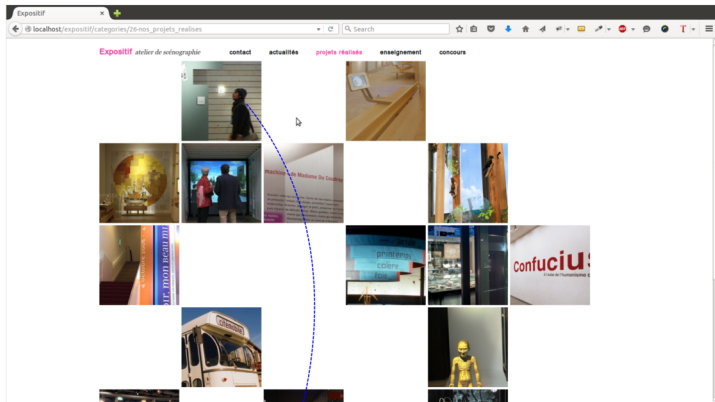


**Chris Carton / Extraits de code**

# Expositif / Mosaïque de projets

L'objectif était de générer une mosaïque à partir des images de "couverture" des projets pour chaque catégorie.  
Selon le brief client, la mosaïque devait comporter des espacements aléatoires.



Au survol de la souris, ce comportement apparaît



Lien vers le projet

```
2 .mosaic-thumbs{
3   width:175px;
4   height:175px;
5   float:left;
6   margin-right: 5px;
7   margin-bottom: 5px;
8 }
9
10
11 .mosaic-thumbs-zone{
12   background-color:#FB3099;
13   padding:5px;
14   top:0px;
15   right:0px;
16   left:0px;
17   bottom:0px;
18   opacity:0;
19   transition:0.5s ease-in;
20 }
21
22 .mosaic-thumbs-zone-mobile{
23   background-color:#FB3099;
24   padding:5px;
25   top:0px;
26   right:0px;
27   left:0px;
28   bottom:0px;
29   opacity:0.4;
30   transition:0.5s ease-in;
31 }
32
```

```
33 .mosaic-thumbs-zone-mobile:hover{
34   opacity:1;
35 }
36
37 .mosaic-thumbs-zone-blank{
38   background-color:transparent;
39   padding:5px;
40   top:0px;
41   right:0px;
42   left:0px;
43   bottom:0px;
44   opacity:0;
45   transition:0.5s ease-in;
46 }
47
48 .mosaic-thumbs-zone-inner{
49   display:table;
50   width: 100%;
51   height:100%;
52 }
53
54 .mosaic-thumbs-zone:hover{
55   opacity:0.8;
56 }
57
58 .mosaic-thumbs-link{
59   display:table-cell;
60   text-align: center;
61   vertical-align: middle;
62 }
63
64
```

CSS

```

1 <?php
2 //Parcours les projets pour cette catégorie
3 foreach($category['Project'] as $project):
4
5     if(!empty($project['Cover']['image'])):
6         $path_min = 'Cover'.DS.$project['id'].DS.'min'.DS.$project['Cover']['image'];
7     else:
8         //Si il n'y a pas de couverture, on met une image générique (fond coloré)
9         $path_min = 'min-lg.png';
10    endif;
11
12    $random = rand(0, 2);
13
14
15
16 for($i=0;$i<random;$i+=):
17 //creation aléatoire d'emplacement(s) vide(s)
18 >
19 <div classe="mosaic-thumbs relative">
20 <div class="absolute white mosaic-thumbs-zone-blank">
21 <div class="mosaic-thumbs-zone-inner">
22
23 </div>
24 </div>
25 </div>
26 <?php
27
28 endfor;
29 >
30
31 <div class="mosaic-thumbs relative">
32 <div class="absolute white mosaic-thumbs-zone">
33 <div class="mosaic-thumbs-zone-inner">
34 <?php
35
36 if(!empty($project['Referencement']['url'])):
37     $project['slug'] = $project['Referencement']['url'];
38 endif;
39
40 if(!empty($project['name'])):
41     echo $this->Html->link($project['name'],array(
42         'controller'=>'projects',
43         'action'=>'view',
44         'id'=>$project['id'],
45         'slug'=>$project['slug']
46     ),
47         array('class'=>'white bold bg-rose pd-10 sz 1-2 mosaic-thumbs-link')
48     );
49 endif;
50
51 >
52 </div>
53 </div>
54
55 <?php
56 //On affiche la miniature
57 echo $this->Html->image($path_min, array(
58     'class' => 'img-responsive'
59 ));
60
61 >
62
63 </div>
64
65 <?php endforeach; >

