

# **Suppression de microposts**

Honoré Hounwanou

# **Suppression de microposts**

Honoré Hounwanou

©2015 Honoré Hounwanou

# Table des matières

<b>Introduction</b> . . . . .	<b>1</b>
<b>1. La vue</b> . . . . .	<b>2</b>
1.1 Ajout du lien de suppression . . . . .	2
1.2 Ajout d'une popup de confirmation . . . . .	6
1.2 Téléchargement de la librairie SweetAlert . . . . .	8
1.3 Ajout de la librairie SweetAlert . . . . .	10
<b>2. La logique</b> . . . . .	<b>14</b>
<b>Conclusion</b> . . . . .	<b>16</b>

# Introduction

Mama Miya,

Hello guys, heureux de vous retrouver dans cette nouvelle vidéo des [TEACHERS DU NET<sup>1</sup>](#) :).

Dans ce tutoriel, nous allons donner la possibilité à nos différents utilisateurs de pouvoir supprimer leurs propres microposts.

Croyez-moi, ce sera assez facile. Ne perdons donc plus de temps et laissons la magie s'opérer.



## Attention !

Lorsqu'un code ne peut pas tenir sur une ligne, vous avez un anti-shash qui est rajouté avant de passer à la ligne suivante. Pensez donc à enlever ce anti-slash si vous faites du copier-coller.

---

<sup>1</sup><http://youtube.com/hounwanou1993>

# 1. La vue

Dans ce chapitre, nous allons nous intéresser à la partie vue. Entendez par **vue**, ce que voit le visiteur de votre site Web. Nous allons ajouter un lien de suppression sur lequel devra cliquer l'utilisateur afin de déclencher l'action de suppression.

## 1.1 Ajout du lien de suppression

Ouvrez votre navigateur et rendez vous au niveau de la page de profil. Vous devrez normalement avoir quelque chose qui ressemble à ceci.



Page de profil

Nous devons ajouter un lien de suppression au niveau de chaque micropost comme présenté sur la capture suivante :



### Ajout du lien de suppression

Si vous ouvrez le fichier `views/profile.view.php`, vous verrez que le code permettant d'afficher la liste de nos microposts est le suivant :

```

1 <?php if(count($microposts) != 0): ?>
2     <?php foreach ($microposts as $micropost): ?>
3         <?php include('partials/_micropost.php'); ?>
4     <?php endforeach; ?>
5 <?php else: ?>
6     <p>Cet utilisateur n'a encore rien poste pour le moment...</p>
7 <?php endif; ?>

```

Pour ajouter un lien de suppression au niveau de chacun de nos microposts, il nous faudra donc modifier le fichier `partials/_micropost.php` vu que c'est ce dernier qui est utilisé au niveau de la boucle `foreach`.

Ouvrez donc le fichier `partials/_micropost.php` et rajoutez la ligne :

```

1 <a href="#"><i class="fa fa-trash"></i> Supprimer</a>

```

Le nouveau contenu du fichier `partials/_micropost.php` devrait ressembler à ceci :

```

1 <article class="media status-media">
2     ...
3     <div class="media-body">
4         <h4 class="media-heading"><?= e($user->pseudo); ?></h4>
5         <p>
6             ...
7             <a href="#"><i class="fa fa-trash"></i> Supprimer</a>
8         </p>
9
10        <?= nl2br(replace_links(e($micropost->content))); ?>
11    </div>
12 </article>

```



## Information

`<i class="fa fa-trash"></i>` nous permet d'afficher l'icône corbeille de [fontawesome](http://fontawesome.github.io/Font-Awesome/icon/trash/)<sup>2</sup>.

Pour l'instant si vous cliquez sur ce lien, rien ne devrait se passer étant donné qu'au niveau de l'attribut `href` de notre balise `<a>`, nous n'avons mis qu'un simple dièse.

Remplaçons donc ce dièse (#) par `delete_micropost.php?id=<?= $micropost->id ?>` comme suit :

```

1 ....
2 <a href="delete_micropost.php?id=<?= $micropost->id ?>"><i class="fa fa-trash"><\
3 /i> Supprimer</a>
4 ....

```

Si vous actualiser à présent la page de profil, vous aurez droit à une belle erreur.


