +
118            parseInt($('#updateLightbox').css('padding')) * 2 + parseInt($('#updateDescription').css('padding')) + 'px';
119        $('#updateDescription').css('width', "calc(100% - " + imgAndSpaceWidth + ")");
120    } else {
121        $('#updateDescription').css('width', '70%');
122    }
123
124    //categories
125    $.each(event.categories, function(index, value){
126        $('#option' + value[0]).attr('selected', 'selected');
127    });
128
129
130    // Save the changes made
131    function saveUpdate(){
132        if(typeof $('#updateLightbox img') !== 'undefined') // new image chosen
133            var oldImgName = $('#updateLightbox img').attr("alt");
134        else
135            var oldImgName = null;
136
137        //Check if the days have new values or not
138        // (the year are obligatory and the empty option of the select has the value of null, so no need to do it for them)
139        if($('#insertStartingDay').val().length == 0)
140            var startingDay = null;
141        else
142            var startingDay = $('#insertStartingDay').val();
143
144        if($('#insertEndingDay').val().length == 0)
145            var endingDay = null;
146        else
147            var endingDay = $('#insertStartingDay').val();
148
149        var tempEvent = {
150            idEvent : $('#updateLightbox').attr('class').split('t')[1],
151            title : $('#updateTitle').val(),
152            description : $('#updateDescription').val(),
153            imgFileName : oldImgName,
154            startingDay : startingDay,
```

```
C:\xampp\htdocs\TPI\js\lightbox.js                                         lundi 19 juin 2017 16:13
155      startingMonth : $('#updateStartingMonth').val(),
156      startingYear : parseInt($('#updateStartingYear').val()),
157      endingDay : endingDay,
158      endingMonth : $('#updateEndingMonth').val(),
159      endingYear : parseInt($('#updateEndingYear').val()),
160      categories : $('#updateCategories').val()
161  }
162
163 //Since we prevent the processing of data (in order to send a file), we must
164 //create the formData object ourselves
164 var fData = new FormData();
165 fData.append('event', JSON.stringify(tempEvent));
166
167 if(typeof $('#fileInput')[0].files[0] !== 'undefined') // new image chosen
168   fData.append('img', $('#fileInput')[0].files[0]);
169
170 //get the range of year before the update
171 $.ajax({
172   url : './functions/interlocutor/rangeYear.php',
173   type : 'GET',
174   dataType : 'json',
175   async: false, //I need the next call to be done after this one
176   success: function(rangeData){
177     oldRangeData = rangeData;
178   }
179 });
180
181 $.ajax({
182   url : './functions/interlocutor/updateEvent.php',
183   type : 'POST',
184   dataType : 'html',
185   processData: false, //prevent jquery from trying to convert the formData to
186   // a string
186   contentType: false, // prevent the contentType headers to be sent, in order
187   // to have the boundary string present
188   data : fData,
189   async: false, //So the user can directly see the result before doing
190   // something else
191   success: function(msg){
192     alert(msg);
193     $('#updateLightbox form')[0].reset();
194     $('#updateLightbox img').remove();
195     $('#updateLightbox").hide();
196     $('#main').empty();
197     //only and admin can update the event, so no need to check if the admin
198     // is connected
199     $('#main').append('');
201
202     $.each(checkedCatIdArray, function(){
203       getEvents(this);
204     });
205
206     displayLightbox('dot' + tempEvent.idEvent); //reload the lightbox with
207     // the new info
208
209     compareRange(tempEvent, oldRangeData);

```

```
C:\xampp\htdocs\TPI\js\lightbox.js                                         lundi 19 juin 2017 16:13
205      }
206    });
207  }
208
209 //Check if the updated event year are smaller or bigger than the min and max of the
210 //if it's the case, reload the slider
211 function compareRange(tempEvent, oldRangeData){
212
213   var newRangeData = [];
214   var mustReload = false;
215
216   if(tempEvent.startingYear < oldRangeData[0]['minYear']){
217     newRangeData[0] = {'minYear' : tempEvent.startingYear, 'maxYear' :
218       oldRangeData[0]['maxYear']};
219     mustReload = true;
220   }
221
222   if(tempEvent.endingYear > oldRangeData[0]['maxYear']){
223     //i need it to be in this format because other functions call addSlider with
224     //it
225     //and addSlider doesn't check the format in which it receive data
226     newRangeData[0] = {'minYear' : newRangeData[0].minYear, 'maxYear' : tempEvent
227       .endingYear};
228     mustReload = true;
229   }
230
231   if(mustReload){
232     $("#slider").empty();
233     addSlider(newRangeData);
234   }
235
236 //Cancel the update, reload the lightbox with the old info
237 function cancelUpdate(){
238   var idEvent = $('#updateLightbox').attr('class').split('t')[1];
239
240   $('#updateLightbox form')[0].reset();
241   $('#updateLightbox img').remove();
242   $('#updateLightbox').hide();
243   displayLightbox('dot' + idEvent);
244
245 //Show the lightbox enabling the creation of an event
246 function addEvent(){
247   $('#lightboxOverlay').show();
248   $('#addLightbox').show();
249
250 //Insert the new event
251 function insertData() {
252   //Check if the days have new values or not
253   //(the year are obligatory and the empty option of the select has the value of
254   //null, so no need to do it for them)
255   if($('#insertStartingDay').val().length == 0)
256     var startingDay = null;
257   else
```