```

## CONTROLLER

```

263 public function view($id=null,$slug=null){
264     if(!empty($id) && is_numeric($id)){
265
266         $category = $this->Model->find('first',array(
267             'conditions'=>array('Category.id'=>$id),
268             'contain'=>array(
269                 'Project'=>array(
270                     'Cover',
271                     'Referencement'=>array('fields'=>'Referencement.url')
272                 )
273             ));
274
275         $this->setActiveCategory($category);
276         $this->set('category',$category);
277     }
278
279
280 }
281

```

Expositif atelier de scénographie

contact

actualités

projets réalisés

enseignement

concours

Permet de définir la catégorie active. Dans le parcours (*foreach*) des catégories, si l'id de la catégorie parcourue est identique à celui de la catégorie active, ajoute dans le code html la classe active (couleur rose).

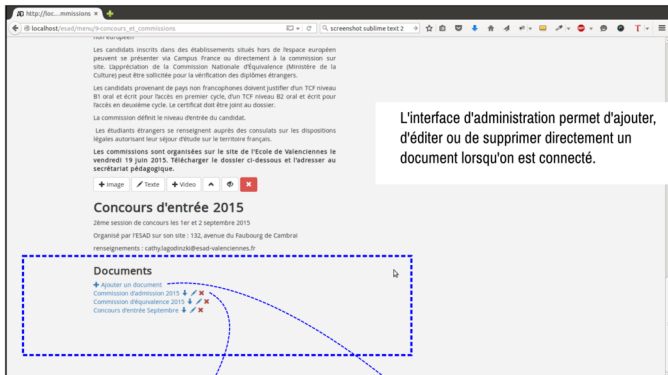
```

288 public function setActiveCategory($category=null){
289     if(!empty($category)):
290         if(!empty($category['Category']['id']) && is_numeric($category['Category']['id'])):
291             $this->set('category_active',$category['Category']['id']);
292         endif;
293     endif;
294

```

# ESAD - Système de document

Système de document pouvant être greffé à n'importe quelle partie du site. Par exemple : un contenu peut avoir un ou plusieurs documents. Une sous-catégorie peut avoir un ou plusieurs documents, etc...



