

Rapport de Stage BTS SIO option SLAM

Du 13 mars 2017 au 21 avril 2017



Charlotte David

Greta Vannes

Année 2016-2017

SOMMAIRE

INTRODUCTION.....	PAGE 3
1.PRÉSENTATION DE DATA STORY.....	PAGE 4
1.1 Présentation de l'entreprise.....	PAGE 4
1.2 Descriptif du projet DS Studio.....	PAGE 5
2.PRÉSENTATION DU TRAVAIL EFFECTUÉ.....	PAGE 7
2.1 Projet Personnel.....	PAGE 7
2.2 Outils et méthode de travail.....	PAGE 8
3.CONCLUSION.....	PAGE 20
ANNEXE 1 - FONCTION SAVE_PIXABAY().....	PAGE 21
ANNEXE 2 - MÉMOS.....	PAGE 24
ANNEXE 3 - CONSTRUISEZ DES PRÉSENTATIONS EFFICACES.....	PAGE 26

INTRODUCTION

Dans le cadre de ma formation en BTS **S**ervices **I**nformatique aux **O**rganisations option **S**olutions **L**ogicielles et **A**pplications **M**étiers, j'ai effectué un stage au sein de l'entreprise Data Story durant 6 semaines du 13 mars 2017 au 21 avril 2017. Cette dernière aide à présenter des données de manière plus efficace avec de la data visualisation interactive.

Le projet principal qui m'a été confié fût l'intégration d'une bibliothèque d'image issue du site web <https://pixabay.com> qui propose des images et des vidéos libres de droits et gratuites ; un éditeur de photo , dans l'application Web DS Studio qui permet à l'utilisateur de créer ses propres présentations de données.

L'objectif de cette intégration est de proposer à l'utilisateur, directement et rapidement, des images en haute résolution qu'il pourra également modifier à l'aide de l'éditeur.

Dans un premier temps, je présenterai l'entreprise, puis j'expliquerai en quoi consistait mon travail, les outils utilisés, les difficultés rencontrées et la méthodologie employée. Je finirai par une conclusion en développant mon point de vue sur cette expérience professionnelle et ce que cela m'a apporté.

1.PRÉSENTATION DE DATA STORY

1.1 Présentation de l'entreprise

L'entreprise Data Story est une société Vannetaise spécialisée dans la data visualisation.

La datavisualisation est la manière de représenter des données de façon visuelle. Pour rendre les données plus lisible et compréhensible, on utilise des graphiques, des diagrammes, des images, des créations graphiques..etc

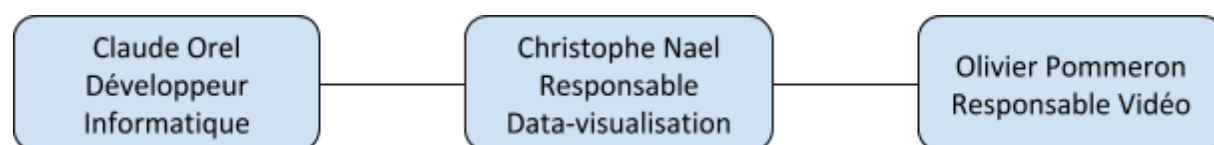
L'entreprise a été fondée en 2014 par trois associés qui se sont rencontrés lors d'une formation d'infographie au Greta de Vannes. Ils sont actuellement localisés au **Parc d'Innovation de Bretagne Sud (PIBS)**.

Elle propose deux offres différentes pour amener les entreprises à présenter leurs données d'une nouvelle façon.

La première offre s'appelle *DS Studio*, c'est une plateforme web basée sur un modèle économique en mode SaaS (Software as a Service). Ce qui signifie que le logiciel est installé sur des serveurs distants et non sur la machine de l'utilisateur.

Elle est composée d'un outil web permettant la création de présentation qui permet d'améliorer la visualisation des données d'une manière simple et efficace.

La seconde offre se nomme *DS Motion*, elle propose la réalisation de contenus animés audiovisuels de courte durée.

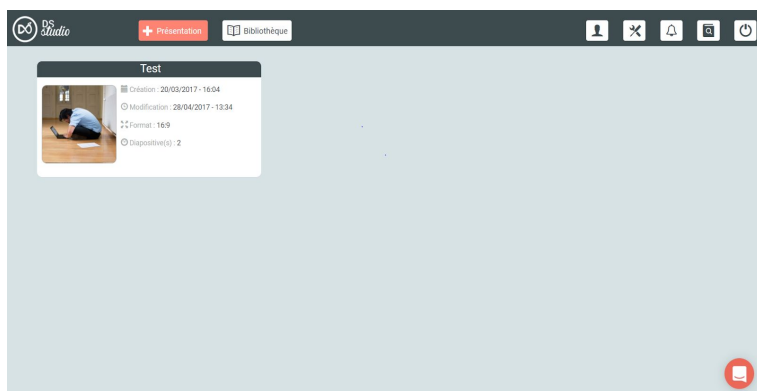


1.2 Descriptif du projet DS Studio

Un constat : lors d'une présentation, l'auditeur se concentre seulement cinq secondes par diapositive ou "slide".

De ce fait, DS Studio a pour objectif d'aider les organisations / entreprises à améliorer leur présentation de données chiffrées grâce à des outils de visualisations et de présentation de l'information.

DS Studio est accessible via la souscription d'un abonnement directement à partir du site internet <https://www.datastory.fr>. Cette application est axée autour de l'optimisation de la représentation graphique de l'information dans l'objectif de maximiser l'impact des données lors de leur présentation.



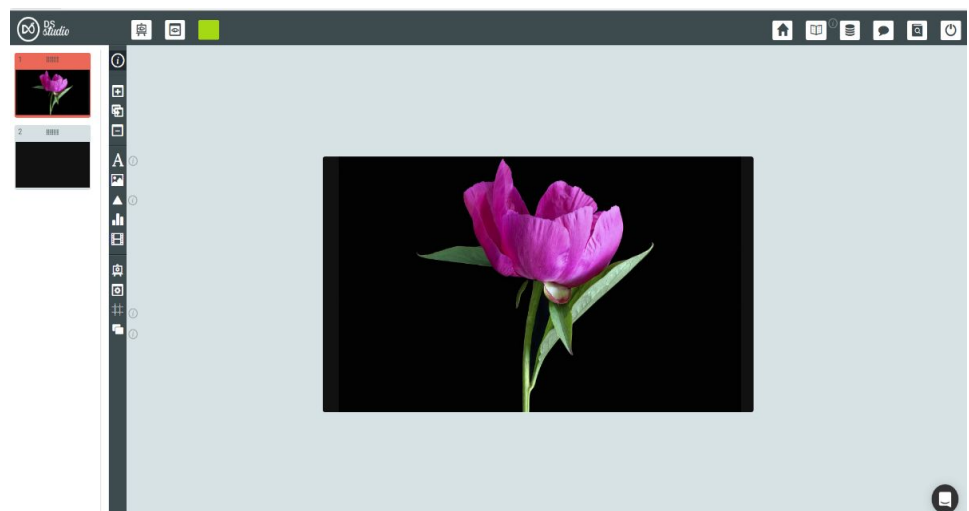
Écran d'accueil Ds Studio

L'écran d'accueil propose :

- Création d'une présentation
- Accès à sa bibliothèque
- Accès aux paramètres de son compte
- Accès aux outils
- Accès aux Notifications
- Accès à un module de formation
- Déconnexion du compte
- Tchat d'aide avec l'équipe Ds Studio