```
C:\xampp\htdocs\TPI\js\lightbox.js                                         lundi 19 juin 2017 16:13
257     var startingDay = $('#insertStartingDay').val();
258
259     if($('#insertEndingDay').val().length == 0)
260         var endingDay = null;
261     else
262         var endingDay = $('#insertStartingDay').val();
263
264     var tempEvent = {
265         title : $('#insertTitle').val(),
266         description : $('#insertDescription').val(),
267         startingDay : startingDay,
268         startingMonth : $('#insertStartingMonth').val(),
269         startingYear : parseInt($('#insertStartingYear').val()),
270         endingDay : endingDay,
271         endingMonth : $('#insertEndingMonth').val(),
272         endingYear : parseInt($('#insertEndingYear').val()),
273         categories : $('#insertCategories').val()
274     }
275
276     //Since we prevent the processing of data (in order to send a file), we must
277     //create the formData object ourselves
278     var fData = new FormData();
279     fData.append('event', JSON.stringify(tempEvent));
280
281     if(typeof $('#insertImgFile')[0].files[0] !== 'undefined') // new image chosen
282         fData.append('img', $('#imgFile')[0].files[0]);
283
284     var oldRangeData;
285
286     //get the range of year before the insert
287     $.ajax({
288         url : './functions/interlocutor/rangeYear.php',
289         type : 'GET',
290         dataType : 'json',
291         async: false, //I need the next call to be done after this one
292         success: function(rangeData){
293             oldRangeData = rangeData;
294         }
295     });
296
297     //insert the data
298     $.ajax({
299         url : './functions/interlocutor/insertEvent.php',
300         type : 'POST',
301         dataType : 'html',
302         processData: false, //prevent jquery from trying to convert the formData to
303         //a string
304         contentType: false, // prevent the contentType headers to be sent, in order
305         //to have the boundary string present
306         data : fData,
307         async: false, //So the user can directly see the result before doing
308         //something else
309         success: function(msg){
310             alert(msg);
311             $('#addLightbox form')[0].reset();
312             $('#updateLightbox img').remove();
313             $("#addLightbox").hide();
314         }
315     });
316
317     //refresh the calendar
318     $('#calendar').fullCalendar('refetchEvents');
319
320     //close the lightbox
321     $('#addLightbox').lightBox();
322
323     //refresh the calendar
324     $('#calendar').fullCalendar('refetchEvents');
```

```
C:\xampp\htdocs\TPI\js\lightbox.js                                         lundi 19 juin 2017 16:13
310      $("#lightboxOverlay").hide();
311      $('main').empty();
312      //only and admin can update the event, so no need to check if the admin
313      //is connected
314      $('main').append('');
316
317      //reload the data so they are up to date
318      $.each(checkedCatIdArray, function(){
319          getEvents(this);
320      });
321
322      //Check if the new event year are smaller or bigger than the min and max
323      //of the old range
324      //if it's the case, create a new object rangeData to send to the slider
325      var newRangeData = [];
326      var mustReload = false;
327
328      if(tempEvent.startingYear < oldRangeData[0]['minYear']){
329          newRangeData[0] = {'minYear' : tempEvent.startingYear, 'maxYear' :
330          oldRangeData[0]['maxYear']};
331          mustReload = true;
332      }
333
334      if(tempEvent.endingYear > oldRangeData[0]['maxYear']){
335          //i need it to be in this format because other functions call
336          //addSlider with it
337          //and addSlider doesn't check the format in which it receive data
338          if(typeof newRangeData[0] != 'undefined')
339              newRangeData[0] = {'minYear' : newRangeData[0].minYear, 'maxYear'
340              : tempEvent.endingYear};
341          else
342              newRangeData[0] = {'minYear' : oldRangeData[0]['minYear'],
343              'maxYear' : tempEvent.endingYear};
344          mustReload = true;
345      }
346
347      if(mustReload){
348          slider.noUiSlider.destroy();
349          addSlider(newRangeData);
350      }
351
352      function displayMiniInfo(){
353          var animeDotWidth = "10px";
354          var animeBorderWidth = "3px";
355          var dot = $(this); //if i use it directly in the setTimeout, the $(this) will
356          //represent the window in place of the dot
357
358          timeoutId = setTimeout(function{
359              dot.animate({
360                  height: "10px",

```

```
C:\xampp\htdocs\TPI\js\lightbox.js                                         lundi 19 juin 2017 16:13
359         width: animeDotWidth,
360         borderWidth: animeBorderWidth,
361         borderColor: '#000000',
362         zIndex: 8
363     }, 250);
364
365     dot.on('click', { idEvent: dot[0].id }, displayLightbox);
366
367     //prepare the info to add to the miniBox
368     var event = eventList[dot[0].id];
369     var dates = treatEventDate(event.startingDay, event.startingMonth, event.
startingYear, event.endingDay, event.endingMonth, event.endingYear);
370
371     //if the two dates are the same, only the first one will be shown
372     //otherwise the two will be
373     var dateString = '<p>' + dates[0];
374
375     if(dates[0] != dates[1])
376         dateString += ' - ' + dates[1];
377
378     dateString += '</p>';
379
380     //add the box and the info
381     dot.append('<div class="miniInfo"></div>');
382     $(".miniInfo").append('<h4>' + event.title + '</h4>');
383     $(".miniInfo").append(dateString);
384
385     //Check that the miniBox isn't outside of the screen (left side) (by trying
386     //to be at the center of the dot) and move it if it's the case
387     var diffLeftEdgeAndDot = parseInt(dot.css('left')) - parseInt($(".miniInfo").
css('right'));
388     if(diffLeftEdgeAndDot < 0)
389         $('.miniInfo').css('right', parseInt($('.miniInfo').css('right')) - Math.
abs(diffLeftEdgeAndDot));
390
391     //Check that the miniBox isn't outside of the screen (right side) (by trying
392     //to be at the center of the dot) and move it if it's the case
393     var miniRightSidePos = parseInt($('.main').css('margin-left')) + (parseInt(dot.
css('left')) + parseInt($(".miniInfo").css('width')) / 2);
394     if(miniRightSidePos > parseInt($('body').css('width'))){
395         var diff = miniRightSidePos - parseInt($('body').css('width'));
396         $('.miniInfo').css('right', parseInt($('.miniInfo').css('right')) + diff);
397     }
398
399     // prevent the box to go up when the dot stacks
400     $('.miniInfo').css('top', parseInt(dot.css('bottom')) + 20);
401
402     //If the starting and ending date are different add the ending dot and the
403     //line between the two
404     if(event.startingYear != event.endingYear){
405         var sliderValues = slider.noUiSlider.get();
406         var nbPixelPerYear = parseInt(dot.parent().css("width")) / (sliderValues[
1] - sliderValues[0]); //width of line / year represented by the line
407         var spaceBetweenDotAndLine = 10;
408
409         // -sliderValues[0] because we want the year difference between the
410         // start of the timeline (not the year 0) and the dot * nb pixel per 1 year
```

