

Créer une barre de progression grâce à PHP 5.4

La version de PHP 5.4 vient avec son lot de nouveautés et propose désormais de suivre la progression d'un fichier en cours de téléchargement grâce aux nouvelles directives et variables de session. Dans cet article, nous allons donc apprendre à exploiter au mieux ces nouvelles fonctionnalités. Nous allons également construire une micro-application Web pour le téléchargement et l'affichage d'images.

Cet article explique :

- Les nouvelles directives de session ainsi que leurs utilités.
- Créer un formulaire de téléchargement d'images en HTML 5 ainsi que son traitement côté serveur.
- Suivre la progression d'un transfert de fichier en temps réel grâce aux requêtes AJAX.

Ce qu'il faut savoir :

- Connaissance basique de PHP, HTML ainsi que JavaScript.
- Avoir une machine avec PHP 5.4 installé.
- Un navigateur Web supportant HTML 5 (toutes les versions récentes le supporte).

Avant la version de PHP 5.4, il était difficile de traquer la progression d'un fichier en cours de téléchargement.

En effet, il fallait obligatoirement utiliser Flash ou un snippet Perl ou installer l'extension PECL « uploadprogress » ou encore utiliser PHP APC.

Dans cet article, nous allons donc apprendre à connaître la progression d'un transfert en cours de téléchargement à l'aide des nouvelles variables de session sans faire appel à des logiciels tiers.

Puis nous allons créer une barre de progression à l'aide du framework JavaScript « jQuery » pour récupérer le pourcentage de progression en AJAX.

Les nouvelles directives qu'offre PHP 5.4 sont présentées dans le Tableau 1

Les directives `session.upload_progress.freq` et `session.upload_progress.min_freq` contrôlent la fréquence de mise à jour des informations de progression de téléchargement.

Notez qu'avec une configuration raisonnable de ces deux options, la surcoût en terme de charge est quasi

nul.

Vous pouvez modifier ces directives dans le fichier de configuration "php.ini".

N'oubliez pas de redémarrer votre serveur pour que ces données soient prises en compte.

En général, vous ne devriez pas modifier la directive `session.upload_progress.prefix`.

La directive `session.upload_progress.name` sert de nom pour un champ de formulaire caché contenant comme valeur, le nom de la session afin de ne pas obtenir deux mêmes noms de session.

Voici un exemple :

```
<input type="hidden" name="<?php echo ini_get('session.
upload_progress.name');?>"
value="NON_DE_VOTRE_SESSION" />
```

et la récupération de la session contenant les informations sur la progression du fichier se fait de cette manière :

Tableau 1. Nouvelles directives de session permettant de traquer la progression d'un fichier

Nom de la directive	Description	Valeur par défaut
<code>session.upload_progress.enabled</code>	Permet de traquer la progression d'un transfert dans la variable super-global <code>\$_SESSION</code>	On
<code>session.upload_progress.cleanup</code>	Nettoie les informations de progression dès que toutes les données POST ont été lues (quand un téléchargement est terminé).	On
<code>session.upload_progress.prefix</code>	Un préfixe utilisé pour la clé de la progression du téléchargement dans la variable super-global <code>\$_SESSION</code>	<code>upload_progress_</code>
<code>session.upload_progress.name</code>	Le nom d'index (enchaîné avec le préfixe) dans la variable super-global <code>\$_SESSION</code>	<code>PHP_SESSION_UPLOAD_PROGRESS</code>
<code>session.upload_progress.freq</code>	La fréquence de la progression du téléchargement qui doit être mis à jour.	1%
<code>session.upload_progress.min_freq</code>	Le délai minimum entre les mises à jour, en secondes.	1

```
$sSessionKey = ini_get('session.upload_progress.prefix')
    . $_POST[ini_get('session.upload_progress.name')];
var_dump($_SESSION[$sSessionKey]);
```

Comme vous l'avez compris, PHP utilise les sessions pour stocker la progression du fichier en cours de transfert.

Dans le Listing 1, vous avez un exemple à ce que devrait ressembler les données stockées en session.