---


<sup>2</sup><http://fontawesome.github.io/Font-Awesome/icon/trash/>


**mercuryseries**

⌚ il y a environ 21 jours  Notice: Undefined property: stdClass::\$id in C:\xampp\htdocs\tutos\social\_network\partials\\_micropost.php on line 8 Call Stack #TimeMemoryFunctionLocation 10.0020129800{main}()...\profile.php:0 20.0100212088require( 'C:\xampp\htdocs\tutos\social\_network\views\profile.view.php' )...\profile.php:30 30.0130217096include( 'C:\xampp\htdocs\tutos\social\_network\partials\\_micropost.php' )...\profile.view.php:84 ">  Supprimer

Je suis mercuryseries


**mercuryseries**

⌚ il y a environ 2 mois  Notice: Undefined property: stdClass::\$id in C:\xampp\htdocs\tutos\social\_network\partials\\_micropost.php on line 8

Erreur après la modification de la valeur de l'attribut href

Si vous lisez attentivement le message d'erreur,

```

1  ( ! ) Notice: Undefined property: stdClass::$id in C:\xampp\htdocs\tutos\social_
2  network\partials\_micropost.php on line 8

```

vous comprendrez qu'il nous indique qu'il ne trouve pas de propriété id au niveau de la ligne 8 du fichier `partials\_micropost.php`.

Si vous regarder la ligne 8, vous verrez qu'il s'agit de notre fameux lien de suppression.

```

1  <a href="delete_micropost.php?id=<?= $micropost->id ?>"><i class="fa fa-trash"><\
2  /i> Supprimer</a></p>

```

Le problème vient du fait qu'au niveau du fichier `profile.php`, lorsque nous récupérons les micro-posts au niveau de la base de données, nous ne récupérons que les champs `content`, `created_at`, ce qui fait que le champ `id` n'est pas du tout accessible au niveau de la vue.

Ouvrez donc le fichier `profile.php` et rajoutez le champ `id` à la liste des champs récupérés.

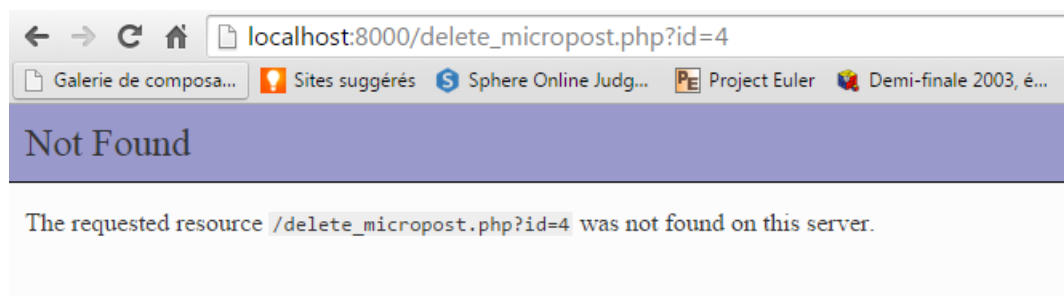


```
1 <?php
2     ...
3
4     if(!$user){
5         redirect('index.php');
6     } else {
7         $q = $db->prepare('SELECT id, content, created_at FROM microposts WHERE \
8 user_id = :user_id ORDER BY created_at DESC');
9         ...
10    }
```

Si vous rechargez la page de profil, vous verrez que l'erreur a maintenant disparu.

Parfait !

Tout ceci est bien beau, mais lorsqu'on clique sur notre fameux lien, nous sommes redirigés vers la page `delete_micropost.php` qui pour l'instant n'existe pas.



Page delete\_post inexistante

Nous allons la créer dans le prochain chapitre et lui ajouter du contenu, mais pour l'instant je vous propose d'ajouter avec un peu de javascript une popup de confirmation.