```
C:\xampp\htdocs\TPI\js\lightbox.js                                         lundi 19 juin 2017 16:13
407         var endingDotLeftPos = (event.endingYear - sliderValues[0]) *
408             nbPixelPerYear;
409
410         // half of dot & border width (during anim) + space between dot & line
411         //("margin")
412         var halfDotWithMargin = parseInt(animeDotWidth) / 2 + parseInt(
413             animeBorderWidth) / 2 + spaceBetweenDotAndLine;
414
415         //diff between beginning/ending year * nb pixel per 1 year - halfDot * 2
416         // (one for each dot)
417         var lineWidth = (event.endingYear - event.startingYear) * nbPixelPerYear
418             - (halfDotWithMargin * 2);
419
420         //add and stylise everything
421         dot.append('<div class="periodLine"></div>');
422         dot.after('<div class="endingDot" id="endingDot' + event.idEvent +
423             '"></div>');
424
425         $(".periodLine").css("left", halfDotWithMargin); //since the initial
426         position of the line is the middle of the dot
427         $(".periodLine").css("width", lineWidth);
428         $("#endingDot" + event.idEvent).css("left", endingDotLeftPos);
429
430     }
431 }, 500);
432 }
433
434 //Erase the displayed info on hover, reset the dot to its initial style
435 function removeMiniInfo(){
436     clearTimeout(timeoutId); //prevent the function after the timeout to start
437
438     //if the mouse leave the dot before the timeout was finish, nothing was added,
439     //so there's nothing to remove
440     if($(this).children().hasClass("miniInfo")){
441         //restore the initial style of the dot
442         $(this).animate({
443             height: "0",
444             width: "0",
445             borderWidth: "5px",
446             borderColor: '#FFFFFF',
447             zIndex: 5
448         }, 250);
449         $(this).empty();
450         $(this).unbind("click");
451
452         //if the starting/ending date were the same the endingDot wasn't added, so
453         //there's no need to remove it
454         if($(this).next().hasClass("endingDot"))
455             $(this).next().remove();
456     }
457
458     //Check which date info (day, month or year) are filled and create two string, one
459     //for the starting year and the other for the ending one
460     function treatEventDate(startingDay, startingMonth, startingYear, endingDay,
461     endingMonth, endingYear){
462         //Check if the starting/ending day/month aren't null (the year can't be)
463         var startingDate = '';
464         var endingDate = '';
```

C:\xampp\htdocs\TPI\js\lightbox.js

lundi 19 juin 2017 16:13

```
453
454     //starting
455     if(startingDay != null)
456         startingDate += startingDay + ' ' ;
457
458     if(startingMonth != null)
459         startingDate += startingMonth + ' ' ;
460
461     //ending
462     if(endingDay != null)
463         endingDate += endingDay + ' ' ;
464
465     if(endingMonth != null)
466         endingDate += endingMonth + ' ' ;
467
468     //add the year (never null)
469     startingDate += startingYear;
470     endingDate += endingYear;
471
472     return [startingDate, endingDate];
473 }
474
475 //Update the ending year min to be the value of the starting year (for the update
form)
476 function minInsertEndingYear(){
477     $("#insertEndingYear").attr("min", $("#insertStartingYear").val());
478 }
479
480 //Update the ending year min to be the value of the starting year (for the insert
form)
481 function minUpdateEndingYear(){
482     $("#updateEndingYear").attr("min", $("#updateStartingYear").val());
483 }
484
```

## 6. Functions/dao.php

C:\xampp\htdocs\TPI\functions\dao.php lundi 19 juin 2017 16:15

```
1 <?php
2 /*
3  * Description : all database request are in this file
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8 require_once("dbCoConfig.php");
9
10 // Create a PDO object following the singleton pattern
11 function myPdo() {
12     static $pdo = null; //thus, the PDO object won't be created again each time.
13
14     if($pdo == null) {
15         try {
16             $pdo = @new PDO('mysql:dbname=' . DB_NAME . ';host=' . HOST, USER,
17                             PASSWORD, array(
18                                 PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8",
19                                 PDO::ATTR_PERSISTENT => true
20                             ));
21
22             $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
23             //allows to send error
24
25         } catch(PDOException $e) {
26             die("Erreur de connexion à la base, contacter l'administrateur");
27         }
28     }
29
30     // Description : Get all the categories id and name
31     // Param :
32     // Return : Either an associative array of the categories or false
33     function getCategories() {
34         try {
35             $req = "SELECT * FROM category";
36             $req = myPdo()->prepare($req);
37             $req->execute();
38             return $req->fetchAll(PDO::FETCH_ASSOC);
39         } catch(Exception $e) {
40             error_log($e);
41             return false;
42         }
43     }
44
45     // Description : Get the minimum and maximum year of the database
46     // The minimum one is necessarily a startingYear but the maximum one
47     // can be either
48     // Param :
49     // Return : Either an associative array of the range or false
50     function getYearRange() {
51         try {
52             $req = "SELECT MIN(startingYear) AS minYear, GREATEST(MAX(startingYear),
53                         MAX(endingYear)) AS maxYear FROM event";
54             $req = myPdo()->prepare($req);
55             $req->execute();
```