Il est également possible d'annuler le téléchargement d'un fichier courant en définissant la clé de session `$_SESSION[$sSessionKey]['cancel_upload']` à la valeur `TRUE`.

Attention, lors du téléchargement de plusieurs fichiers dans la même requête, cette action n'annulera que le fichier qui est actuellement en cours de téléchargement, ainsi que ceux en attente de téléchargement, mais n'annulera pas les téléchargements terminés avec succès.

Lorsqu'un téléchargement est annulé en utilisant

Listing 1. Données de progression stockées en session

```
<?php
$_SESSION["upload_progress_NON_DE_VOTRE_SESSION"] = array(
    "start_time" => 1234567890, // L'heure de la requête
    "content_length" => 57343257, // Longueur du contenu POST
    "bytes_processed" => 453489, // Quantité d'octets reçus et traités
    "done" => false, // TRUE lorsque le gestionnaire POST a terminé avec succès, sinon FALSE
    "files" => array(
        0 => array(
            "field_name" => "file1", // Nom du champ <input />
            // Les 3 éléments suivants sont équivalents à ceux dans $_FILES
            "name" => "image.png",
            "tmp_name" => "/tmp/phpHYOAod",
            "error" => 0,
            "done" => true, // TRUE lorsque le gestionnaire POST a terminé de gérer ce fichier
            "start_time" => 1234567890, // L'heure de début de requête
            "bytes_processed" => 57343250, // Quantité d'octets reçus et traités pour ce fichier
        ),
    ),
);
```

Listing 2. Fichier `index.php` qui contient le formulaire

```

<?php
/**
 * Check the version of PHP
 */
if (version_compare(PHP_VERSION(), '5.4.0', '<'))
    exit('ERROR: Your PHP version is ' . PHP_VERSION() . ' but this script requires PHP 5.4.0 or higher.');
```

```

/**
 * Check if "session.upload_progress" is enabled
 */
if (!intval(ini_get('session.upload_progress.enabled')))
    exit('session.upload_progress.enabled is not enabled, please activate it in your PHP config file to use
    this script.');
```

```

require_once 'Upload.class.php';
?>
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8" />
    <title>PHP 5.4 Session Upload Progress Bar Demo</title>
    <meta name="description" content="PHP 5.4 Session Upload Progress Bar" />
    <meta name="keywords" content="PHP, session, upload, progress bar" />
    <meta name="author" content="Pierre-Henry Soria" />
    <link rel="stylesheet" href="/static/css/common.css" />
  </head>

  <div id="container">

    <header>
      <h1>Example for Progress Bar with PHP 5.4 and jQuery</h1>
    </header>

    <!-- Debug Mod --> <!-- <form action="upload.php?show_transfer=on" method="post" id="upload_form"
      enctype="multipart/form-data" target="result_frame"> -->
    <form action="upload.php" method="post" id="upload_form" enctype="multipart/form-data" target="result_
      frame">
      <fieldset>
        <legend>Upload Images</legend>
        <input type="hidden" name="<?php echo ini_get('session.upload_progress.name');?>" value="<?php
          Upload::UPLOAD_PROGRESS_PREFIX ?>" />
        <label for="file">Images: <input type="file" name="files[]" id="file" multiple="multiple"
          accept="image/*" required="required" />
        <small><em>You can select multiple files at once by clicking multiple files while holding down the
          "CTRL" key.</em></small></label>
        <button type="submit" id="upload">Upload!</button>
        <button type="reset" id="cancel">Cancel Upload</button>

        <!-- Progress bar here -->
        <div id="upload_progress" class="hidden center progress">
          <div class="bar"></div>
        </div>

      </fieldset>
    </form>

    <iframe id="result_frame" name="result_frame" src="about:blank"></iframe>

    <footer>
      <p>By <strong><a href="http://ph-7.github.com">PH7</a></strong> &copy; 2012.</p>
    </footer>

```

```

</div>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
<script src="./static/js/ProgressBar.class.js"></script>
<script>
$( '#upload' ).click(function() {
    (new UploadBar).upload();
});
$( '#cancel' ).click(function() {
    (new UploadBar).cancel();
});
</script>
</html>

```

cette méthode, la clé `error` du tableau `$_FILES` sera définie à `UPLOAD_ERR_EXTENSION`.