## CONTROLLER

```
public function delete($id=null){
    $Model_name = $this->Model_name;

    if(!empty($id) && is_numeric($id)){
        $record = $this->Model->find('first',array(
            'conditions'=>array($Model_name['.id'=>$id]
        ));
        if(!empty($record)){
            //Permet de supprimer le fichier 'physiquement'
            $this->remove($id);

            if($this->Model->delete($id,$cascade=true)){
                $this->Session->setFlash('Document supprimé !');
                $ref_controller = Inflector::pluralize(strtolower($record[$Model_name]['ref']));
                return $this->redirect(array(
                    'controller'=>$ref_controller,
                    'action'=>'index',
                    $record[$Model_name]['ref_id']
                ));
            }else{
                throw new NotFoundException("Ce document n'existe pas !");
            }
        }else{
            throw new NotFoundException('Identifiant incorrect');
        }
    }else{
        throw new NotFoundException('Identifiant incorrect');
    }
}

//Permet de supprimer le fichier 'physiquement'
public function remove($id=null){
    $Model_name = $this->Model_name;

    if(!empty($id) && is_numeric($id)){
        $record = $this->Model->find('first',array(
            'conditions'=>array($Model_name['.id'=>$id]
        ));
        if(!empty($record[$Model_name]['file'])){
            $path_to_file = 'files/'.$Model_name.$record[$Model_name]['ref'].'.'.$record[$Model_name]['ref_id'].'.'.$record[$Model_name]['file'];
            $this->Model->deleteFile($path_to_file);
        }
    }
    endif;
}
```

```
public function add($ref=null,$ref_id=null){
    $Model_name = $this->Model_name;
    $controller_redirect = Inflector::pluralize(strtolower($Model_name));

    if(!empty($ref_id){
        throw new NotFoundException('Identifiant incorrect');
    }

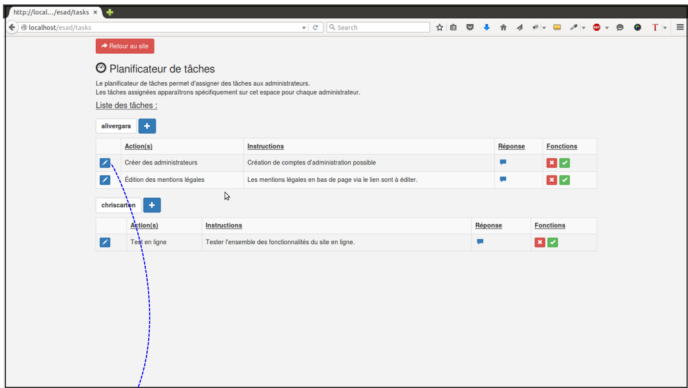
    if(!empty($ref) && is_string($ref)){
        //Recherche la première entrée vide
        $record = $this->Model->find('first', array(
            'conditions'=>array(
                $Model_name['.ref'=>$ref,
                $Model_name['.ref_id'=>$ref_id,
                $Model_name['.name'=>'
            ));
        //Si une entrée vide existe déjà
        if(!empty($record)){
            $this->Model->id = $record[$Model_name]['id'];
            //Redirection vers edit
            $this->redirect(array(
                'controller'=>$controller_redirect,
                'action'=>'edit',
                $record[$Model_name]['id']
            ));
        }else{
            //Si on l'insère
            $record = array(
                'ref'=>$ref,
                'ref_id'=>$ref_id
            );
            $this->Model->save($record,false);
            $redirect_id = $this->Model->getLastInsertId();
            //Redirection vers edit
            $this->redirect(array(
                'controller'=>$controller_redirect,
                'action'=>'edit',
                $redirect_id
            ));
        }
    }
    endif;
}
```

# ESAD - Administration / Planificateur de tâche

Lorsque l'on se connecte en tant qu'administrateur, le système ajoute à l'interface cet élément d'administration donnant accès à diverses fonctionnalités dont le planificateur de tâches.



Le planificateur de tâches permet d'attribuer à chaque administrateur du site des tâches à effectuer. C'est une sorte de "to-do list".



```
public function edit($id=null){  
    $this->layout = 'simple';  
  
    if(!empty($id) && is_numeric($id)){  
        $task = $this->Task->find('first', array(  
            'conditions' => array('Task.id' => $id)  
        ));  
        if(!empty($task)){  
            $this->set('task', $task);  
            $this->Task->id = $task['Task']['id'];  
  
            if(empty($this->request->data)){  
                $this->request->data = $task;  
            } else:  
                if($this->request->is(array('post', 'put'))){  
                    $data = $this->request->data;  
                    if($this->Task->save($data)){  
                        $this->Session->setFlash(__('Tâche éditée !'));  
                        $this->redirect(array('controller' => 'tasks', 'action' => 'index'));  
                    }  
                }  
            endif;  
        } else:  
            throw new NotFoundException("Cette tâche n'existe pas !");  
        } else:  
            throw new NotFoundException('Identifiant incorrect');  
        }  
}
```