```
C:\xampp\htdocs\TPI\functions\dao.php                                         lundi 19 juin 2017 16:15
54     return $req->fetchAll(PDO::FETCH_ASSOC);
55 } catch(Exception $e) {
56     error_log($e);
57     return false;
58 }
59 }
60
61 // Description : Get the name of a category from its id
62 // Param : idCategory (int)
63 // Return : Either the name of the category or false
64 function getCatName($idCategory) {
65     try {
66         $req = "SELECT name FROM category WHERE idCategory = :idCategory";
67         $req = myPdo()->prepare($req);
68         $req->bindParam(':idCategory', $idCategory);
69         $req->execute();
70         return $req->fetch(PDO::FETCH_NUM);
71     } catch(Exception $e) {
72         error_log($e);
73         return false;
74     }
75 }
76
77 // Description : Get all the event (and their data) between this two years and
78 // belonging to this category
79 // Param : idCategory (int), minYear (int), maxYear (int)
80 // Return : Either an associative array of the event or false
81 function getEvents($idCategory, $minYear, $maxYear) {
82     try {
83         $req = "SELECT DISTINCT event.* FROM event JOIN event_has_category
84             USING(idEvent) WHERE idCategory = :idCategory AND startingYear BETWEEN
85             :minYear AND :maxYear ORDER BY startingYear";
86         $req = myPdo()->prepare($req);
87         $req->bindParam(':idCategory', $idCategory);
88         $req->bindParam(':minYear', $minYear);
89         $req->bindParam(':maxYear', $maxYear);
90         $req->execute();
91         return $req->fetchAll(PDO::FETCH_ASSOC);
92     } catch(Exception $e) {
93         error_log($e);
94         return false;
95     }
96
97 // Description : Get all the users
98 // Param : /
99 // Return : Either an associative array of the user or false
100 function getUser() {
101     try {
102         $req = "SELECT * FROM user";
103         $req = myPdo()->prepare($req);
104         $req->execute();
105         return $req->fetchAll(PDO::FETCH_ASSOC);
106     } catch(Exception $e) {
107         error_log($e);
108         return false;
109     }
110 }
```

```
C:\xampp\htdocs\TPI\functions\dao.php                                         lundi 19 juin 2017 16:15
108 }
109
110 // Description : Get All the categories for an event with its id
111 // Param : idEvent (int)
112 // Return : Either an array of the categories or false
113 function getAllCatForEvent ($idEvent) {
114     try {
115         $req = "SELECT idCategory FROM event_has_category WHERE idEvent = :idEvent";
116         $req = myPdo ()->prepare($req);
117         $req->bindParam(':idEvent', $idEvent);
118         $req->execute();
119         return $req->fetchAll(PDO::FETCH_NUM);
120     } catch (Exception $e) {
121         error_log($e);
122         return false;
123     }
124 }
125
126 // Description : Update the event with the new data
127 // Param : idEvent (int), title (string), description (string), imgFileName (string),
128 //          startingDay (int), startingMonth (string), startingYear (int), endingDay
129 //          (int), endingMonth (string), endingYear(int)
130 // Return : / (may throw an exception)
131 function updateEvent ($idEvent, $title, $description, $imgFileName, $startingDay,
132 $startingMonth, $startingYear, $endingDay, $endingMonth, $endingYear){
133     try {
134         $req = "UPDATE event SET title = :title, description = :description,
135             imgFileName = :imgFileName, startingDay = :startingDay, startingMonth =
136             :startingMonth, startingYear = :startingYear, endingDay = :endingDay,
137             endingMonth = :endingMonth, endingYear = :endingYear WHERE idEvent =
138             :idEvent";
139         $req = myPdo ()->prepare($req);
140         $req->bindParam(':idEvent', $idEvent);
141         $req->bindParam(':title', $title);
142         $req->bindParam(':description', $description);
143         $req->bindParam(':imgFileName', $imgFileName);
144         $req->bindParam(':startingDay', $startingDay);
145         $req->bindParam(':startingMonth', $startingMonth);
146         $req->bindParam(':startingYear', $startingYear);
147         $req->bindParam(':endingDay', $endingDay);
148         $req->bindParam(':endingMonth', $endingMonth);
149         $req->bindParam(':endingYear', $endingYear);
150         $req->execute();
151     } catch (Exception $e) {
152         error_log($e);
153         //i don't return false here because this request is in a transaction, it
154         //seem more logical to process it higher
155         throw $e;
156     }
157 }
158
159 // Description : Update the category with the new data
160 // Param : catId (int), catName (string)
161 // Return : / (may throw an exception)
162 function updateCategory ($catName, $catId) {
163     try {
164         $req = "UPDATE category SET name = :catName WHERE idCategory = :idCategory";
165     }
```

```
C:\xampp\htdocs\TPI\functions\dao.php                                         lundi 19 juin 2017 16:15
158     $req = myPdo ()->prepare($req);
159     $req->bindParam(':idCategory', $catId);
160     $req->bindParam(':catName', $catName);
161     $req->execute();
162     return true;
163 } catch(Exception $e) {
164     error_log($e);
165     //i don't return false here because this request is in a transaction, it
166     //seem more logical to process it higher
167     throw $e;
168 }
169
170 // Description : Insert the new event
171 // Param : title (string), description (string), imgFileName (string), startingDay
172 // (int), startingMonth (month), startingYear (int), endingDay (int), endingMonth
173 // (string), endingYear (int)
174 // Return : / (may throw an exception)
175 function insertEvent($title, $description, $imgFileName, $startingDay, $startingMonth,
176 , $startingYear, $endingDay, $endingMonth, $endingYear) {
177     try {
178         $req = "INSERT INTO event (title, description, imgFileName, startingDay,
179         startingMonth, startingYear, endingDay, endingMonth, endingYear)
180         VALUES(:title, :description, :imgFileName, :startingDay, :startingMonth,
181         :startingYear, :endingDay, :endingMonth, :endingYear)";
182         $req = myPdo ()->prepare($req);
183         $req->bindParam(':title', $title);
184         $req->bindParam(':description', $description);
185         $req->bindParam(':imgFileName', $imgFileName);
186         $req->bindParam(':startingDay', $startingDay);
187         $req->bindParam(':startingMonth', $startingMonth);
188         $req->bindParam(':startingYear', $startingYear);
189         $req->bindParam(':endingDay', $endingDay);
190         $req->bindParam(':endingMonth', $endingMonth);
191         $req->bindParam(':endingYear', $endingYear);
192         $req->execute();
193     } catch(Exception $e) {
194         error_log($e);
195         //i don't return false here because this request is in a transaction, it
196         //seem more logical to process it higher
197         throw $e;
198     }
199
200 // Description : Insert the new event-category relation
201 // Param : idEvent (int), idCategory (int)
202 // Return : / (may throw an exception)
203 function insertRelation($idEvent, $idCategory) {
204     try {
205         $req = "INSERT INTO event_has_category (idEvent, idCategory)
206         VALUES(:idEvent, :idCategory)";
207         $req = myPdo ()->prepare($req);
208         $req->bindParam(':idEvent', $idEvent);
209         $req->bindParam(':idCategory', $idCategory);
210         $req->execute();
211     } catch(Exception $e) {
212         error_log($e);
213     }
214 }
```