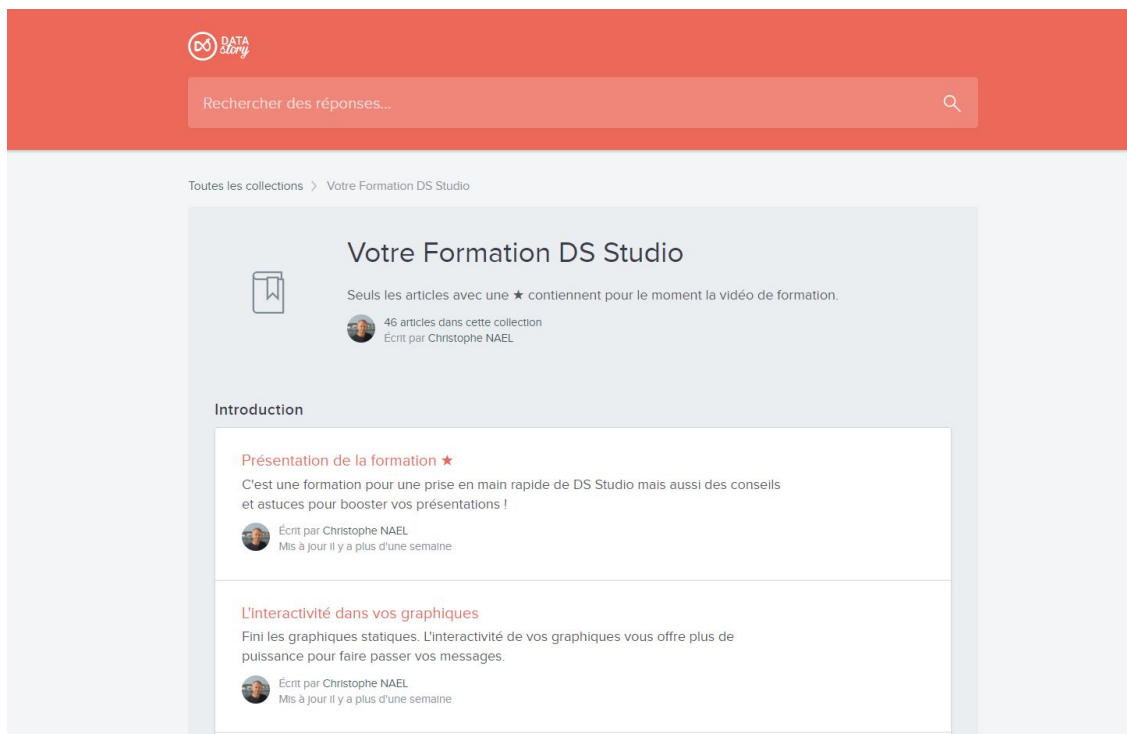
Outils disponibles:

- Insérer du texte
- Insérer une image
- Insérer une forme
- Insérer un graphique
- Insérer une vidéo



L'application Web DS Studio a voulu garder les standards d'ergonomie et de visuel par rapport à d'autres outils de création de présentation (ex : Powerpoint®), pour faciliter la prise en main du logiciel.

Une de leurs différences se situe dans l'envie de former l'utilisateur à la data-visualisation avec leur module de formation qui regroupe des tutoriels vidéos et des articles.



Ce module est accessible dans l'application Ds Studio, il est en cours d'élaboration. Ce sont de courtes vidéos (3 minutes maximum) qui guident l'utilisateur sur le fonctionnement de l'application.

DataStory a également créé un blog qui regroupe des articles expliquant les principes de bases de la data-visualisation : <https://www.datastory.fr/blog>. Cf annexe 3 l'article : "Construisez des présentations efficaces !" rédigé par Christophe Nael.

2. PRÉSENTATION DU TRAVAIL EFFECTUÉ

2.1 Projet Personnel

Mon Objectif de stage était d'intégrer la bibliothèque D'image Pixabay et un éditeur photo : Adobe Creative Cloud pour l'application Web DS studio.

Pixabay est un site internet qui propose plus de 900 000 images (photos, vecteurs, illustrations) et vidéos en haute qualité. Elles sont libres de droits , c'est à dire qu'on peut les utiliser gratuitement à condition d'indiquer qu'elles proviennent de pixabay.

Pixabay propose un API , une **A**pplication **P**rogramming **I**nterface ou Interface Applicative de Programmation, qui permet d'échanger les données entre le site pixabay et l'application Web Studio. En résumé, Pixabay propose aux développeurs d'ajouter une de leurs fonctionnalités (recherche d'images dans leur bibliothèque) à leur application ou site Web.

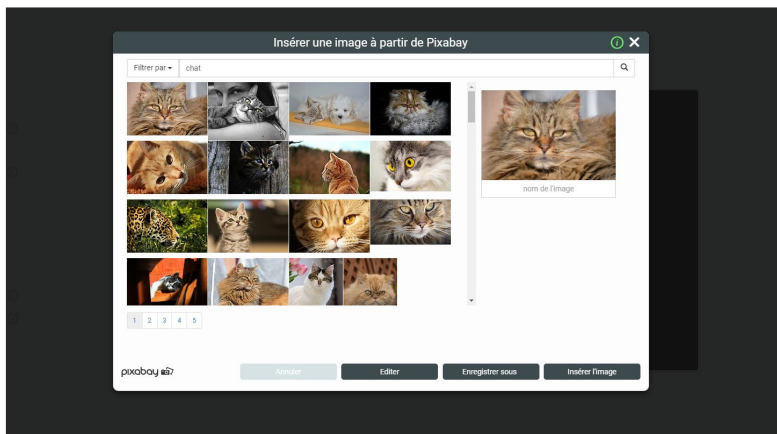
Pour l'éditeur photo , l'équipe de Ds Studio a choisi d'utiliser le SDK adobe creative cloud. Un SDK est un Kit de développement de logiciel. C'est à dire que Adobe fournit son éditeur d'image aux développeurs ainsi que les outils pour le modifier.

Utilité du rajout de cette fonctionnalité :

Actuellement les utilisateurs peuvent importer des images de leur ordinateur à la « bibliothèque » de DS Studio. S'ils veulent modifier leurs images, ils doivent le faire de leur côté.

Avantages :

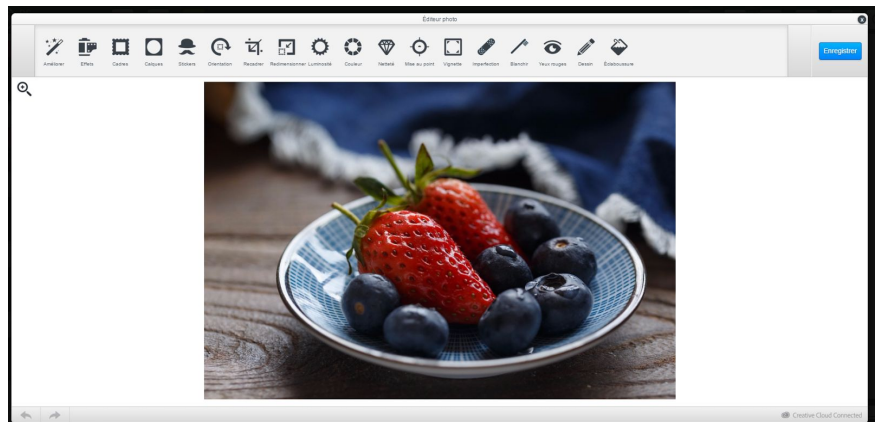
- Gain de temps
- Facilité d'emploi
- Images en haute résolutions
- Éditable directement dans l'application Web
- Grande quantité d'images proposées



Fonctionnalités de la fenêtre Pixabay :

- Chercher une image en filtrant par type (photos, illustrations, images vectorielles)
- 100 résultats par page jusqu'à 500 images proposées
- Editer l'image
- Sauvegarder l'image
- Insérer l'image sur la diapositive ou "slide"

L'éditeur d'image Adobe Creative propose une multitude d'outils pour modifier son image : redimensionner, rogner, contraster, pivoter..etc



2.2 Outils et méthode de travail

Outils et langages utilisés :

Environnement de Développement ou IDE / Éditeur de Texte : Sublime Text 2

Langages Utilisés : HTML 5, CSS 3, PHP, JavaScript.