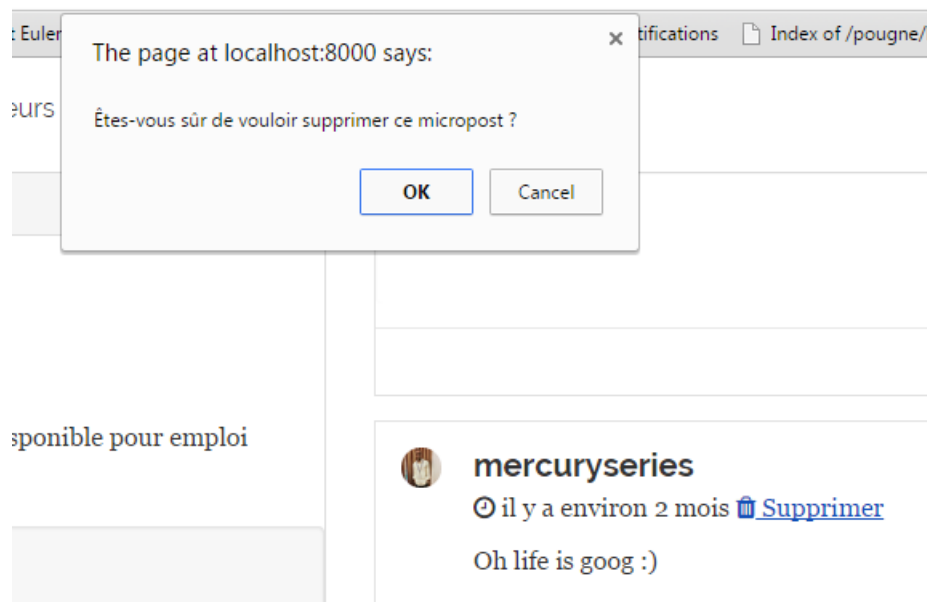
## 1.2 Ajout d'une popup de confirmation

Pour des opérations aussi délicates que la suppression d'un micropost, il faudrait toujours demander une confirmation avant d'effectuer ladite opération. En effet, une fois le micropost supprimé, l'on ne pourra plus faire marche arrière. Il faudrait donc être sûr à 100% que c'est ce que veut notre utilisateur, car il peut arriver qu'il ait cliqué sur le lien de suppression par inadvertance.

Ajouter une confirmation est chose facile en javascript. Il suffit d'utiliser la méthode **confirm** comme suit :

```
1 ...
2 <a onclick="return confirm('Êtes-vous sûr de vouloir supprimer ce micropost ?');\
3 " href="delete_micropost.php?id=<?= $micropost->id ?>"><i class="fa fa-trash"></\
4 i> Supprimer</a>
5 ...
```

Si vous actualisez la page et que vous cliquez de nouveau sur le lien de suppression, vous devrez normalement avoir maintenant une boîte de dialogue de confirmation.