```
C:\xampp\htdocs\TPI\functions\dao.php                                         lundi 19 juin 2017 16:15
206         //i don't return false here because this request is in a transaction, it
207         // seem more logical to process it higher
208     }
209 }
210
211 // Description : Insert the new category
212 // Param : catName (string)
213 // Return : Either true or false
214 function insertCategory($catName){
215     try {
216         $req = "INSERT INTO category(name) VALUES(:catName)";
217         $req = myPdo()->prepare($req);
218         $req->bindParam(':catName', $catName);
219         $req->execute();
220         return true;
221     } catch(Exception $e) {
222         error_log($e);
223         return false;
224     }
225 }
226
227 // Description : Delete the event
228 // Param : idEvent (int)
229 // Return : Either true or false
230 function deleteEvent($idEvent){
231     try {
232         $req = "DELETE FROM event WHERE idEvent = :idEvent";
233         $req = myPdo()->prepare($req);
234         $req->bindParam(':idEvent', $idEvent);
235         $req->execute();
236         return true;
237     } catch(Exception $e) {
238         error_log($e);
239         return false;
240     }
241 }
242
243 // Description : Delete the event-category relation
244 // Param : idEvent (int)
245 // Return : / (may throw an exception)
246 function deleteRelation($idEvent){
247     try {
248         $req = "DELETE FROM event_has_category WHERE idEvent = :idEvent";
249         $req = myPdo()->prepare($req);
250         $req->bindParam(':idEvent', $idEvent);
251         $req->execute();
252     } catch(Exception $e) {
253         error_log($e);
254         //i don't return false here because this request is in a transaction, it
255         // seem more logical to process it higher
256         throw $e;
257     }
258
259 // Description : Delete the category
260 // Param : catId (int)
```

```
C:\xampp\htdocs\TPI\functions\dao.php  
lundi 19 juin 2017 16:15  
261 // Return : Either true or false  
262 function deleteCategory($catId){  
263     try {  
264         $req = "DELETE FROM category WHERE idCategory = :catId";  
265         $req = myPdo ()->prepare($req);  
266         $req->bindParam(':catId', $catId);  
267         $req->execute();  
268         return true;  
269     } catch(Exception $e) {  
270         error_log($e);  
271         return false;  
272     }  
273 }
```

## 7. Functions/dbCoConfig.php

```
C:\xampp\htdocs\TPI\functions\dbCoConfig.php  
lundi 19 juin 2017 16:15  
1 <?php  
2 /*  
3 * Description : Constant for the database connexion  
4 * Author : Zoé Cugni  
5 * Date : June 2017  
6 * Context : school last project (TPI)  
7 */  
8  
9 define("DB_NAME", "chronoview");  
10 define("HOST", "localhost");  
11 define("USER", "chrono_superAdmin");  
12 define("PASSWORD", "CDDdHCUUSTBbMrYTj");
```

## 8. Functions/interlocutor/event.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\event.php                                         lundi 19 juin 2017 16:16
1 <?php
2 /*
3  * Description : Answer to a call ajax,
4  *                 retrieve data on events in the database and form an array with them
5  * Author : Zoé Cugni
6  * Date : June 2017
7  * Context : school last project (TPI)
8 */
9
10 $idCategory = filter_input(INPUT_GET, 'idCategory', FILTER_VALIDATE_INT);
11 $minYear = filter_input(INPUT_GET, 'minYear', FILTER_VALIDATE_INT);
12 $maxYear = filter_input(INPUT_GET, 'maxYear', FILTER_VALIDATE_INT);
13
14 require_once('../dao.php');
15
16 if($idCategory != false && $minYear != false && $maxYear != false) {
17     $eventArray = getEvents($idCategory, $minYear, $maxYear);
18     $catName = getCatName($idCategory);
19
20     foreach($eventArray as $event)
21         $event['categories'] = getAllCatForEvent($event['idEvent']);
22
23     array_unshift($eventArray, $catName);
24     array_unshift($eventArray, $idCategory);
25
26     echo json_encode($eventArray);
27 }
```

## 9. Functions/interlocutor/getCategories.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\getCategories.php                               lundi 19 juin 2017 16:16
1 <?php
2 /*
3  * Description : Answer to a call ajax, retrieve the categories
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8 require_once('../dao.php');
9
10 echo json_encode(getCategories());
```

## 10. Functions/interlocutor/getSession.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\getSession.php
lundi 19 juin 2017 16:17

1 <?php
2 /*
3  * Description : Answer to a call ajax, check if the admin is connected or not
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8 session_start();
9 require_once('../dao.php');

10
11 if(isset($_SESSION['admin']) && $_SESSION['admin'] == true)
12     echo true;
13 else
14     echo false;
```

## 11. Functions/interlocutor/loginCheck.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\loginCheck.php
lundi 19 juin 2017 16:18

1 <?php
2 /*
3  * Description : Answer to a form, check if the given login are legit
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8
9 session_start();
10 require_once('../dao.php');
11 $salt = 'Lorem ipsum dolor sit amet';
12
13 if(isset($_SESSION['admin']) && $_SESSION['admin']){ //If the admin was connected he
14     clicked on the 'disconnect' button
15     session_unset();
16     header("Location: ../../index.php");
17     exit();
18 } else {
19     $email = filter_input(INPUT_POST, 'mail', FILTER_SANITIZE_EMAIL);
20     $pw = filter_input(INPUT_POST, 'pw', FILTER_SANITIZE_STRING) . $salt;
21
22     if(filter_var($email, FILTER_VALIDATE_EMAIL) != false){
23         foreach(getUser() as $user){
24             if($user['email'] == $email && password_verify($pw, $user['pwHash']) &&
25                 $user['isAdmin'] == '1'){
26                 $_SESSION['admin'] = true;
27                 header("Location: ../../index.php");
28                 exit();
29             }
30         }
31     }
32     header("Location: ../../login.php?error=true");
```

## 12. Functions/interlocutor/rangeYear.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\rangeYear.php                               lundi 19 juin 2017 16:18
1 <?php
2 /*
3  * Description : Answer to an ajax call, retrieve the range of years
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8 require_once("../dao.php");
9 echo json_encode(getYearRange());
```