Framework : JQuery.

Logiciel de version décentralisé : GIT

Méthodologie employée :

En début de stage, j'ai pris du temps pour me repérer dans le code et comprendre comment il était imbriqué.

JavaScript étant un langage inconnu pour moi, j'ai dû me mettre à niveau à l'aide de cours sur internet et de mon tuteur.

Tout au long du stage j'ai fait de la mise en forme, cela m'a permis d'acquérir de nombreuses connaissances, notamment en terme de propriétés CSS.

Pendant le stage, c'est le concept AJAX que j'ai le plus utilisé. C'est pourquoi je vais développer cette partie.

En effet, je l'ai utilisé pour l'API Pixabay ainsi que l'éditeur lors de la sauvegarde des images et des modifications faites par l'éditeur.

Ajax est l'acronyme d'*Asynchronous JavaScript and XML*, ce qui, transcrit en français, signifie « JavaScript et XML asynchrones ».

Intérêts de AJAX :

- Mettre à jour une page Web sans recharger la page
- Demander des données à partir d'un serveur - après le chargement de la page
- Recevoir des données d'un serveur - après le chargement de la page
- Envoyer les données à un serveur - en arrière-plan

La première étape fût de récupérer les images pixabay correspondant à la recherche de l'utilisateur.

```
function pixa_search(type) {  
1.     var requete= $("#searchInput").val(), /*permet de récupérer la requête du champ input dans la page  
html */  
2.     page = sessionStorage.getItem('page'),  
3.     langue = (localStorage.getItem('user_lang')).substring(0,2), /*récupère la langue utilisée par l  
e navigateur du client*/  
4.     data = "key=873858-80b16b1d58bf59fa0991afd18&safesearch=true&per_page=100&response_group=high_re  
solution&q="+requete+"&lang="+langue+"&image_type="+type+"&page="+page;  
5.     $.ajax({  
6.         type: 'GET',  
7.         url: 'https://pixabay.com/api/',  
8.         data: data,  
9.         dataType: "json",  
10.        success: function(json) {  
11.            var test = "";  
12.            sessionStorage.setItem('result', JSON.stringify(json.hits)); /*permet de transformer l'objet j  
son en string*/  
13.            $.each(json.hits, function(index, val) {  
14.                test += "<li class='item' data-w='450' data-h='300'><img class='img_pixa' src='"+val.preview  
URL+"' data-webUrl='"+val.largeImageURL+"' data-id='"+val.id_hash+"'></li>"; /* récupère les valeurs nécessair  
es*/  
15.            });  
16.            $("#formulaire_pixa .data_pixa").html(test);  
17.            // affiche les résultats de la recherche dans le panneau pixabay  
18.            img_pixa_info(); /* appel de la fonction qui retourne les infos pour une image cliquée*/  
19.        }  
20.    });  
21. }
```

function pixa_search()

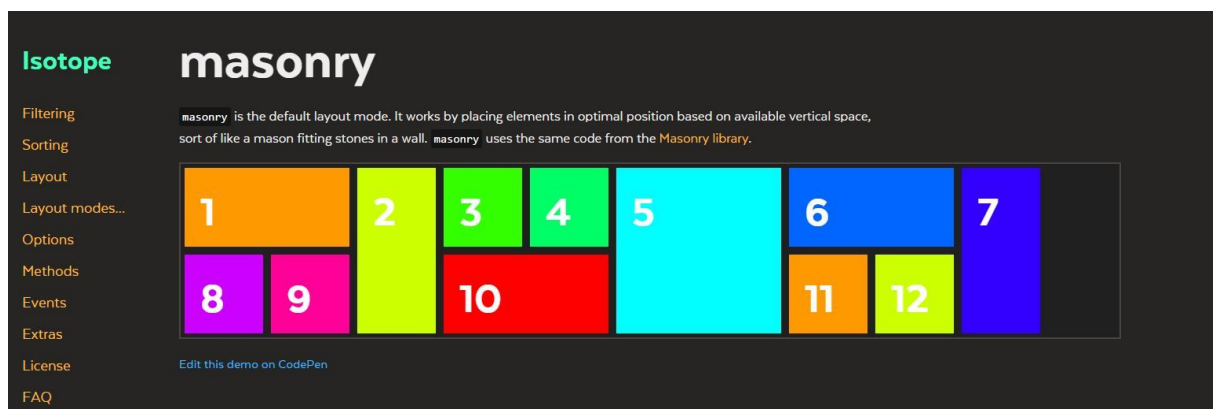
La documentation de l'api pixabay <https://pixabay.com/api/docs> est assez complète et propose de personnaliser les recherches. On peut par exemple, choisir le nombre d'images qui s'affichent avec "per_page", nous avons choisi 100 images par page. Nous avons également activé "safesearch" qui permet d'éviter l'affichage d'images compromettantes.

La fonction `pixa_search()` permet d'envoyer la requête/recherche de l'utilisateur à Pixabay sous forme d'objet JSON qui sera ensuite transformé en texte. JSON (*JavaScript Object Notation*) est un format de données textuel qui permet de représenter des informations structurées. Voici un exemple de données au format JSON :

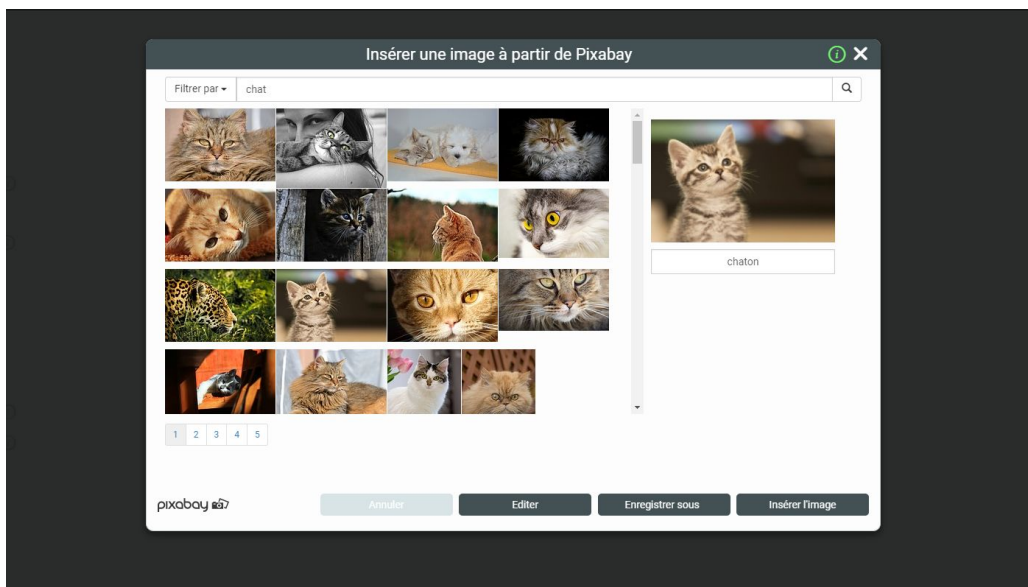
```
1 {
2   "total": 4692
3   "totalHits": 500,
4   "hits": [
5     {
6       "id": 195893,
7       "pageURL": "https://pixabay.com/en/blossom-bloom-flower-yellow-close-195893/",
8       "type": "photo",
9       "tags": "blossom, bloom, flower",
10      "previewURL": "https://static.pixabay.com/photo/2013/10/15/09/12/flower-195893_150.",
11      "previewWidth": 150,
12      "previewHeight": 84,
13      "webformatURL": "https://pixabay.com/get/35bbf209db8dc9f2fa36746403097ae226b796b9e1",
14      "webformatWidth": 640,
15      "webformatHeight": 360,
16      "imageWidth": 4000,
17      "imageHeight": 2250,
18      "imageSize": 4731420,
19      "views": 7671,
20      "downloads": 6439,
21      "favorites": 1,
22      "likes": 5,
23      "comments": 2,
24      "user_id": 48777,
25      "user": "Josch13",
26      "userImageURL": "https://static.pixabay.com/user/2013/11/05/02-10-23-764_250x250.jp
27    },
28    {
29      "id": 14724,
30      ...
31    },
32    ...
33  ]
34 }
```