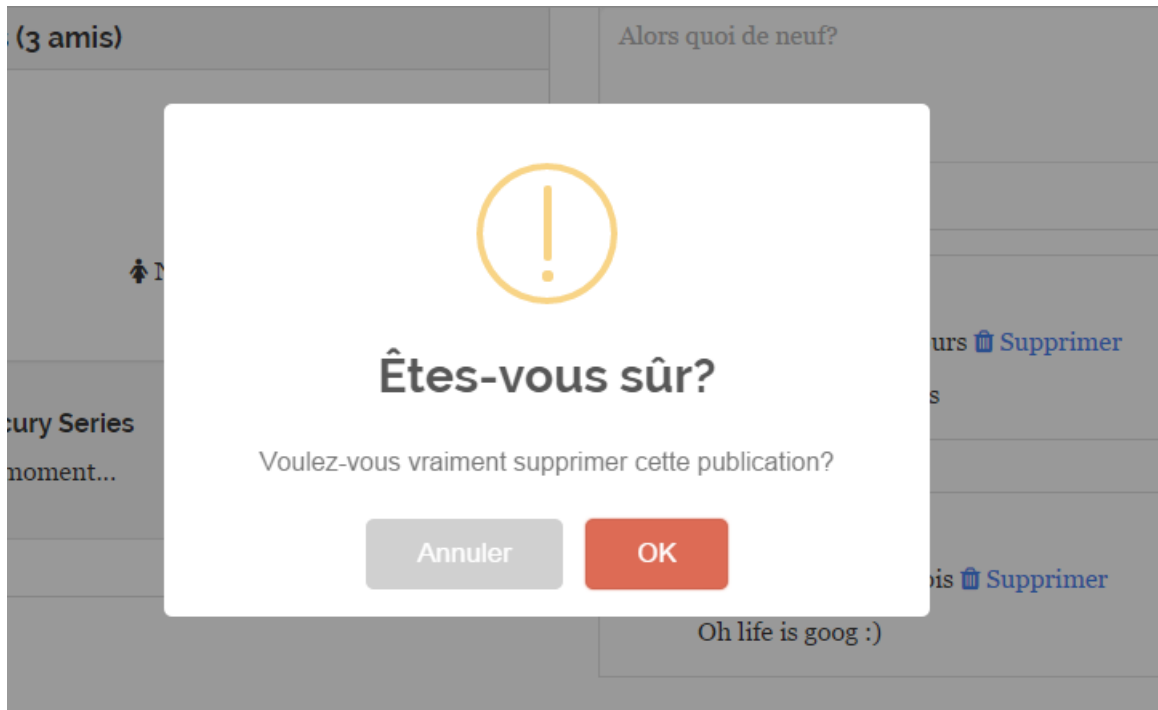
Affichage d'une popup de confirmation

Si vous cliquez sur **Annuler**, rien ne se passera. Par contre, si vous cliquez sur **OK**, vous serez redirigé vers la page `delete_micropost.php`. Cool n'est-ce pas ?

En effet, la méthode `confirm` retourne **true** si l'utilisateur clique sur le bouton **OK** et **false** dans le cas contraire. Nous ne faisons donc que retourner cette valeur booléenne retournée par la méthode `confirm` lorsque l'utilisateur cliquera sur le lien de suppression, ce qui aura pour effet de rediriger l'utilisateur ou non vers la page de suppression de micropost.

Vous savez quoi ? Nous n'allons pas nous arrêter en si bon chemin. Ce que je vous propose, c'est afficher une boîte de dialogue beaucoup plus élégante.

Voyez par vous même :)



Boîte de dialogue de confirmation beaucoup plus élégante

## 1.2 Téléchargement de la librairie SweetAlert

Nous allons ici utiliser la librairie SweetAlert afin de rendre un tant soit peu jolie notre boîte de dialogue de confirmation.

Le site officiel de SweetAlert est <http://t4t5.github.io/sweetalert/><sup>3</sup>.

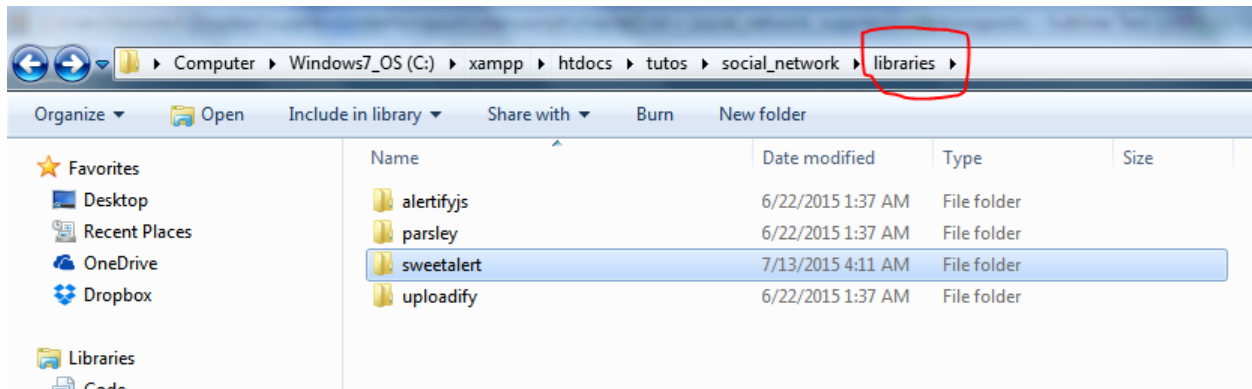
Cliquez sur ce lien suivant <https://github.com/t4t5/sweetalert/archive/master.zip><sup>4</sup> afin de télécharger la dernière version en date de SweetAlert.

Créez dans un premier temps, un sous-dossier **sweetalert** au niveau de notre dossier **libraries**.

---

<sup>3</sup><http://t4t5.github.io/sweetalert/>

<sup>4</sup><https://github.com/t4t5/sweetalert/archive/master.zip>

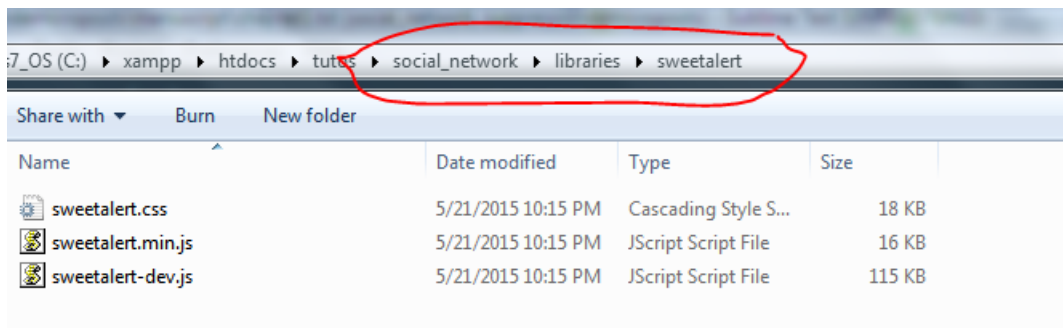


#### Création du sous-dossier sweetalert

Dézipper ensuite le fichier ZIP téléchargé.