## 13. Functions/interlocutor/updateEvent.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\updateEvent.php                                         lundi 19 juin 2017 16:18
1  <?php
2  /*
3   * Description : Answer to an ajax call, update an event if valid data are given
4   * Author : Zoé Cugni
5   * Date : June 2017
6   * Context : school last project (TPI)
7  */
8  require_once('../dao.php');
9
10 //TO_DO : faire filter_input_array
11 $event = json_decode($_POST['event']);
12 $oldImgName = $event->imgFileName;
13
14 try {
15     if(!empty($_FILES)){ //new image
16         $imgName = $_FILES['img']['name'];
17         $imgFilePath = '../../img/' . $imgName;
18         move_uploaded_file($_FILES['img']['tmp_name'], $imgFilePath);
19
20         if(!$oldImgName) //no need to check if the file exist, if the old
21             path is given it must exist
22             unlink('../../img/' . $oldImgName);
23     } else { // no new image
24         $imgName = $oldImgName;
25     }
26
27     myPDO()->beginTransaction();
28
29     updateEvent($event->idEvent, $event->title, $event->description, $imgName,
30     $event->startingDay, $event->startingMonth, $event->startingYear, $event->
31     endingDay, $event->endingMonth, $event->endingYear);
32     deleteRelation($event->idEvent);
33
34     foreach($event->categories as $idCat)
35         insertRelation($event->idEvent, $idCat);
36
37     echo 'Événement bien modifié!';
38 } catch (Exception $e){
39     if(!empty($imgFilePath) && file_exists($imgFilePath)) //cancel upload if
40         there was a problem
41         unlink($imgFilePath);
42
43     myPDO()->rollBack();
44
45     echo "Une erreur est survenue, modification impossible. Contacter
46     l'administrateur";
47 }
```

## 14. Functions/interlocutor/updateCategory.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\updateCategory.php                                         lundi 19 juin 2017 16:19
1 <?php
2 /*
3  * Description : Answer to an ajax call, update a category if valid data are given
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8 require_once('../dao.php');
9
10 $catName = filter_input(INPUT_POST, 'newName', FILTER_SANITIZE_STRING);
11 $catId = filter_input(INPUT_POST, 'id', FILTER_VALIDATE_INT);
12
13 if(updateCategory($catName, $catId)) {
14     echo 'Catégorie bien mise à jour!';
15 } else {
16     echo "Une erreur est survenue, modification impossible. Contacter
17         l'administrateur";
18 }
```

## 15. Functions/interlocutor/insertCategory.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\insertCategory.php                                         lundi 19 juin 2017 16:20
1 <?php
2 /*
3  * Description : Answer to a call ajax, insert a category if a valid name is given
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8
9 require_once('../dao.php');
10
11 $catName = filter_input(INPUT_POST, 'name', FILTER_SANITIZE_STRING);
12
13 if(insertCategory($catName)) {
14     echo 'Catégorie bien ajoutée!';
15 } else {
16     echo "Une erreur est survenue, ajout impossible. Contacter l'administrateur";
17 }
```

## 16. Functions/interlocutor/insertEvent.php

```
C:\xampp\htdocs\TPI\functions\interlocutor\insertEvent.php                                         lundi 19 juin 2017 16:20
1  <?php
2  /*
3   * Description : Answer to a call ajax, insert an event if valid data are given
4   * Author : Zoé Cugni
5   * Date : June 2017
6   * Context : school last project (TPI)
7   */
8
9  require_once ('../dao.php');
10
11 //TO_DO : faire filter_input_array
12 $event = json_decode($_POST['event']);
13
14 try {
15     if (!empty($_FILES)) { //new image
16         $imgName = $_FILES['img']['name'];
17         $imgFilePath = '../../img/' . $imgName;
18         move_uploaded_file($_FILES['img']['tmp_name'], $imgFilePath);
19     } else {
20         $imgName = null;
21     }
22
23     myPDO() ->beginTransaction();
24
25     insertEvent($event->title, $event->description, $imgName, $event->startingDay,
26     , $event->startingMonth, $event->startingYear, $event->endingDay, $event->
27     endingMonth, $event->endingYear);
28     $idPost = myPDO() ->lastInsertId();
29
30     foreach ($event->categories as $idCategory)
31         insertRelation($idPost, $idCategory);
32
33     myPDO() ->commit();
34
35     echo 'Evénement bien ajouté!';
36 } catch (Exception $e) {
37     if (!empty($imgFilePath) && file_exists($imgFilePath)) //cancel upload if
38     there was a problem
39     unlink($imgFilePath);
40
41     myPDO() ->rollBack();
42     echo "Une erreur est survenue, ajout impossible. Contacter l'administrateur";
43 }
```

## 17. Functions/interlocutor/deleteCategory.php

C:\xampp\htdocs\TPI\functions\interlocutor\deleteCategory.php lundi 19 juin 2017 16:19

```

1 <?php
2 /*
3  * Description : Answer to a call ajax, delete a category if a valid id is received
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8
9 require_once('../dao.php');
10 $catId= filter_input(INPUT_POST, 'id', FILTER_VALIDATE_INT);
11
12 if(deleteCategory($catId)){
13     echo 'Catégorie bien supprimée!';
14 } else {
15     echo "Une erreur est survenue, suppression impossible. Contacter
16         l'administrateur";
17 }
```

## 18. Functions/interlocutor/deleteEvent.php

C:\xampp\htdocs\TPI\functions\interlocutor\deleteEvent.php lundi 19 juin 2017 16:20

```

1 <?php
2 /*
3  * Description : Answer to a call ajax, delete an event if a valid id is received
4  * Author : Zoé Cugni
5  * Date : June 2017
6  * Context : school last project (TPI)
7 */
8
9 require_once('../dao.php');
10 $eventId= filter_input(INPUT_POST, 'id', FILTER_VALIDATE_INT);
11
12 if(deleteEvent($eventId)){
13     echo 'Événement bien supprimé!';
14 } else {
15     echo "Une erreur est survenue, suppression impossible. Contacter
16         l'administrateur";
17 }
```

## 19. Css/style.scss