Utiliser AJAX pour une progression en temps réel

Les informations sur la progression du fichier que PHP stocke dans une session n'est pas particulièrement utile pour la requête de téléchargement en tant que tel, mais pendant le téléchargement.

Dans cette partie, nous allons donc envoyer des requêtes AJAX avec la librairie jQuery pour afficher en temps réel la progression du transfert des fichiers vers le serveurs.

Nous allons commencer par la page d'index qui contiendra le formulaire de téléchargement.

Ce formulaire contient un champ de type « file » ainsi que l'attribut `multiple` (disponible depuis HTML5) qui permet de sélectionner plusieurs fichiers en même temps.

Il nous faut aussi ajouter des crochets au non du champ afin que PHP sache que les différents fichiers sont stockés dans un tableau.

La méthode PHP `addFile()` de la classe `Upload` qui récupèrera les fichiers sera également adapté à l'aide d'une boucle `for`. Voir Listing 4.

Nous ajoutons aussi l'attribut `accept` au champ de type « file » qui permet de forcer au client le type de fichier à télécharger (chez nous, le type sera `image/*` afin d'accepter tous les types d'images). Mais gardez à l'esprit que cet attribut ne remplace en aucun cas la vérification côté serveur, car il est susceptible d'être modifié/supprimé par le client.

Pour la barre de progression, nous lui ajoutons la visibilité `hidden` pour qu'elle ne soit pas visible par défaut. Par la suite, nous allons la rendre visible pendant le téléchargement d'un fichier à l'aide de JavaScript.

Enfin, notez que les fichiers JavaScript sont placés en bas de la page afin d'éviter de ralentir le chargement de la page.

Et maintenant, voici le fichier «upload.php» qui servira à récupérer les informations sur

le fichier uploadé.

Nous ajoutons une variable `$_GET` pour avoir le choix entre l'envoi l'affichage du fichier uploadé ou l'affichage sur du contenu des variables super-global `$_SESSION` et `$_FILES`.

Pour le faire, il suffit d'ajouter ce paramètre dans le formulaire comme ceci :

```

<form action="upload.php?show_transfer=on"
method="post" id="upload_form"
enctype="multipart/form-data" target="result_frame">

```

Maintenant nous allons créer une classe PHP qui permettra de gérer l'envoi des fichiers vers le serveur, calculer le pourcentage du fichier en cours de transfert et l'affichage du contenu des deux variables `$_SESSION` et `$_FILES` afin de mieux comprendre le fonctionnement.

Nous allons maintenant créer une petite classe JavaScript qui va permettre de récupérer les informations en AJAX.

Nous créons quatre méthodes.

La première, `upload()` va soumettre le formulaire et appellera la méthode `progress()` toutes les secondes et demies pour récupérer le pourcentage de la progression du fichier.

La seconde, `cancel()` va permettre de stopper le chargement du fichier en cours de téléchargement. Nous utilisons aussi la méthode `fadeOut()` de jQuery afin de faire disparaître la barre de progression.

La troisième, `progress()` qui est appelée par la méthode `upload()` récupère donc le pourcentage du transfert en AJAX et affiche la barre de progression à l'aide de la méthode `show()` de jQuery. Cette méthode est appelée toutes les demies secondes.

Enfin, la dernière méthode `resetProgressBar()` est utilisée par `upload()` et `cancel()` pour réinitialiser la barre de progression à 0%

Notez aussi la présence de la variable `ome` faisant appel à la variable `this` (l'objet de la classe) est nécessaire pour résoudre les problèmes de résolution de portée de cette variable dans les fonctions de rappels jQuery.