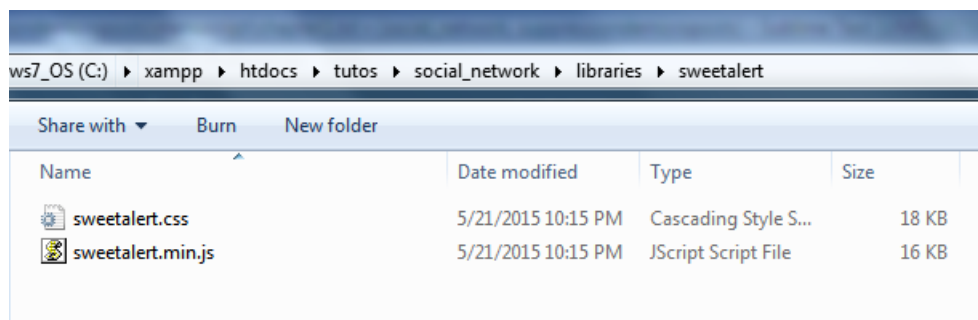
Copiez ensuite tout le contenu de son sous-dossier **dist** au niveau du sous-dossier **sweetalert** que nous avons créé précédemment.

Le contenu de votre sous-dossier **sweetalert** devrait ressembler à ceci :



#### Copie du contenu du sous-dossier dist

Nous allons juste utiliser les fichiers **sweetalert.css** et **sweetalert.min.js**. Vous pouvez donc supprimer le fichier **sweetalert-dev.js** qui est le fichier javascript non minifié.



#### Copie du contenu du sous-dossier dist

## 1.3 Ajout de la librairie SweetAlert

### 1.3.1 Ajout du fichier sweetalert.css

Une fois la librairie SweetAlert téléchargée, il va nous falloir inclure les fichiers *sweetalert.css* et *sweetalert.min.js* au niveau de notre application web.

Ouvrez donc le fichier `partials/_header.php` et rajouter cette ligne :

```
1     ...
2     <link rel="stylesheet" href="libraries/sweetalert/sweetalert.css">
3     ...
4 </head>
```

### 1.3.2 Ajout du fichier sweetalert.min.js

Créez un fichier `main.js` au niveau du dossier `assets/js` si ce n'est pas déjà fait.

Ouvrez ensuite le fichier `views/profile.view.php` et rajouter ces trois lignes tout juste en bas :

```
1 <script src="assets/js/jquery.min.js"></script>
2 <script src="libraries/sweetalert/sweetalert.min.js"></script>
3 <script src="assets/js/main.js"></script>
4 ...
5 </body>
6 </html>
```



#### Attention !

Il faudrait que le fichier `assets/js/main.js` soit chargé après `jquery.min.js` et `sweetalert.min.js` vu qu'il va dépendre de ces deux librairies.

Ouvrez le fichier `assets/js/main.js` nouvellement créé et ajouter lui comme contenu ceci :

```
1 $(document).ready(function(){
2     swal({
3         title: "Êtes-vous sûr?",
4         text: "Voulez-vous vraiment supprimer cette publication ?",
5         type: "warning",
6         showCancelButton: true,
7         cancelButtonText: "Annuler",
8         confirmButtonText: "Oui",
9         confirmButtonColor: "#DD6B55"
10    });
11 });
```

Ce code permet de dire, dès que le DOM est chargé (ou pour faire simple, dès que la page est chargée), il faudra afficher une popup de confirmation SweetAlert ayant :