```
C:\xampp\htdocs\TPI\css\style.scss                                         lundi 19 juin 2017 16:21
1   $lightboxPadding: 15px;
2   $lightboxBorder: 5px;
3   $lightboxTopPos: 5%;
4   $lightboxSideMargin: 5%;
5
6   $dotColor: white;
7
8   $miniInfoWidth: 140px;
9   $miniInfoPadding: 5px;
10
11  $endingDotColor: lightgreen;
12
13  $loginWidth: 100px;
14
15  $sidebarBlockPadding : 5px;
16
17  $leftFormWidth : 50%;
18
19  $sliderWidthReduc : 50px;
20
21 // *** General ***
22
23 html {
24     height: 100%;
25 }
26
27 body {
28     margin:0;
29     height: 100%;
30     overflow-y: hidden; //for some reason there was some blank space added that made
31     a vertical scrollbar appear
32     background-image: url('../img/library1.jpg');
33     background-size: cover;
34     background-repeat: no-repeat;
35     font-family: 'PT Sans Caption', sans-serif;
36     font-family: 'Jura', sans-serif;
37 }
38
39 #blackCloak {
40     width: 100%;
41     height: 100%;
42     position: absolute;
43     top: 0;
44     background-color: rgba(0,0,0,0.3);
45 }
46
47 h1 {
48     text-align: center;
49     display: inline-block;
50     color: white;
51     margin: auto;
52     width: 70%;
53 }
54
55 .inline {
56     display: inline-block;
57 }
```

C:\xampp\htdocs\TPI\css\style.scss lundi 19 juin 2017 16:21

```
57
58     .clear {
59         clear: both;
60     }
61
62     label, input {
63         display: block;
64         margin-bottom: 10px;
65         border: 0;
66     }
67
68     input[type=number]::-webkit-inner-spin-button,
69     input[type=number]::-webkit-outer-spin-button {
70         opacity: 1;
71     }
72
73     //Login
74     #login {
75         background-image: url('../img/library2.jpg');
76         background-size: cover;
77         background-repeat: no-repeat;
78
79         form {
80             width: 200px;
81             margin: auto;
82             margin-top: 23%;
83         }
84
85         label {
86             color: white;
87             font-weight: bold;
88             margin: auto;
89             width: 170px;
90         }
91
92         label:nth-of-type(2) {
93             margin-top: 20px;
94         }
95
96         input {
97             margin: auto;
98         }
99     }
100
101    #login input[type='submit'] {
102        margin: auto;
103        background-color: #3FB8AF;
104        padding: 5px 10px;
105        color: white;
106        margin-top: 20px;
107    }
108
109    #error {
110        color: black;
111        background-color: white;
112        border: 2px red solid;
113        padding: 5px;
```

```
C:\xampp\htdocs\TPI\css\style.scss                                         lundi 19 juin 2017 16:21
114         margin-top: 15px;
115         width: 200px;
116     }
117
118 // *** Sidebar ***
119 #sidebarBlock {
120     padding: $sidebarBlockPadding;
121     color: white;
122     display: inline-block;
123
124     h2 {
125         margin: 0;
126         border-bottom: 1px solid white;
127         margin-bottom: 5px;
128     }
129 }
130
131 #listCat {
132     margin: auto;
133     list-style-type: none;
134     padding: 0;
135     display: none;
136     position: absolute;
137     z-index: 9;
138
139     li {
140         font-size: 20px;
141         padding-top: 15px;
142         border-bottom: 1px white dotted;
143         padding-right: 10px;
144     }
145
146     li:hover, .selected {
147         background-color: white;
148         border-bottom: 1px black dotted;
149         color: black;
150     }
151
152     .important:hover {
153         background-color: transparent;
154         border-bottom: 1px white dotted;
155         color: white;
156     }
157
158     .edit, .supp {
159         width: 20px;
160         position: absolute;
161     }
162
163     .supp {
164         right: 5px;
165     }
166
167     input {
168         border: 1px solid black;
169         display: inline-block;
170         color: black;
```

```
C:\xampp\htdocs\TPI\css\style.scss                                         lundi 19 juin 2017 16:21
171      }
172
173      p {
174          margin: 0;
175          display: inline-block;
176      }
177  }
178
179 #newCat {
180     img {
181         width: 20px;
182         margin-left: 10px;
183     }
184 }
185
186 // *** Main & Timelines ***
187
188 main {
189     width: unquote("calc(100% - #{$sliderWidthReduc})");
190     height: 100%;
191     margin: auto;
192 }
193
194 .timelineName{
195     position: absolute;
196     right: 0;
197     color: white;
198 }
199
200 .timeline {
201     height: 3px;
202     position: relative;
203     width: 100%;
204     margin-top: 100px;
205 }
206
207 .timeline:nth-child(odd){
208     background-color: rgb(102, 0, 51);
209 }
210
211 .timeline:nth-child(even){
212     background-color: rgb(204, 204, 0);
213 }
214
215 .dot, .endingDot {
216     border-radius: 100%;
217     background-color: $dotColor;
218     position: absolute;
219     bottom: -2.5px;
220 }
221
222 .endingDot {
223     border: lightgreen solid 3px;
224     width: 10px;
225     height: 10px;
226 }
227
```