Listing 3. *upload.php*

```
<?php
require_once 'Upload.class.php';

/**
 * If the request GET['show_transfer'] is "on", it shows $_FILES and $_SESSION
 * using the var_dump PHP function.
 * Otherwise it sends the image to the server and displays with information about the image.
 */
$sMethod = (!empty($_GET['show_transfer']) && $_GET['show_transfer'] == 'on') ? 'show' : 'addFile';
echo (new Upload)->$sMethod();
```

Listing 4. *Class Upload.class.php*

```
<?php
class Upload
{

    const UPLOAD_PROGRESS_PREFIX = 'progress_bar';

    private $_sMsg, $_sUploadDir, $_sProgressKey;

    // The short array syntax (only for PHP 5.4.0 and higher)
    private $_aErrFile = [
        UPLOAD_ERR_OK          => 'There is no error, the file uploaded with success.',
        UPLOAD_ERR_INI_SIZE    => 'The uploaded file exceeds the upload_max_filesize directive in php.ini.',
        UPLOAD_ERR_FORM_SIZE   => 'The uploaded file exceeds the MAX_FILE_SIZE directive that was specified in
            the HTML form.',
        UPLOAD_ERR_PARTIAL     => 'The uploaded file was only partially uploaded.',
        UPLOAD_ERR_NO_FILE     => 'No file was uploaded.',
        UPLOAD_ERR_NO_TMP_DIR  => 'Missing a temporary folder.',
        UPLOAD_ERR_CANT_WRITE  => 'Failed to write file to disk.',
        UPLOAD_ERR_EXTENSION  => 'A PHP extension stopped the file upload.'
    ];

    public function __construct()
    {
        // Session initialization
        if ('' === session_id()) session_start();

        $this->_sUploadDir = './uploads/';
        $this->_sProgressKey = strtolower(ini_get('session.upload_progress.prefix') . static::UPLOAD_PROGRESS_
            PREFIX);

        /**
         // You can also retrieve the session this way.
         $this->_sProgressKey = strtolower(ini_get('session.upload_progress.prefix') . $_POST[ini_get('session.
            upload_progress.name')]);
         */
    }

    /**
     * @return integer Percentage increase.
     */
    public function progress()
    {
        if(!empty($_SESSION[$this->_sProgressKey]))
        {
            $aData = $_SESSION[$this->_sProgressKey];
            $iProcessed = $aData['bytes_processed'];
```

```

        $iLength = $aData['content_length'];
        $iProgress = ceil(100*$iProcessed / $iLength);
    }
    else
    {
        $iProgress = 100;
    }

    return $iProgress;
}

/**
 * @return object this
 */
public function addFile()
{
    if(!empty($_FILES))
    {
        $this->_sMsg = '';

        foreach($_FILES as $sKey => $aFiles)
        {
            for($i = 0, $iNumFiles = count($aFiles['tmp_name']); $i < $iNumFiles; $i++)
            {
                /**
                 * Assign values of array in simple variables.
                 */
                $iErrCode = $aFiles['error'][$i];
                $iSize = $aFiles['size'][$i];
                $sFileName = $aFiles['name'][$i];
                $sTmpFile = $aFiles['tmp_name'][$i];
                $sFileDest = $this->_sUploadDir . $sFileName;
                $sTypeFile = $aFiles['type'][$i];

                /**
                 * Check files.
                 */
                $bIsImgExt = (strtolower(substr(strrchr($sFileName, '.'), 1))); // Get the file extension
                if(($bIsImgExt == 'jpeg' || $bIsImgExt == 'jpg' || $bIsImgExt == 'png' || $bIsImgExt ==
                'gif') && (strstr($sTypeFile, '/') === 'image'))
                {
                    if($iErrCode == UPLOAD_ERR_OK)
                    {
                        move_uploaded_file($sTmpFile, $sFileDest);
                        $this->_sMsg .= '<p style="color:green; font-weight:bold; text-align:center">Successful "' . $sFileName . '" file upload!</p>';
                        $this->_sMsg .= '<p style="text-align:center">Image type: ' . str_replace('image/',
                        '', $sTypeFile) . '<br />';
                        $this->_sMsg .= 'Size: ' . round($iSize / 1024) . ' KB<br />';
                        $this->_sMsg .= '<a href="' . $sFileDest . '" title="Click here to see the original
                        file" target="_blank"></a></p>';
                    }
                    else
                    {
                        $this->_sMsg .= '<p style="color:red; font-weight:bold; text-align:center">Error
                        while downloading the file "' . $sFileName . '"<br />';
                        $this->_sMsg .= 'Error code: ' . $iErrCode . '<br />';
                        $this->_sMsg .= 'Error message: ' . $this->_aErrFile[$iErrCode] . '</p>';
                    }
                }
            }
        }
    }
    else
    {