Issu de la documentation de l'API PIXABAY

En terme de visuel pour l'affichage des résultats, le développeur voulait se diriger vers le style "masonry" de Isotope®, car c'est ce qu'il utilise habituellement :



Lors de mes recherches et de mes questionnements sur l'API Pixabay, j'ai découvert sur le site qu'il existait un plugin spécialement conçu pour l'affichage des images Pixabay : FlexImages. <https://goodies.pixabay.com/jquery/flex-images/demo.html>



Fenêtre de l'application Web affichant les résultats avec FlexImages

```
$('.data_pixa').flexImages({  
  • container: '.item',  
  • object: 'img',  
  • rowHeight: 140,  
  • truncate : 1  
  • });
```

Après cette recherche, l'utilisateur peut soit sauvegarder l'image, soit l'éditer, soit l'insérer directement sur la « slide ». Dans chacun des cas, l'image doit être sauvegardée.

Lorsque l'utilisateur clique sur une image, il lui attribue un nom. La fonction *img_pixa_info()* permet ceci ainsi que d'attribuer les valeurs nécessaires à la sauvegarde comme l'URL de l'image Pixabay.

```

1.     function img_pixa_info(){
2.         $('#img_pixa').on('click',function(){
3.             var att = $(this).attr('data-id');
4.             var url = $(this).attr('data-webUrl');
5.             var img_list = JSON.parse(sessionStorage.getItem('result'));
6.             $.each(img_list, function(index, data) {
7.                 if (att==img_list[index].id_hash) {
8.                     $('#image_info_pixa').empty();
9.                     $('#image_info_pixa').html("<img class='thumb_pixa' style = 'width:100%;' src='"+data.webformatURL+"'">"<b>");
10.                    $('#image_info_pixa').append("<input type='text' class='form-control' style = 'text-align:center;' id='name_pixa' placeholder='nom de l'image' />");
11.                    $('#image_info_pixa').append("<input type='hidden' value='"+att+" ' id='att_pixa' />");
12.                    $('#image_info_pixa').append("<input type='hidden' value='"+url+" ' id='url_pixa' />");
13.                    return false; /* == break , permet d'afficher seulement les valeurs des images sélectionnés*/
14.                }
15.            });
16.        });

```

Fonction img_pixa_info()

Une fois les image affichées dans la fenêtre pixabay de DS Studio, il a fallu les enregistrer. J'ai utilisé un appel AJAX pour se faire. Fonction *pixa_save()* en annexe 1.

```

1.     var xid = $("#xid").val(); //récupere l'id de l'utilisateur
2.     var url_pixa = $('#url_pixa').val(); // récupère l'url de l'image pixabay
3.     var name_pixa = $("#name_pixa").val(); // récupère le nom de l'image donnée par l'utilisateur
4.     var data_pixa = "&url="+url_pixa+"&name="+name_pixa+"&xid="+xid; // les données envoyées vers
edit,php
5.     $.ajax({
6.         type: 'POST', // POST car on envoie des données
7.         url: "edit.php", // ressource ciblée
8.         data:data_pixa, // envoie les données data_pixa à edit,php
9.         (... )
10.    }).

```

Partie tronquée de l'appel AJAX qui se dirige vers edit,php

```

1.      /*****
2.      SAVE IMG PIXABAY
3.      *****/
4.      if (isset($_POST['url'])){ // s'il existe un url envoyé
5.          $dossier = './file-user/repository/'.strtolower($uid).'/images/Pixabay/';
6.          $chemin = 'file-user/repository/'.strtolower($uid).'/images/Pixabay/';
7.          $url = $_POST['url']; // récupération de l'url de l'image pixabay
8.          $name = $_POST['name']; // récupération du nom de l'image
9.          $extension = '';
10.         if (strpos(strtolower($url), 'jpg')>1){ // si c'est une image.jpg
11.             $extension = '.jpg';
12.         }
13.         if( strpos(strtolower($url), 'jpeg')>1){
14.             $extension = '.jpeg';
15.         }
16.         if( strpos(strtolower($url), 'png')>1 ){
17.             $extension = '.png';
18.         }
19.         if(!is_dir($dossier)){ // si le dossier n'existe pas
20.             mkdir($dossier,0777,true); // on crée le dossier avec les droits d'écriture
21.         }
22.         $mon_image = $chemin.$name.$extension;
23.         copy($url, $mon_image);
24.         echo ($mon_image);

```

permet l'enregistrement de l'image dans la bibliothèque de l'utilisateur - edit.php

L'image est enregistrée dans le répertoire Pixabay, l'utilisateur peut y accéder depuis l'application.

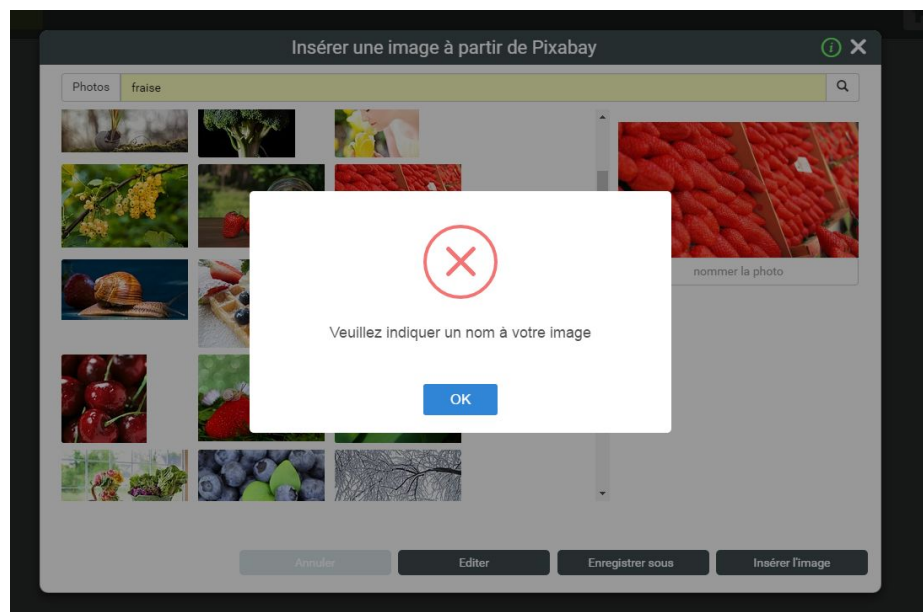
Il a fallu anticiper les erreurs éventuelles de l'utilisateur lors de l'utilisation de cette fonctionnalité :

- pas de nom à l'image
- pas d'image correspondante à la recherche de l'utilisateur
- l'utilisateur clique sur « sauvegarder » alors qu'il n'a pas sélectionné d'image.

Au lieu d'utiliser les « alertes » classiques de Javascript, j'ai utilisé le plug'in SweetAlert (<http://t4t5.github.io/sweetalert>) qui propose des « alertes » responsives et personnalisables.



Alerte classique de Javascript



Alerte du plug'in SweetAlert

Intégration de l'éditeur SDK Adobe Créative.

Pour l'intégration du SDK, voir la documentation fournie par Adobe sur <https://creativesdk.adobe.com/docs/web/#/articles/imageeditorui/index.html>.