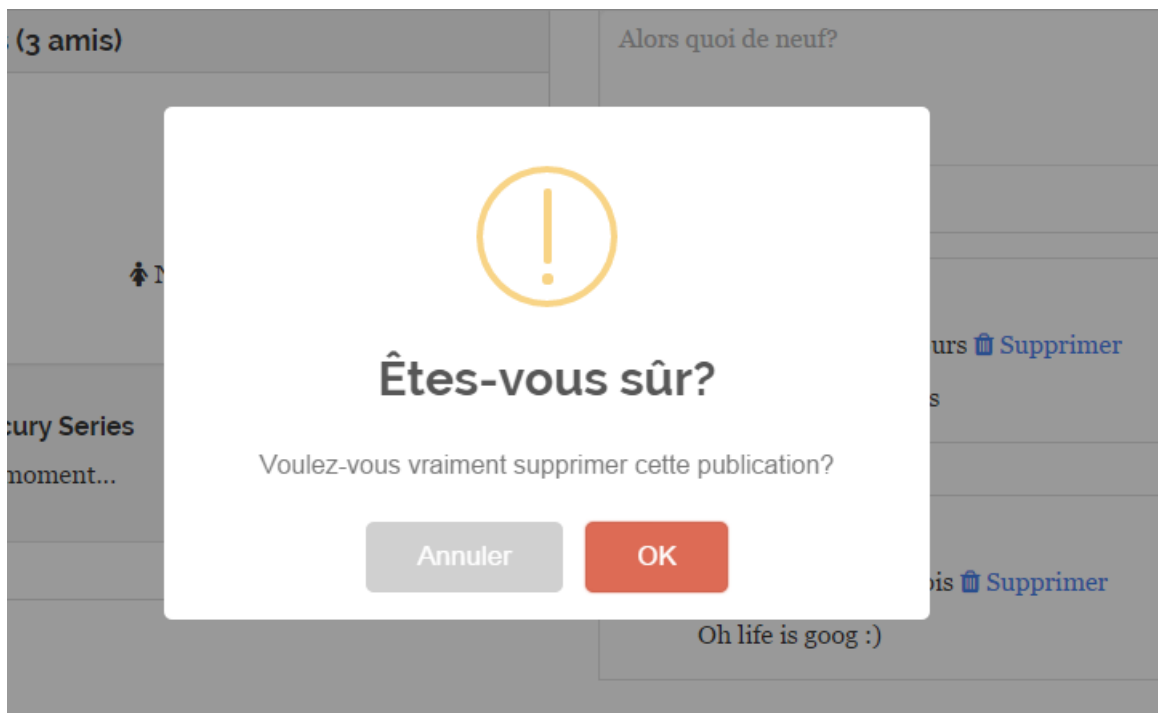
- comme titre : **Êtes-vous sûr ?**.
- Comme text : **Voulez-vous vraiment supprimer cette publication ?**.
- Comme type de popup : **warning**.
- Un bouton d'annulation qui devrait être affiché.
- Comme texte pour le bouton d'annulation : **Annuler**.
- Comme texte pour le bouton de confirmation : **Oui**.
- Comme couleur pour le bouton de confirmation : **#DD6B55**.

Pour plus d'informations, je vous invite à lire la documentation sur le site officiel de [SweetAlert](http://t4t5.github.io/sweetalert/)<sup>5</sup>.

Si vous actualisez à présent la page de profil, vous devez voir cette popup apparaître.

---

<sup>5</sup><http://t4t5.github.io/sweetalert/>



Boîte de dialogue de confirmation

Notre objectif sera de faire en sorte qu'elle puisse apparaître uniquement lorsqu'on cliquera sur un lien de suppression pas dès que la page est chargée.

Pour ce faire, ouvrez le fichier `assets/js/main.js` et remplacer son contenu par :

```
1 $(document).ready(function(){
2     $('[data-confirm]').on('click', function(e){
3         e.preventDefault(); //Annuler l'action par défaut
4
5         //Récupérer la valeur de l'attribut href
6         var href = $(this).attr('href');
7
8         //Récupérer la valeur de l'attribut data-confirm
9         var message = $(this).data('confirm');
10
11         //On aurait pu écrire aussi
12         //var message = $(this).attr('data-confirm');
13
14         //Afficher la popup SweetAlert
15         swal({
16             title: "Êtes-vous sûr?",
17             text: message, //Utiliser la valeur de data-confirm comme text
18             type: "warning",
```

```

19         showCancelButton: true,
20         cancelButtonText: "Annuler",
21         confirmButtonText: "Oui",
22         confirmButtonColor: "#DD6B55"
23     }, function(isConfirm){
24         if(isConfirm) {
25             //Si l'utilisateur clique sur Oui,
26             //Il faudra le rediriger l'utilisateur vers la page
27             //de suppression
28             window.location.href = href;
29         }
30     });
31 });

```

Ce nouveau code que nous venons d'ajouter permet de dire que lorsqu'on cliquera sur n'importe lequel des éléments (un lien, un bouton...) qui aura l'attribut **data-confirm** alors il faudra afficher notre jolie popup de confirmation SweetAlert. Dans le cas où l'utilisateur confirme la suppression, il faudra le rediriger vers la page `delete_micropost.php`.