```
C:\xampp\htdocs\TPI\css\style.scss                                         lundi 19 juin 2017 16:21
228     .dot {
229         border: $dotColor solid 5px;
230         width: 0;
231     }
232
233     .periodLine {
234         border: 3px dotted $endingDotColor;
235         position: absolute;
236         z-index: 8;
237         bottom: -1.5px;
238     }
239
240 // Lightbox & addLightBox
241
242 #addLightbox, #lightbox, #updateLightbox {
243     width: unquote("calc(100% - 2 * #{$lightboxSideMargin} - 2 * #{$lightboxPadding} - 2 * #{$lightboxBorder})");
244     height: unquote("calc(100% - 2 * #{$lightboxTopPos} - 2 * #{$lightboxPadding} - 2 * #{$lightboxBorder})");
245     padding: $lightboxPadding;
246     position: absolute;
247     top: $lightboxTopPos;
248     margin: 0 $lightboxSideMargin;
249     z-index: 10;
250     overflow-y: auto;
251 }
252
253 #lightboxOverlay {
254     height: 100%;
255     width: 100%;
256     z-index: 9;
257     background-color: rgb(119, 119, 119);
258     opacity: 0.9;
259     position: absolute;
260     top: 0;
261     display: none;
262 }
263
264 // Lightbox & updateLightbox
265 #lightbox, #updateLightbox {
266     background-color: black;
267     color: white;
268     border: $lightboxBorder solid white;
269
270     img {
271         float: right;
272         max-width: 250px;
273         margin-left: 15px;
274     }
275
276     button, input[type='submit'] {
277         color: white;
278         background-color: black;
279         text-decoration: none;
280         border: 1px solid white;
281         padding: 10px;
282         float: right;

```

C:\xampp\htdocs\TPI\css\style.scss lundi 19 juin 2017 16:21

```
283         margin-left: 10px;
284     }
285 }
286
287 // *** Lightbox ***
288
289 #lightbox {
290     text-align: justify;
291
292     h1 {
293         margin-top: 0;
294         text-align: left;
295     }
296
297     p {
298         line-height: 30px;
299         font-size: 17px;
300     }
301 }
302
303 // *** Add ***
304 #addLightbox {
305     display: none;
306     background-color: white;
307     color: black;
308     border: $lightboxBorder solid black;
309
310     h1 {
311         text-align: center;
312         color: black;
313         width: 100%;
314         margin-bottom: 20px;
315     }
316
317     form {
318         margin: auto;
319     }
320
321     textarea {
322         width: 100%;
323         display: block;
324     }
325
326     #insertStartingDay, #insertStartingMonth, #insertStartingYear, #insertEndingDay,
327     #insertEndingMonth, #insertEndingYear {
328         display: inline-block;
329     }
330
331     #left label:nth-of-type(2), #right label:nth-of-type(1), #right
332     label:nth-of-type(2) {
333         margin-top: 30px;
334     }
335
336     select {
337         margin: 0 20px;
338         border: 0;
339     }
```

C:\xampp\htdocs\TPI\css\style.scss lundi 19 juin 2017 16:21

```
338
339     #insertCategories {
340         margin: 0;
341         padding: 10px;
342         width: 250px;
343         height: 120px;
344     }
345
346     input, select {
347         border-bottom: 1px dashed grey;
348     }
349
350     input[type='submit']{
351         background-color: #3FB8AF;
352         color: white;
353         padding: 8px;
354         width: 100px;
355         border: 0;
356         margin: auto;
357         margin-top: 50px;
358     }
359
360     textarea, #insertCategories {
361         border: 1px dashed grey;
362     }
363
364     #left {
365         float: left;
366         width: $leftFormWidth;
367     }
368
369     #right {
370         float: right;
371         width: 100% - $leftFormWidth - 5%;
372     }
373
374     #leftInnerBlock, #rightInnerBlock {
375         margin: auto;
376     }
377 }
378
379 #insertDtartingDateBlock, #insertEndingDateBlock {
380     input {
381         display: inline-block;
382     }
383 }
384
385 #startingDateBlock label:nth-of-type(1) {
386     margin-top: 0 !important;
387 }
388
389 // *** Modif ***
390
391 #addIcon {
392     width: 50px;
393     position: absolute;
394     top: 5px;
```

C:\xampp\htdocs\TPI\css\style.scss lundi 19 juin 2017 16:21

```
395     right: 5px;
396     border: 2px solid white;
397     border-radius: 50%;
398   }
399
400
401 // *** Update lightbox ***
402 #updateLightbox {
403   display: none;
404
405   input[type='file'] {
406     display: block;
407     margin-top : 15px;
408   }
409
410   #updateTitle {
411     font-size: 2em;
412     margin-bottom: 0.67em;
413     display: inline-block;
414   }
415
416   textarea {
417     padding: 10px;
418     padding-bottom: 0;
419   }
420
421   #dateSection input, select, p{
422     display: inline-block;
423     margin-right: 10px;
424     margin-bottom: 10px;
425   }
426
427   #updateStartingDay, #updateEndingDay{
428     width: 40px;
429   }
430
431   #updateStartingYear, #updateEndingYear {
432     width: 50px;
433   }
434 }
435
436 #rightBlock {
437   display: inline-block;
438   vertical-align: top;
439   margin-left: 10px;
440
441   img {
442     float: none !important;
443     margin-left: 0 !important;
444   }
445
446   select, input {
447     display: block;
448   }
449
450   select {
451     width: 100%;
```

C:\xampp\htdocs\TPI\css\style.scss lundi 19 juin 2017 16:21

```
452      }
453
454     input {
455         margin-left: 0 !important;
456     }
457 }
458
459
460 // *** Mini Info ***
461
462 .miniInfo {
463     width: $miniInfoWidth;
464     border: 1px solid white;
465     color: white;
466     padding: $miniInfoPadding;
467     position: relative;
468     right: $miniInfoWidth / 2 - $miniInfoPadding;
469     z-index: 8;
470
471     h4, p {
472         margin: 0;
473     }
474 }
475
476 // *** Slider ***
477
478 #slider {
479     position: absolute;
480     bottom: 0;
481     width: unquote("calc(100% - #{$sliderWidthReduc})");
482     margin: 0 ($sliderWidthReduc / 2);
483 }
484
485 // overide of noUiSlider default style
486 .noUi-target {
487     border: 0;
488 }
489
490 .noUi-horizontal .noUi-handle {
491     top: -10px;
492     left: 0;
493     width: 5px;
494 }
495
496 .noUi-horizontal .noUi-handle:last-of-type {
497     left: -2px;
498 }
499
500 .noUi-horizontal .noUi-handle:before, .noUi-horizontal .noUi-handle:after {
501     content: none;
502 }
503
504 .noUi-base {
505     background-color: #FAF9F5;
506 }
507
508 .noUi-connect {
```

```
C:\xampp\htdocs\TPI\css\style.scss                                         lundi 19 juin 2017 16:21
509     background-color: #52374D;
510 }
511
```