Le problème le plus important rencontré fut le fait que l'éditeur devait fermer la fenêtre Pixabay car il y avait un conflit entre Pixabay et l'éditeur, pour pouvoir lancer la fenêtre de l'éditeur. Or l'image sélectionnée par l'utilisateur n'était pas encore enregistrée.

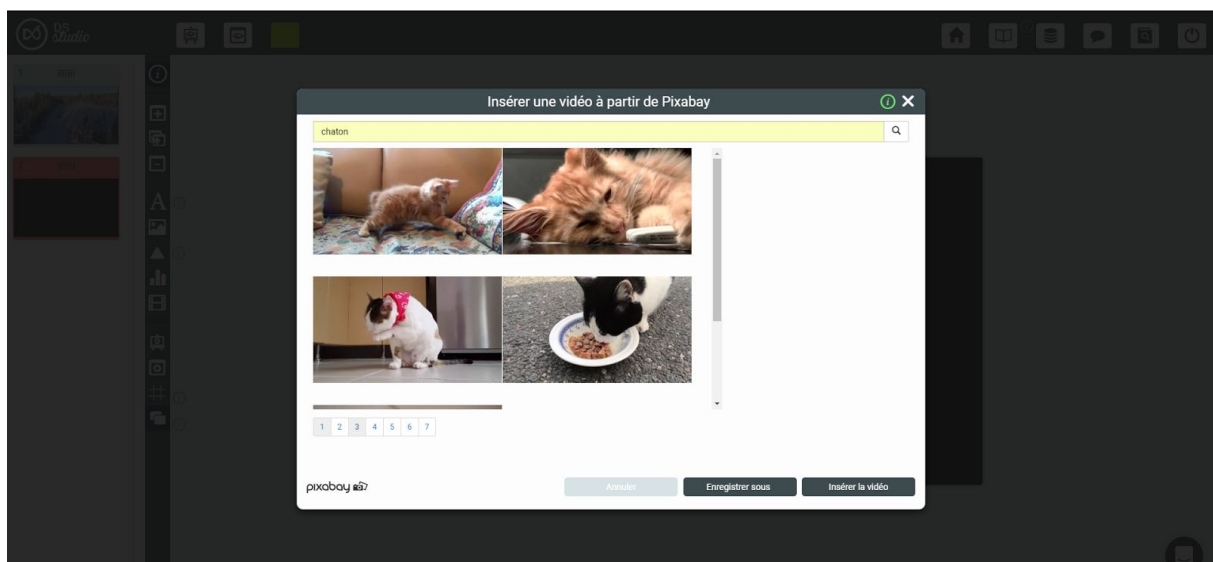
Pour régler ce problème, j'ai utilisé l'Api Web Storage qui permet de conserver des données dans le navigateur.

```
//récupération des données des images pixabay avant la fermeture de la fenêtre
1.      var name_pixabay = $("#name_pixa").val();
2.      localStorage.setItem('name_pixa', name_pixabay);
3.      var name_pixa = localStorage.getItem('name_pixa');
        partie tronquée de la fonction editeur().
```

Une fois l'image éditée, elle est enregistrée et accessible dans la bibliothèque d'image de l'utilisateur, l'originale est également conservée.

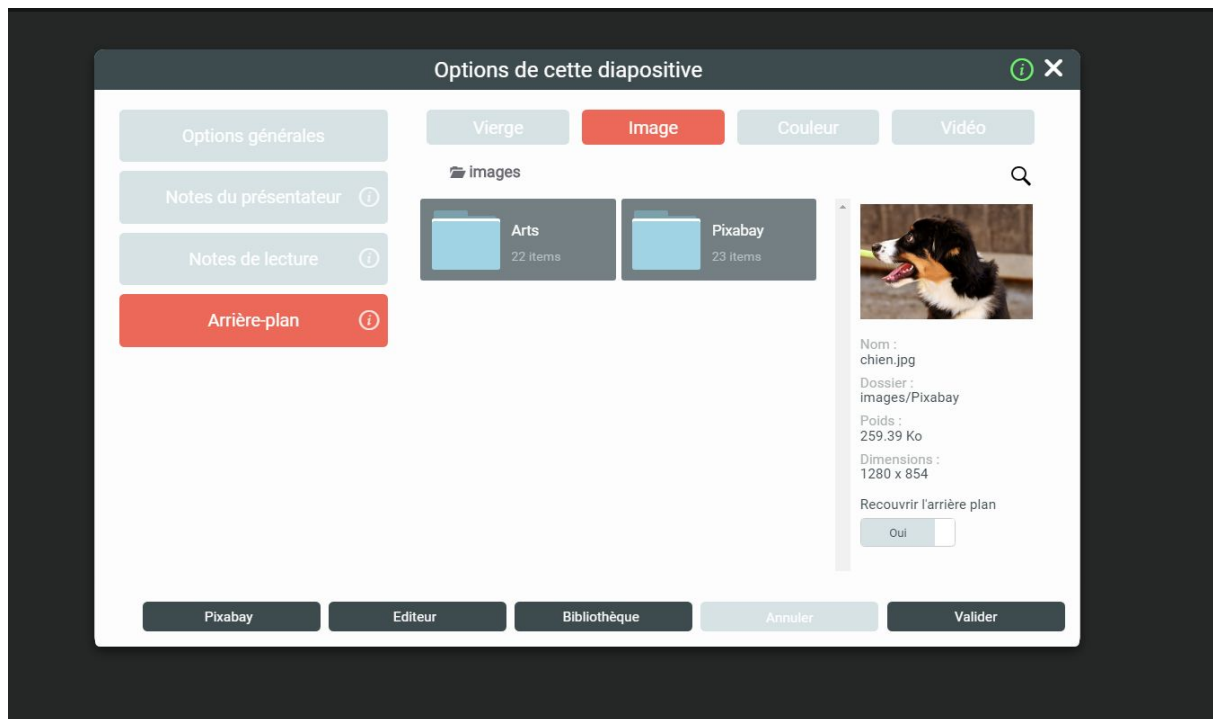
Mon objectif de stage étant atteint à ce stade, mon tuteur m'a donné quelques missions annexes.

J'ai rajouté une fenêtre d'insertion de vidéo depuis Pixabay :



fenêtre des vidéos Pixabay

Enfin, j'ai ajouté la fonctionnalité pixabay et éditeur dans la partie « arrière-plan » des options de la slide . En effet, pour avoir une slide plus percutante , il est recommandé de positionner les images en arriere plan recouvrant toute la slide. Cette fonctionnalité a donc été rajoutée , pour diriger l'utilisateur à faire ce choix ;