Ouvrez donc ensuite le fichier `partials/_micropost.php` et remplacer :

```

1 <a onclick="return confirm('Êtes-vous sûr ? de vouloir supprimer ce micropost');\
2 " href="delete_micropost.php?id=<?= $micropost->id ?>"><i class="fa fa-trash"></\
3 i> Supprimer</a></p>

```

par

```

1 <a data-confirm="Voulez-vous vraiment supprimer cette publication ?" href="delet\
2 e_micropost.php?id=<?= $micropost->id ?>"><i class="fa fa-trash"></i> Supprimer<\
3 /a></p>

```

Tout ce que nous faisons, c'est enlever le code javascript précédemment ajouté et rajouter ensuite l'attribut **data-confirm** à notre lien afin que lorsqu'on clique sur ce dernier, la fameuse popup puisse apparaître.

La valeur de l'attribut **data-confirm** fera office de texte pour notre boîte de dialogue.

Ainsi dorénavant si vous souhaitez voir afficher une popup de confirmation lorsqu'on clique sur un élément particulier, il vous suffira d'ajouter l'attribut **data-confirm = "Votre message"** à cet élément et le tour est joué ! Essayer d'ajouter cet attribut où vous le voulez afin de confirmer mes dires.

Ouvrez maintenant votre navigateur et actualiser la page de profil. Tout devrait normalement marcher à la perfection.

Ne croyez-vous pas qu'il serait maintenant temps de supprimer un ou deux microposts :).



## 2. La logique

La suppression du micropost en elle-même n'a vraiment rien de bien compliqué.

Créez le fichier `delete_micropost.php` au niveau de la racine de votre projet, puis coller le code suivant comme contenu :

```
1  <?php
2  session_start();
3
4  require("includes/init.php");
5  include('filters/auth_filter.php');
6
7
8  if(!empty($_GET['id'])){
9      $q = $db->prepare('SELECT user_id FROM microposts WHERE id = :id');
10     $q->execute([
11         'id' => $_GET['id']
12     ]);
13
14     $data = $q->fetch(PDO::FETCH_OBJ);
15     $user_id = $data->user_id;
16
17     if($user_id == get_session('user_id')){
18         $q = $db->prepare('DELETE FROM microposts WHERE id = :id');
19         $q->execute([
20             'id' => $_GET['id']
21         ]);
22         set_flash("Votre publication a été supprimée avec succès!");
23     }
24 }
25
26 redirect('profile.php?id='.get_session('user_id'));
```

- Nous vérifions dans un premier temps que l'utilisateur connecté est bel et bien l'auteur du micropost qu'il souhaite supprimer. Cela va permettre de faire en sorte qu'un utilisateur lambda ne puisse pas supprimer les microposts d'un autre utilisateur juste parcequ'il connaît les identifiants de ces microposts. Remarquez que nous n'affichons aucun message du genre **Vous n'êtes pas autorisé à effectuer cette action**. En effet, moins on en dit, mieux c'est !

- Si l'utilisateur connecté est l'auteur du micropost, nous supprimons le micropost et nous affichons un message flash.
- Vu que dans tous les cas, l'utilisateur doit être redirigé vers sa page de profil, nous ajoutons la ligne de redirection en fin de script.



Avant suppression du micropost



Après suppression du micropost

Et voilà, tout est bien qui finit bien :)

# Conclusion

Houra ! Chacun de nos utilisateurs a maintenant la possibilité de pouvoir supprimer ses microposts. Comme vous avez pu le voir il n'y avait vraiment rien de bien compliqué. Je suis certain que vous aurez pu écrire ce code les yeux bandés.

Si vous avez néanmoins quelques incompréhensions au niveau de certaines parties, n'hésitez surtout pas à m'envoyer un mail, je me ferai un plaisir de vous répondre **[mercuryseries@gmail.com](mailto:mercuryseries@gmail.com)**.

So thank you guys for watching. Don't forget to subscribe and have a nice. Peace !