```

```

        $this->_sMsg .= '<p style="color:red; font-weight:bold; text-align:center">File type
        incompatible. Please save the image in .jpg, .jpeg, .png or .gif</p>';
    }
}
}
else
{
    $this->_sMsg = '<p style="color:red; font-weight:bold; text-align:center">You must select at least
    one file before submitting the form.</p>';
}
return $this;
}
/**
 * @return object this
 */
public function show()
{
    ob_start();
    echo '<p><strong>$_FILES Result:</strong></p><pre>';
    var_dump($_FILES);
    echo '</pre>';
    echo '<p><strong>$_SESSION Result:</strong></p><pre>';
    var_dump($_SESSION);
    echo '</pre>';
    $this->_sMsg = ob_get_clean();

    return $this;
}
/**
 * Cancel the file download.
 *
 * @return object this
 */
public function cancel()
{
    if (!empty($_SESSION[$this->_sProgressKey]))
        $_SESSION[$this->_sProgressKey]['cancel_upload'] = true;

    return $this;
}
/**
 * Get the JSON informational message.
 *
 * @param integer $iStatus, 1 = success, 0 = error
 * @param string $sTxt
 * @return string JSON Format.
 */
public static function jsonMsg($iStatus, $sTxt)
{
    return '{"status":' . $iStatus . ', "txt":"' . $sTxt . '"}';
}
/**
 * Get the informational message.
 *
 * @return string
 */
public function __toString()
{
    return $this->_sMsg;
}
}

```

Listing 5. Classe `UploadBar.class.js` pour récupérer les informations sur la progression en AJAX

```

function UploadBar()
{
    var oMe = this; // Self Object

    this.upload = function()
    {
        this.resetProgressBar();

        // Submit the form
        $('#upload_form').submit();

        setTimeout(function() {oMe.progress()}, 1500);
    };

    this.cancel = function()
    {
        $.post('download_progress.ajax.php', {param : 'cancel'}, function(oData)
        {
            console.log('Upload cancelled');
        });

        this.resetProgressBar();
        $('#upload_progress').delay(2000).fadeOut();
    };

    this.progress = function()
    {
        $.post('download_progress.ajax.php', {param : 'progress'}, function(iPercentage)
        {
            var sPercentage = iPercentage + '%';

            $('#upload_progress').show();
            $('#upload_progress .bar').html(sPercentage);
            $('#upload_progress .bar').width(sPercentage);

            if(iPercentage < 100)
                setTimeout(function() {oMe.progress()}, 500);
        });
    };

    this.resetProgressBar = function()
    {
        $('#upload_progress .bar').html('0%');
        $('#upload_progress .bar').width('0%');
    };
}

```


Pour terminer, voici le fichier PHP qui est utilisé pas la classe `UploadBar.class.js` pour récupérer les informations de la classe `Upload.class.php`.

Listing 6. Fichier `download_progress.ajax.php`

```
<?php
require_once 'Upload.class.php';
$$Parameter = (!empty($_POST['param'])) ? strip_
    tags($_POST['param']) : '';
$oUpload = new Upload;
switch($$Parameter)
{
    case 'progress':
        echo $oUpload->progress();
        break;
    case 'cancel':
        $oUpload->cancel();
        break;
}
```

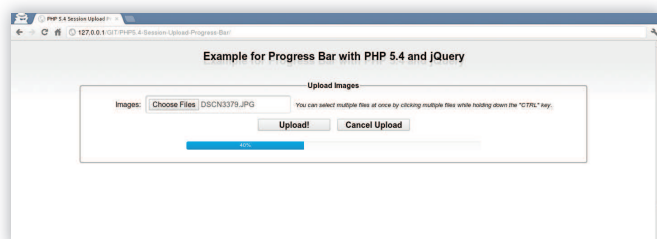


Figure 1. Fichier en cours de téléchargement...