Panneau des options de la diapositive

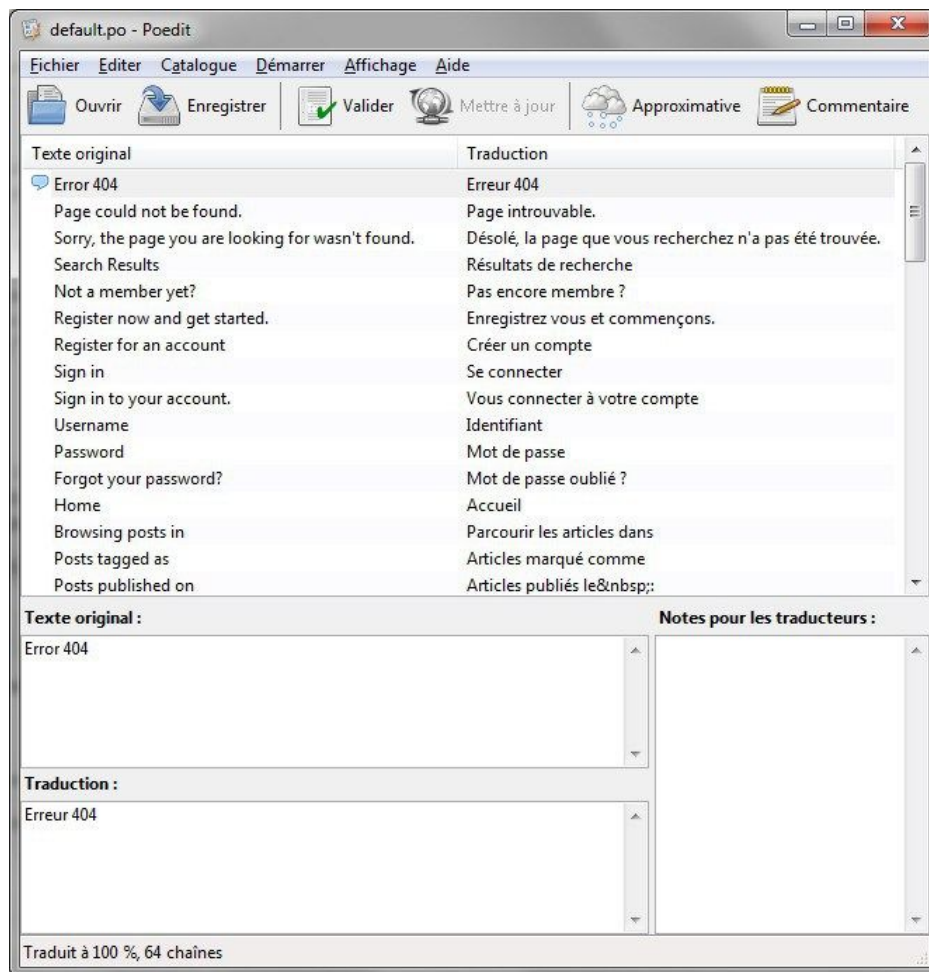
J'ai terminé mon stage par une traduction des textes que j'avais rajouté. Pour cela j'ai utiliser gettext et le logiciel POEdit.

gettext est la bibliothèque logicielle du projet GNU qui sert à l'internationalisation de logiciels. Elle est couramment utilisée pour écrire des programmes multilingues.

POedit est un logiciel libre qui recherche et propose une traduction également modifiable par l'utilisateur et génère des fichier .po qui sont requis pour gettext.

```
01. <li><a data-value="all" ><?php echo _(' Toutes les images ' ) ?></a></li>
```

Dans le fichier Html , on écrit l'intitulé avec un echo .



Interface du logiciel POedit.

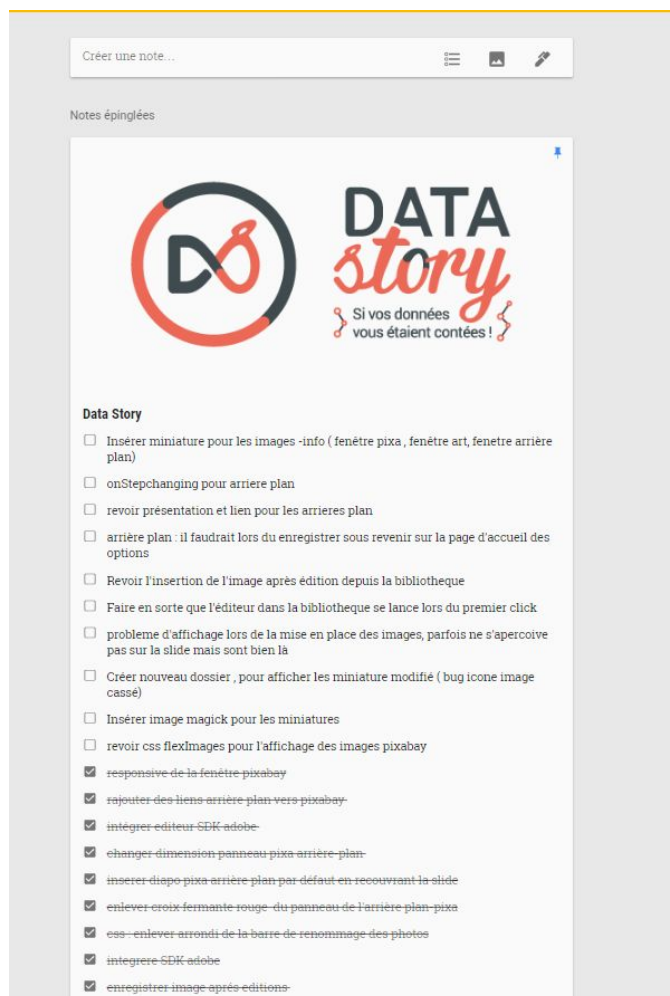
Malheureusement, mon travail durant ce stage n'a pas pu être déployé sur l'application DS Studio, car ils allaient changer de serveur et préféraient attendre le nouveau pour l'intégrer.

L'application *DS Studio* devrait sortir officiellement en mai 2017.

J'avais pris l'habitude de prendre des notes à la main et de faire mes "TO DO LIST" sur papier, ce qui n'était pas une solution pour moi (problème de relecture et perte des données) j'ai choisi de m'aider de Google KEEP comme TO DO List qui garde une trace des travaux effectués.

Cela peut paraître négligeable, mais lors de mon stage, j'ai été très vite confrontée à des bugs que je considérais comme petit, moyen etc. cette To Do List me permettait de les prioriser en plus de ne pas les oublier. De plus, lors de mon travail, il arrivait souvent qu'on discute avec l'équipe et qu'on rajoute des fonctionnalités.

Cette organisation m'a aidé à prioriser mon travail et à ne pas perdre le fil. Voir en Annexe 2 , les mémo de google Keep sous forme de liste.



3.CONCLUSION

Ce deuxième stage a été complètement différent du premier. En effet, je n'étais plus là pour «découvrir» le métier de développeur.

Je pense que j'ai été vraiment mise à ma future place professionnelle.

Mon tuteur m'a laissé autonome dans mon organisation du projet, ce qui fût assez agréable et m'a appris à m'organiser dans mon travail en priorisant les tâches que j'avais à faire.

J'ai apprécié travailler pour une application Web car le front-end et le back-end sont mélangés à part égale.

J'ai apprécié la passion de mon tuteur Christophe Nael et son partage de connaissance en terme de visualisation de données , domaine qui m'était totalement inconnu jusque là.

J'ai également pu constater que le plus long dans le développement n'a pas été l'intégration des fonctionnalités mais la résolution de « petits bugs ».

J'ai aussi vu que développer c'est être tout le temps à faire des modifications, des ajouts ou des suppression de fonctionnalités. Rien n'est figé, et malgré mes réticences du début à « casser » mon code pour intégrer une autre fonctionnalité, cela est nécessaire mais en faisant attention ; ce qui demande de la rigueur.

Je me suis sentie valorisée dans mon travail, car lors de discussion en équipe et des décisions d'ajouter, de supprimer ou de modifier quelque chose, c'était moi qui faisait le changement, de savoir que énormément de choses sont possibles est très satisfaisant. c'est arrivé que rarement je ne puisse pas modifier la fonctionnalité ou le visuel. Cet aspect de mon futur métier est très enrichissant.

J'apprécie également apprendre tout les jours , j'ai pu constater que le développeur était toujours dans cette dynamique. Les langages évoluent , de nouveau plugin apparaissent sans cesse. Cela nécessite , bien sur , une veille constante sur les nouveautés.

