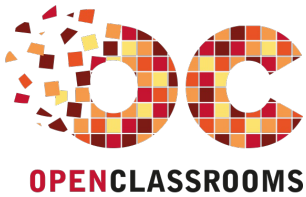
Empêcher le téléchargement direct de fichiers multimédia

Par Christophe Tafani-Dereeper (christophetd)



www.openclassrooms.com

Dernière mise à jour le 11/05/2012

Sommaire

Empêcher le téléchargement direct de fichiers multimédia

Par  Christophe Tafani-Dereeper (christophetd)

Mise à jour : 11/05/2012
Difficulté : Intermédiaire Durée d'étude : 2 heures

Bonjour à tous !

Dans ce tutoriel, je vais présenter une technique permettant de proposer des fichiers multimédias (musique, vidéo...) à la lecture mais en en empêchant le téléchargement direct (par l'URL).
Cela peut vous être utile notamment si vous souhaitez diffuser des musiques que vous avez créées, des vidéos que vous avez filmées, etc.

La méthode présentée n'est pas infaillible : des logiciels dits d'enregistrement de flux peuvent la contourner. Cela dit, il en va de même pour de nombreux sites (y compris YouTube, Dailymotion, Facebook, Grooveshark et j'en passe) et ça reste à mes yeux quelque chose qui vaut le coup d'être mis en place.

Sommaire du tutoriel :

Introduction

Outils utilisés

Dans ce tutoriel, j'utiliserai la bibliothèque PDO pour l'accès à la base de données et le lecteur flash Dewplayer, simple d'utilisation et de mise en place (téléchargement et documentation).

Ce lecteur ne peut lire que les fichiers mp3. Libre à vous d'en utiliser un autre (qu'il soit en flash, en javascript...) ; le principe est le même.

Le problème

Lorsque vous souhaitez rendre un fichier musical disponible à l'écoute, on obtient quelque chose comme ceci :

Code : HTML - index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>Écouter nos créations</title>
 </head>
 <body>
  <!-- Le DewPlayer !-->
  <object type="application/x-shockwave-flash" data="dewplayer-
mini.swf" width="160" height="20" id="dewplayer" name="dewplayer">
```

```
        <param name="wmode" value="transparent" />
        <param name="movie" value="dewplayer-mini.swf" />
        <!-- Le chemin vers le fichier audio à lire est à insérer ici !-
->
        <param name="flashvars" value="mp3=media/musique.mp3" />
      </object>
     </body>
    </html>
```

Bien évidemment, cela fonctionne à merveille. Seulement, n'importe quel internaute peut télécharger le fichier en écoute *via* l'URL indiquée dans le code source.
On pourrait être tenté d'utiliser un htaccess, dans lequel on placerait le code suivant :

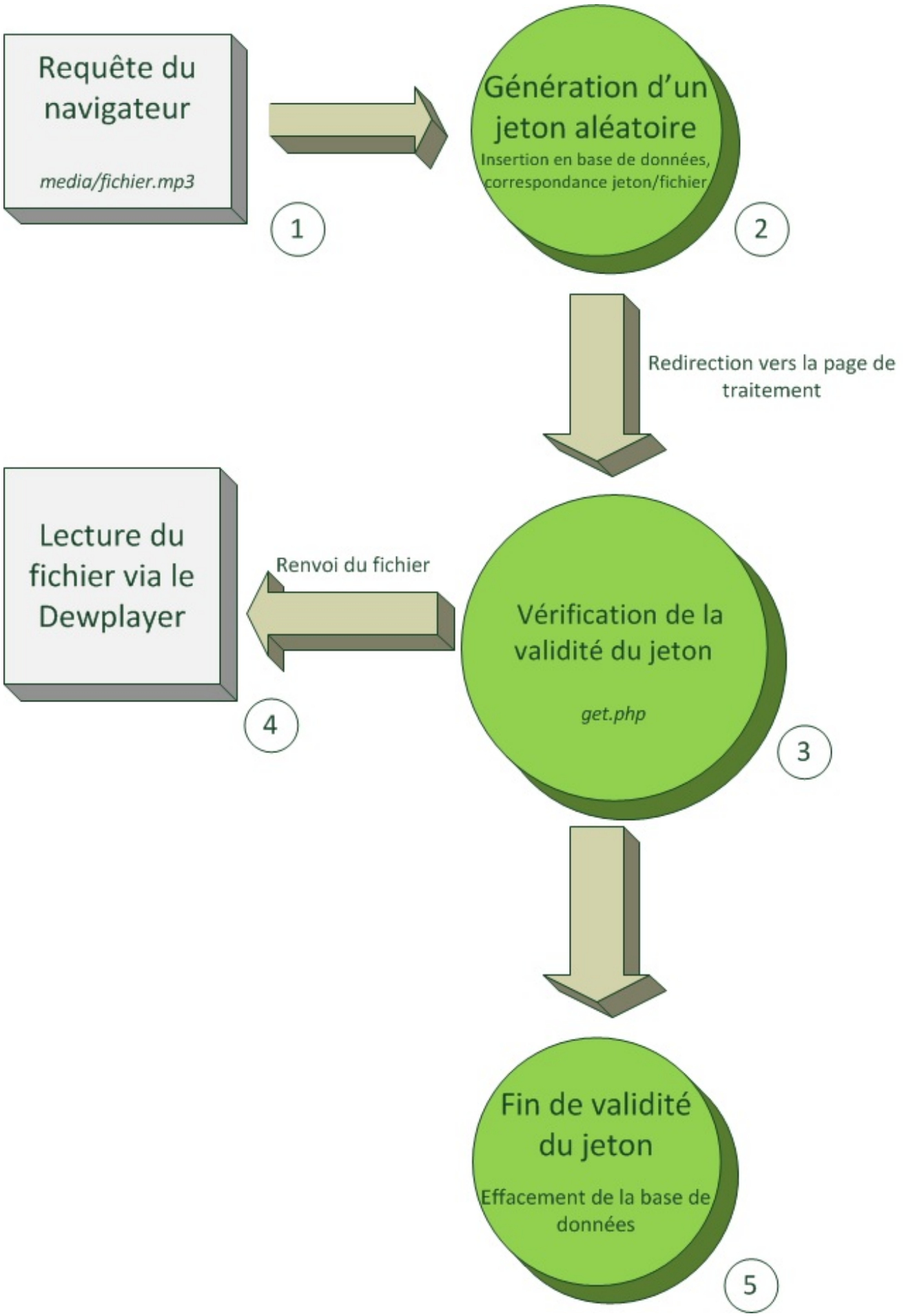
Code : Apache - media/.htaccess