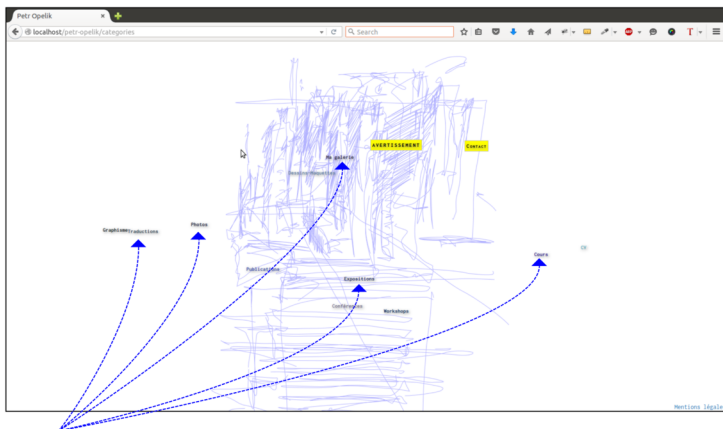
## CONTROLLER

### VUE (edit)



# Petr Opelik - Catégories et projets

Les catégories peuvent contenir des catégories (représentation intervalaire) et des projets.



Les catégories sont disposées aléatoirement sur la page.

```
$.fn.takePlace=function(){  
  
    var randomTop = getRandom(200,$(window).height()-200);  
    var randomLeft = getRandom(200,$(window).width()-200);  
  
    var width = parseInt($(this).width());  
    var height = parseInt($(this).height());  
  
    var top = parseInt($(this).css('top',randomTop+'px'));  
    var left = parseInt($(this).css('left',randomLeft+'px'));  
  
    $(this).nextAll('.sub').css('top',randomTop+'px');  
    $(this).nextAll('.sub').css('left',randomLeft+'px');  
  
};
```

Jquery

```
// On renvoie un nombre aléatoire entre une valeur min (include)  
// et une valeur max (exclue)  
function getRandom(min, max) {  
    return Math.floor(Math.random() * (max - min) + min);  
}
```

# Petr Opelík - Gestion de l'arborescence via le back office

Vue administrateur. Les catégories sont affichées en gras.

VUE



```
echo '<li>';
echo $this->Html->link('Ajouter une catégorie',array(
    'controller'=>'categories',
    'action'=>'addtoTree',
    'Main',
    0,
    0
),
    array('class'=>'link-edit','escape'=>false)
);
echo '</li>';

echo '<li>';
$span = '<span class="glyphicon glyphicon-plus"></span>';
echo $this->Html->link($span, 'Ajouter un projet',array(
    'controller'=>'projects',
    'action'=>'addtoTree',
    'Main',
    0
),
    array('class'=>'link-edit','escape'=>false)
);
echo '</li>';
```

```
foreach($categories as $category){

    //Affiche le titre de la catégorie
    echo '<li class="main-root sz 1-2 bold mg-lt-' . $margin_left . '>';
    echo '</li>';

    if(!empty($sadmin) && $sadmin == true):

        $next_margin_left = $margin_left+20;

        //Ajouter une catégorie
        echo '<li class="mg-lt-' . $next_margin_left . '>';
        echo '</li>';

        //Ajouter un projet
        echo '<li class="mg-lt-' . $next_margin_left . '>';
        echo '</li>';

        //Liens vers les projets qui appartiennent à la catégorie
        foreach($category['Project'] as $project):

            if($project['enable']==0){
                $disable_class = ' disable';
            }else{
                $disable_class = '';
            }

            echo '<li class="mg-lt-' . $next_margin_left . $disable_class . '>';
            if(empty($project['name'])){
                $project['name'] = 'Projet sans titre';
            }
            echo $this->Html->link($project['name'],array(
                'controller'=>'projects',
                'action'=>'view',
                $project['id']
            ),
                array('class'=>'no-style','escape'=>false)
            );
            echo '</li>';

        endforeach;

    endif;

    if(is_array($category['children']) && !empty($category['children'])){
        $margin_left_next = $margin_left+20;
        $this->Tree->recursive($category['children'],$margin_left_next,$sadmin);
    }
}
```