ANNEXE 1 - FONCTION SAVE_PIXABAY()

```
function save_pixabay(id,genre){
11.     var button_id =id;
12.     var xid = $("#xid").val();
13.     var att_pixa = $('#att_pixa').val();
14.     var url_pixa = $('#url_pixa').val();
15.     var name_pixa = $("#name_pixa").val();
16.     var data_pixa = "att="+att_pixa+"&url="+url_pixa+"&name="+name_pixa+"&xid="+xid;
17.     if(url_pixa === undefined){
18.         swal({
19.             title: __("Veuillez sélectionner une image ! "),
20.             type: "error",
21.             confirmButtonText: __("C'est parti !")
22.         });
23.     }
24.     else{
25.         if(name_pixa !== ''){
26.             $.ajax({
27.                 type: 'POST',
28.                 url: "edit.php",
29.                 data:data_pixa,
30.                 success: function(data){
31.                     if(data.substring(0,9) != "file-user"){
32.                         swal({
33.                             title: __("Pixabay est indisponible"),
34.                             type: "error",
35.                         });
36.                     }
37.                     else{
38.                         if(button_id == 'save_pixa' ){
39.                             console.log("passage save pixabay");
40.                             swal({
41.                                 title: __("Image enregistrée"),
42.                                 type: "success",
43.                             });
44.                         }
45.                         if (button_id == 'btedit-image_pixa'){
46.                             //if( genre == 'editeur'){
47.                             var chemin = "https://sandbox.datastory.fr/" + data;
48.                             $("#editable").attr("src",chemin);
49.                             editeur(chemin,'btedit-image_pixa');
50.                             //}
51.                         }
                    }
                }
            });
        }
    }
}
```

```

52.         if (button_id == 'btsave-image' ){
53.             console.log("passage inserer");
54.             if ( genre == 'normal' ){
55.                 var i = 'e1=0&e2=0&e3=300&e4=200&e5=image&adding=1&content='+data+'&xid=' + xid,
56.
57.                 u = xid.substr(20);
58.                 $.ajax({
59.                     url: "edit.php",
60.                     type: "POST",
61.                     iframe: !0,
62.                     data: i,
63.                     dataType: "json",
64.                     success: function (t) {
65.                         console.log(data);
66.                         var i = t[0],
67.                         z = $("#xid").val().substr(0, 10)+"_"+i,
68.                         v = $("#xid").val().substr(0, 10),
69.                         m = {id:m,x:0,y:0,l:300,h:200,t:"image",p:"",c:data.substr(21),z:t[1],o:""}
70.                         ,
71.                         a = '<img id="elem' + i + '" data-id="' + i + '" class="element image " dat
72.                         a-type="image" style="left:0;top:30px;width:300px;height:200px;z-index:' + t[1] + '"></div
73.                         >';
74.                         var history = ((sessionStorage.getItem('action_history') === null)?[]:JSON.
75.                         parse(sessionStorage.getItem('action_history')));
76.                         if (history.length >= 5){
77.                             history.splice(0,1);
78.                             history.push("add|"+z);
79.                             sessionStorage.setItem('action_history', JSON.stringify(history));
80.                             $("#bloc_slide .reveal").append(a);
81.                             init_elem();
82.                             sessionStorage.setItem(z,JSON.stringify(m));
83.                             gen_prev_canvas(".slide.active .prev","#phantom",v);
84.                             setTimeout($.fn.colorbox.close(),1000);
85.                             dsMonitor(v, null);
86.                             pusher_event("update-slide",(xid.substr(0, 10)) );
87.                             load_slide(xid.substr(0, 10));
88.                             swal({
89.                                 title: __("Image insérée"),
90.                                 type: "success",
91.                                 });
92.                             }
93.                         }
94.                     });
95.             } if (genre == 'bg'){
96.                 var p = 'savebg=1&bg='+data+'&xid=' + xid,
97.                 c = xid.substr(20);
98.                 $.ajax({

```

```

94.         url: "edit.php",
95.         type: "POST",
96.         data: p,
97.         success: function (t) {
98.             var image_url = data.substring(21),
99.                 slide_id = $("#xid").val().substring(0, 10),
100.                 supported_images = ["jpg", "peg", "png", "gif"],
101.                 pathLength = data.length,
102.                 ext = (data.substring(pathLength-3, pathLength));
103.             if ( data !== "" && supported_images.indexOf(ext) != -1 ) { //image
104.                 $('#bloc_slide section .bg-video, #phantom .bg-video').remove();
105.                 $('#bloc_slide section, #phantom').css({
106.                     'background-image': 'url(' + data + ')',
107.                     'background-color': 'none'
108.                 });
109.             }
110.             for(i = 0; i < messslides.length; i++) {
111.                 if (messslides[i].slide == slide_id) {
112.                     messslides[i].bg = image_url;
113.                     break;
114.                 }
115.             }
116.             gen_prev_canvas(".slide.active .prev", "#phantom", slide_id);
117.             pusher_event("update-slide", slide_id);
118.             setTimeout($.fn.colorbox.close(), 300);
119.             swal({
120.                 title: __("Image insérée en arrière-plan"),
121.                 text: __("Par défaut elle recouvre la slide"),
122.                 type: "success",
123.             });
124.         }
125.     });
126.     }
127.     }
128.     }
129.     }
130.     });
131. } else {
132.     swal({
133.         text: __("Veuillez indiquer un nom à votre image"),
134.         type: "error",
135.     });
136. }
137. }
138. }

```

ANNEXE 2 - MÉMOS



- responsive de la fenêtre pixabay
- ajouter des liens arrière plan vers pixabay
- Insérer miniature pour les images -info (fenêtre pixa , fenêtre art, fenêtre arrière plan)
- onStepchanging pour arrière plan
- revoir présentation et lien pour les arrieres plan
- intégrer editeur SDK adobe
- changer dimension panneau pixa arrière-plan
- insérer diapo pixa arrière plan par défaut en recouvrant la slide
- arrière plan : il faudrait lors du enregistrer sous revenir sur la page d'accueil des options
- enlever croix fermante rouge du panneau de l'arrière plan-pixa
- css : enlever arrondi de la barre de renommage des photos
- intégrer SDK adobe
- enregistrer image après éditions
- régler problème d'affichage avec l'image de l'éditeur
- enregistrer l'image originale et l'image après éditions (sous la forme nom1)
- rajouter fonction éditer à toutes les images
- problème d'actualisation des photos, se met en arrière plan puis change après
- problème d'écriture dans le SDK adobe
- problème d'icônes , nom qui passe sur les autres dans editeur : changer la font-size
- voir les problèmes d'éditeur sous firefox ou les autres
- modifier mauvaise traduction des icons de SDK creative
- Revoir l'insertion de l'image après édition depuis la bibliothèque
- ajouter alerte lors de la modification des images dans la bibliothèque
- Faire en sorte que l'éditeur dans la bibliothèque se lance lors du premier click : problème lors du chargement de IHM1 , appel de la fonction éditeur
- problème d'affichage lors de la mise en place des images, parfois ne s'aperçoit pas sur la slide mais sont bien là
- Voir bug dans arrière-plan - pixabay - image retouché - recouvrir slide sinon tout marche
- Enregistrer les images modifiée de la bibliothèque avec un 1 derrière
- Créer nouveau dossier , pour afficher les miniature modifié (bug icône image cassé)
- éviter que l'image soit écraser image_1 lors de l'enregistrement
- Rajouter fonction php pour la traduction des items dans les panneaux pixabay
- ouvrir la fenêtre de la bibliothèque après édition
- enlever bouton editer du pixabay des options créer data dans options , et php if dans html
- faire lien avec le bouton editeur dans les options arrière plan
- revoir affichage portrait dans pixabay , sont déformé dans les miniature