```
# On interdit l'accès au fichier pour tout le monde...
Deny from all
#... sauf pour le serveur lui-même
Allow from 127.0.0.1
```

Le problème est que lorsqu'un client exécute la page, c'est son navigateur, **avec son IP**, qui demande au serveur le fichier *musique.mp3*.

Une solution

La solution que je vous propose est basée sur le principe des *tokens* (*jeton*, en anglais) : en générant un token unique au sein même de la page et en redirigeant les requêtes allant vers les fichiers multimédia vers une page PHP, il nous est possible d'autoriser l'accès au fichier une seule et unique fois, lorsque le navigateur en aura besoin pour le lire.
Petit schéma :



Génération du token

Il nous faut tout d'abord un token unique. Pour cela, je vous propose d'utiliser la fonction uniqid, qui nous en générera un de 23 caractères (lire la doc pour plus d'info) :

Code : PHP - index.php

```php
<?php
// Génération du token
$token = uniqid(rand(), true);
```

Voici maintenant ce que nous allons faire :

- insérer le token généré en base de données, tout en l'associant à un fichier multimédia ;
- demander au dewplayer de lire le fichier.

La table que je vous propose contient deux champs : le token et le fichier auquel il correspond. Je vous donne le SQL, à exécuter dans PhpMyAdmin par exemple.

Secret (cliquez pour afficher)

Code : SQL

```sql
--
-- Table structure for table `media_tokens`
--

CREATE TABLE IF NOT EXISTS `media_tokens` (
  `token` varchar(23) NOT NULL,
  `fichier` text NOT NULL,
  UNIQUE KEY `token` (`token`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Vous remarquerez que j'ai mis le champ *token* en varchar 23 : c'est-à-dire qu'il ne pourra pas contenir plus de 23 caractères. Ce qui est logique, car un hash généré par la fonction uniqid en contient autant.

Pour commencer, on va créer une instance de PDO puis insérer le token précédemment créé dans la base de données :

Code : PHP - index.php

```php
<?php
// Connexion à la base de données
$pdo = new PDO('mysql:dbname=tests;host=localhost', 'christophe',
'thepasswordisalie');

// Insertion du token dans la base de données
$q = $pdo->exec("INSERT INTO media_tokens
SET token='$token', fichier='musique.mp3'");
if($q === FALSE) {
 exit('Une erreur est survenue.');
}
```

Note : j'utilise ici un fichier *musique.mp3*, que je place dans *media/*. Libre à vous de faire votre propre adaptation, bien entendu.