Conclusion

Dans cet article, vous avez vu les nouvelles directives de session ainsi que les variables de session générées par PHP qui permettent de traquer la progression des fichiers envoyés vers le serveur.

Vous avez vu leurs utilités et la manière de vous en servir.

Enfin vous avez appris à afficher ces informations en temps réel dans une barre de progression en utilisant AJAX avec la librairie jQuery.

Sur Internet

- <http://github.com/pH-7/PHP5.4-Session-Upload-Progress-Bar> – La totalité du code source abordée dans cet article,
- <http://github.com/pH-7/PHP5.4-Session-Upload-Progress-Bar/blob/master/static/css/common.css> – Le fichier CSS du projet qui n'est pas évoqué dans cet article,
- <http://php.net/manual/fr/session.upload-progress.php> – La documentation officielle à propos de la progression d'un téléchargement en session,
- <http://php.net/manual/fr/session.configuration.php> – La liste des différentes directives de session,
- <http://jquery.com> – Le site officiel de la librairie jQuery. Vous y trouverez également une documentation très riche et en français.

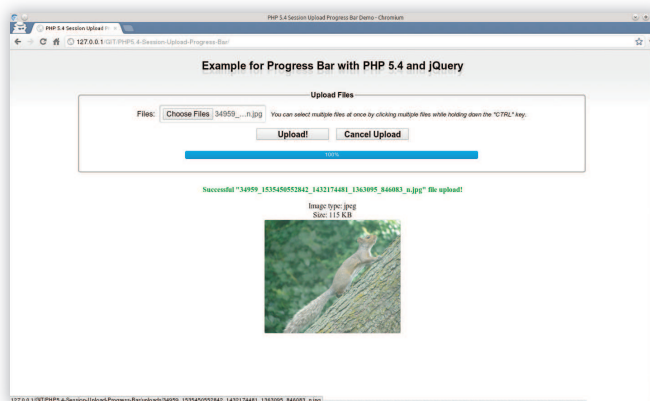


Figure 2. Barre de progression générée à l'aide de PHP 5.4 et de jQuery

PIERRE-HENRY SORIA

Pierre-Henry Soria travaille dans le développement Web depuis plusieurs années. Il se spécialise sur l'accessibilité, le référencement ainsi que la sécurité et les techniques possibles pour éviter les intrusions et les failles de sécurité. Il travaille également sur les parseurs et analyseurs lexicaux et syntaxiques en langages de bas niveaux. Il peut être contacté par email à : phenrysoria@gmail.com

Dépannage

Si la barre de progression ne fonctionne pas (exemple : la variable `$_SESSION` est vide), voici quelques vérifications que vous pouvez essayer de faire avant de demander de l'aide sur un forum PHP d'entraide.

1. La directive `session.upload_progress.enabled` est bien activée.
2. Le fichier est assez grand pour au moins 5 secondes de téléchargement.
3. Le téléchargement de fichiers et la limite de taille de fichiers sont correctement configurés.
4. Les directives `session.upload_progress.freq` et `session.upload_progress.min_freq` sont adéquates pour les tailles de fichiers avec lesquels vous allez les uploader.
5. La directive `file_uploads` est bien activée.
6. Les directives `upload_max_filesize` et `post_max_size` sont réglées sur une taille suffisante pour le(s) fichier(s) à télécharger.
7. Le répertoire indiqué dans la directive `upload_tmp_dir` est bien en permission « écriture ».
8. Vérifiez également les messages d'erreur dans le tableau `$_FILES` pour obtenir des informations supplémentaires.