- Insérer image magick pour les miniatures
- revoir css flexImages pour l'affichage des images pixabay
- revoir taille des images après editeur
- Créer compteur pour nommage des photo rééditées tout en sauvegardant l'original
- revoir qualité de l'image après édition : max size à 1200 avec SDK adobe
- faire dropdown pour les boutons d'options
- ajouter bouton erreur pour l'éditeur " sélectionnez une image"
- revoir les panneaux qui s'ouvrent à la fin des actions ex: après editeur revenir sur page image...etc
- insérer pixabay vidéo
- changer couleur panneau editeur en gris et le bouton enregistrer
- recherche par défaut des images pixabay en photo
- ajouter alerte si pixabay ne renvoie aucun résultat
- insertion vidéo ne fonctionne pas
- insérer pagination pour résultats pixabay
- savoir dans quel page on est à la pagination pixabay
- agrandir bouton outil dans options
- insérer alerte lorsque pixabay n'est pas disponible
- changer couleur bouton enregistrer de l'éditeur
- nom des fichier éviter espace - fonctions new_csv 4850
- réduire espace css entre pagination et résultat
- css probleme de hauteur avec li et div dans vidéo (elles sont coupées)
- ajouter page pour résultat vidéo
- voir responsive de la fenêtre pixabay
- revoir outils
- descendre le nommage margin

ANNEXE 3 - CONSTRUISEZ DES PRÉSENTATIONS EFFICACES

By: Christophe Nael | 26/Jan/2017 |

Nous nous intéressons à deux mondes en particulier qui sont la data visualisation et la création de présentations efficaces. L'un et l'autre sont en fait très proches car ils sont tous deux au service de l'efficacité visuelle de vos messages. Mais c'est quoi un message efficace ? Eh bien, c'est un message clair présenté avec simplicité pour être compris de tous. Malheureusement, nous voyons encore trop souvent des présentations qui manquent d'impact et qui finissent par perdre leur audience au bout de quelques minutes. Après un précédent article sur [les bonnes pratiques pour construire un graphique efficace](#), voici quelques conseils qui vous aideront lors de la construction de vos prochains supports de présentation.

1) Simplifiez votre message

Il est quasiment impossible de lire un texte abondant sur une diapositive et écouter en même temps un orateur. En fait, notre vitesse d'écoute n'est pas la même que notre vitesse de lecture. Vous en avez sûrement déjà fait l'expérience ! Dans ce cas, vous faites le choix, soit d'écouter sans lire le contenu de la diapositive soit l'inverse. Autrement dit, entre support et orateur, l'un des deux ne sert à rien.

Orateur et support de présentation sont complémentaires. La solution : simplifier votre message en ne retenant que les titres ou les sous-titres, et conserver à l'oral le reste de votre propos que vous aurez pris soin de résumer dans les notes du présentateur de votre outil.

2) Ne remettez aucun support avant la fin

Si vous souhaitez capter l'attention de votre auditoire, il sera plus efficace de remettre ce support de présentation après la réunion. Limitez-vous à un ordre du jour synthétique dans un premier temps ! En effet, éveiller la curiosité de votre auditoire est capital pour maintenir son attention. Si le public a votre support sous les yeux, cela deviendra mission impossible ; le plus grand nombre commencera à lire votre document par la fin.

3) Une idée par slide

Une idée = une diapositive. Cette démarche vous amène à avoir un esprit de synthèse dans la construction de votre présentation. L'idée qui doit vous guider est : quel est le message qui doit être retenu par le public à l'issue de votre présentation ? La réponse à cette

question doit être le fil conducteur pour chacune de vos diapositives.

Pour vous aider, votre message doit se raconter comme une histoire, car nous avons tous été bercés par cette approche depuis l'enfance. Une construction simple autour du contexte, la problématique, la solution et le moyen de la mettre en oeuvre, réussira à répondre à la plupart des situations. Nous parlons là de storytelling.

4) Bannissez les listes à puces !

Dans cet esprit, il vous faut bannir les listes à puces sur vos diapositives. Pourquoi ? Et bien parce que vous ne pouvez pas faire passer de message sans émotion et dans une liste à puces...il n'y en a pas ! Ne retenez que les idées essentielles de votre « traditionnelle » liste à puces et reprenez le principe, une idée = une diapositive. Appuyez cette idée par une image, une citation, une anecdote ! Ayez toujours en tête le message que vous souhaitez faire passer. Présenter vos idées de cette façon ne vous prendra pas plus de temps qu'auparavant et vous donnerez ainsi à votre propos la chance de marquer les esprits !

5) Laissez libre cours à votre inspiration !

Ces listes à puces se retrouvent systématiquement dans la plupart des modèles traditionnels de présentation. Oubliez-les ! Autres classiques du genre : votre logo reproduit au même emplacement sur chaque diapositive, ou encore un bloc de texte sagement rangé à côté d'un bloc image. Tout cela manque cruellement d'originalité. Votre exposé ressemblera à celui du voisin et il vous sera difficile de créer la surprise, un autre ingrédient nécessaire pour rendre votre présentation efficace.

Il est intéressant de disposer de modèles pour vous inspirer, afin de structurer votre présentation mais il est préférable de construire votre diapositive à partir d'un support vierge.

Revenons un instant sur la question du logo. En effet, le faire apparaître sur chaque diapositive n'apporte aucune valeur ajoutée à la compréhension de votre message. Mettez-le une fois pour toutes en début de présentation, cela suffira amplement.

6) Taillez dans le gras

De manière générale, simplifier votre message revient à supprimer le superflu. Alors, fini les hauts ou pieds de pages inutiles ou encore les éléments de décoration graphique sans intérêt. Tout ce qui n'est pas utile à la transmission de votre message doit être retiré. Votre diapositive sera terminée lorsque vous n'aurez plus rien à ôter !

7) Clarté et émotion

Vous l'aurez compris « L'Homme est un animal visuel ». Dans cette recherche à la fois de clarté et d'émotion en ce qui concerne la transmission de vos messages, les images tiennent une place importante.

Pour cela, il faut utiliser des images de qualité sans les déformer. Placez-les en plein écran plutôt que sur une partie réduite de votre diapositive avec un texte suffisamment lisible. Votre message sera plus impactant.

8) Ne pas lésiner sur la taille de votre police

Que signifie un texte lisible justement ? C'est un texte qui doit être écrit dans une police appropriée avec un bon niveau de contraste et avec une taille suffisante, pour pouvoir être lu jusqu'au dernier rang de la salle de réunion.

Il n'y a rien de plus agaçant que de ne pas pouvoir déchiffrer ce qui est écrit sur une diapositive. En règle générale, ce qui est trop petit à l'écran devra être retiré. La capacité à maintenir l'attention de votre auditoire se joue également à ce niveau là. Car si votre public doit faire un effort conséquent pour comprendre les éléments de chacune de vos diapositives, il se découragera très vite.

9) Appuyez-vous sur la Data Visualization pour transmettre vos données chiffrées

La transmission de votre message doit s'appuyer également sur des graphiques. L'idéal est de pouvoir bénéficier d'un outil qui intègre ces mêmes principes d'efficacité à la représentation graphique. C'est pour cela que nous avons développé l'application Web [DS Studio](#). Elle permet en plus l'utilisation de graphiques interactifs qui ajoutera une puissance inédite à vos idées.

10) 5 secondes pour comprendre

Enfin pour conclure, si votre auditoire met plus de CINQ secondes à comprendre le sens de votre diapositive, c'est qu'elle manque de clarté !

Si vous prenez l'habitude d'appliquer ces différents conseils, votre public vous remerciera et vous aurez atteint votre objectif, à savoir faire passer votre message de la façon la plus efficace qui soit.

Afin de vous immerger complètement dans cette manière d'aborder vos prochaines présentations, je vous invite à lire notamment quelques ouvrages tels que *Presentation Zen* de [Garr Reynolds](#), *Musclez vos présentations* d'Olivier Richard ou *Boostez vos présentations avec le storytelling* de Yaël Gabison.

De notre côté, nous avons fait en sorte d'intégrer le plus grand nombre de ces bonnes pratiques dans l'application Web [DS Studio](#), dans l'idée d'en faciliter l'approche aux utilisateurs. Alors, à vous de jouer maintenant !