Maintenant, la question se pose : quelle URL donner au lecteur ? C'est là qu'intervient l'URL rewriting (si vous ne savez pas de quoi il s'agit, je vous invite à lire ceci) : nous allons rediriger toutes les URL de la forme **[unToken].mp3** vers la page *get.php*, qui devra recevoir en paramètre le token.

Code : Apache - .htaccess

```
# Activation de l'URL Rewriting
RewriteEngine On

# Règle de réécriture. Vous pouvez ajouter les extensions que vous
souhaitez
RewriteRule ^([a-zA-Z0-9]{23})\.mp3$ get.php?token=$1 [L]

# Si vous voulez gérer plusieurs extensions, vous pouvez utiliser
quelque chose dans le genre :
# RewriteRule ^([a-zA-Z0-9]{23})\.(mp3|ogg|wav|avi|etc)$ get.php?
token=$1 [L]
```

Nous pouvons donc indiquer l'adresse **[tokenGénéré].mp3** au lecteur :

Code : PHP - index.php

```
<!DOCTYPE html>
<html>
    <head>
        <title>Écouter nos créations</title>
 </head>
 <body>
  <!-- Le DewPlayer !-->
  <object type="application/x-shockwave-flash" data="dewplayer-
mini.swf" width="160" height="20" id="dewplayer" name="dewplayer">
   <param name="wmode" value="transparent" />
   <param name="movie" value="dewplayer-mini.swf" />
   <!-- Le chemin vers le fichier audio à lire est à insérer ici !-
->
<param name="flashvars" value="mp3=<?php echo $token ?>.mp3" />
  </object>
 </body>
</html>
```

Lorsque le navigateur chargera la page, le lecteur, quel qu'il soit, fera appel à l'URL que nous avons fait pointer vers *get.php* et exécutera donc le script en lui passant le token généré en paramètre.

Si le token est valide, ça voudra dire que c'est bel et bien notre page de lecture qui essaye d'accéder au fichier.

Envoi du fichier multimédia

Je vous rappelle brièvement ce que le script *get.php* doit faire :

1. vérifier que le token passé en paramètre est valide ;
2. si c'est le cas, envoyer au lecteur le fichier correspondant grâce à la fonction readfile ;
3. **effacer le token de la base de données**, afin qu'il ne soit plus valide et que l'on ne puisse pas accéder au fichier depuis l'URL.

Tout d'abord, on va vérifier que le token a bien été passé en paramètre :

Code : PHP - get.php

```
<?php
// Si le token n'a pas été passé en paramètre
if(!isset($_GET['token'])) {
 exit(); // Arrêt du script
```

```
}
$token = $_GET['token'];
```

Ensuite, nous allons vérifier que le token soit bien dans la base de données. Si c'est le cas, ça voudra dire qu'il est valide et que c'est en effet notre page de lecture qui veut y accéder ; sinon, qu'il ne l'est pas.

Code : PHP - get.php

```
<?php
// Connexion à la base de données
$pdo = new PDO('mysql:dbname=tests;host=localhost', 'christophe',
'thepasswordisalie');

// Préparation de la requête
$q = $pdo->prepare('SELECT fichier FROM media_tokens WHERE
token=:token');
$q->bindValue(':token', $_GET['token']);

// Exécution !
$q->execute();

// Si le token correspond à un enregistré en base de données
if($q->rowCount() == 1) {
 // Actions à exécuter quand le token est valide
}
else {
 // Actions à exécuter quand il n'est pas valide
}
```

J'utilise ici le concept des requêtes préparées, plus lisibles et pratiques (les paramètres de la requête sont automatiquement sécurisés par PDO, pas besoin de s'embêter avec ça). Plus d'infos.

Dans le cas où le token est valide, il va tout d'abord nous falloir récupérer le nom du fichier qui lui est associé :

Code : PHP - get.php

```
<?php
// Si le token correspond à un enregistré en base de données
if($q->rowCount() == 1) {
 // On récupère les résultats de la requête
 $resultat = $q->fetch();
 $fichier = 'media/'.$resultat['fichier']; // Dans le cas où vos
fichiers multimédia se trouvent dans media/
 $extension = end(explode('.', $fichier)); // Nous en aurons besoin
plus tard
}
```

Nous allons maintenant vérifier que le fichier auquel est relié le token existe bien ; si c'est le cas, on pourra supprimer le token de la base de données et envoyer le fichier au lecteur.

Code : PHP - get.php

```
<?php
// Si le fichier existe bien et qu'il est lisible, on peut
l'envoyer au lecteur
if(file_exists($fichier) && is_readable($fichier)) {
 // On peut maintenant effacer le token qui est en base de données
 $pdo->exec("DELETE FROM media_tokens WHERE
token='".$_GET['token']."'");
```

```
    // Envoi du fichier
}
```

Pour envoyer le fichier au lecteur, nous allons utiliser deux fonctions. Header nous permettra de lui indiquer ce qu'on lui envoie par le biais du header HTTP *Content-type* (fichier audio mp3, vidéo avi, etc.), et readfile d'envoyer le fichier. Le navigateur n'y verra ainsi que du feu.
La fonction *header* s'utilise comme ceci :

Code : PHP

```php
<?php
// Fichier audio MP3
header('Content-type: audio/mp3');

// Fichier vidéo MPEG
header('Content-type: video/mpeg');
```

On a donc :

Code : PHP - get.php

```php
<?php
// Envoi du fichier
header('Content-type: audio/mp3');
readfile($fichier);
exit();
```

Comme dit plus haut, le lecteur que j'utilise n'est compatible qu'avec les fichiers audio mp3. Pour une compatibilité avec un autre lecteur supportant d'autres types de formats et de médias, vous pouvez faire quelque chose comme :

Code : PHP

```php
<?php
$video_extensions = array('avi', 'mpeg' /* ... */);
$audio_extensions = array('mp3', 'ogg'  /* ... */);

// S'il s'agit d'une vidéo
if(in_array($extension, $video_extensions)) {
     header('Content-type: video/'.$extension);
}
// Sinon, s'il s'agit d'un fichier audio
elseif(in_array($extension, $audio_extensions)) {
     header('Content-type: audio/'.$extension);
}
readfile($fichier);
```

Dans le cas où le token n'est pas valide, nous pourrions ne rien faire, ou tout simplement arrêter le script, mais j'ai mieux. Je vous propose de renvoyer un petit son d'*erreur* qui aura voulu accéder au fichier par l'URL.

Code : PHP

```php
<?php
header('Content-type: audio/mp3');
readfile('media/fail.mp3');
exit();
```

Vous pouvez télécharger le fichier *fail.mp3* ici : http://failsound.com/fail.mp3.

Code complet du script :

Secret (cliquez pour afficher)

Code : PHP - get.php

```php
<?php
if(!isset($_GET['token'])) {
 exit();
}
$token = $_GET['token'];

// Connexion à la base de données
$pdo = new PDO('mysql:dbname=tests;host=localhost', 'christophe',
'thepasswordisalie');

// Préparation de la requête
$q = $pdo->prepare('SELECT fichier FROM media_tokens WHERE
token=:token');
$q->bindValue(':token', $_GET['token']);

// Exécution !
$q->execute();

// Si le token correspond à un enregistré en base de données
if($q->rowCount() == 1) {
 // On récupère les résultats de la requête
 $resultat = $q->fetch();
 $fichier = 'media/'.$resultat['fichier']; // Dans le cas où vos
fichiers multimédia se trouvent dans media/
 $extension = end(explode('.', $fichier));

 // Si le fichier existe bien et qu'il est lisible, on peut le
renvoyer au navigateur
 if(file_exists($fichier) && is_readable($fichier)) {
  // On peut maintenant effacer le token qui est en base de
données
  $pdo->exec("DELETE FROM media_tokens WHERE
token='".$_GET['token']."'");

  // Ce header permet d'indiquer au navigateur quel à type de
fichier il doit associer celui qu'on lui renvoie
  header('Content-type: audio/'.$extension);
  readfile($fichier);
  exit();
 }
}
else {
 header('Content-type: audio/mp3');
 readfile('media/fail.mp3');
 exit();
}
```

Restriction d'accès

Il ne nous reste plus qu'une dernière chose à faire. Elle est très importante, car le système que l'on vient de coder ne serait d'aucune utilité sans elle.
Nous allons interdire l'accès au répertoire contenant les fichiers multimédias à tout le monde, sauf au serveur-lui même ; comme cela, seul le script pourra y accéder.

Code : Apache - media/.htaccess

```
deny from all
allow from 127.0.0.1
```

Notez que si vous faites des tests en local, vous aurez tout de même accès au répertoire **media/** car l'adresse avec laquelle votre ordinateur envoie les requêtes est 127.0.0.1 (logique, puisqu'en quelque sorte, il s'envoie des requêtes à lui-même).
Voilà, c'est terminé. J'espère que cette méthode vous sera utile.
En cas de problème ou si vous avez une question, n'hésitez pas à poster un commentaire, je les lis tous !

@+