

Blog Dreaminvasion - développement et administration réseaux

PHP : Classe d'envoi d'email

publié par admin le 2007-09-18 23:31:00



Je le sais il existe pas mal de classe en PHP permettant l'envoi d'email. Mais en cherchant je ne trouvais pas toujours de que je voulais au niveau des fonctionnalités des classes fournis.

Aussi j'ai décidé de développer une classe permettant aussi bien l'envoi d'email text / HTML, que l'envoi d'e-mailing.

Cette classe est compatible PHP4 et PHP5 (je ne l'ai pas totalement développé pour PHP5 car certain de mes projets sont encore sur des hébergeurs n'ayant que PHP4 :().

Les fonctionnalités de la classe sont :

- Envoi d'email au format TEXT.
- Envoi d'email au format TEXT /HTML.
- Envoi d'email au format HTML en utilisant un Template.
- Envoi d'email prioritaire.
- Configuration d'un email de réponse différente de l'email de l'émetteur.

Téléchargement de la classe :

[Classe PHP d'envoi email version 1.0 \(Archive ZIP, 2 Ko\)](#)

Utilisation de la classe pour l'envoi d'un email au format TEXT/PLAIN :

Voici l'utilisation classique de cette classe.

```
<?php

//Inclusion de la classe
Include('?email.class.php?');

// Initialisation de la classe
$email = new Email();

/* Obligatoire */
// Votre nom ou celle de votre site web (Nom qui apparaîtra dans la messagerie de votre destinataire)
$email->SetNomSite("Mon site web");
// Email utilisé pour l'envoi.
$email->SetEmetteur("emetteur@domain.tld");
// Email à utiliser pour la réponse de votre correspondant.
```

```

$email->SetReturnPath("mon.email.de.retour@domain.tld");

// Email du destinataire
$email->SetDestinataire("email@domain.tld");

// Sujet de votre email
$email->SetSujet("Sujet du mail");

// Message au format TEXT/plain
$email->SetMessageTexte("Message au format text/plain");

/* Options facultative */
// L'email est envoyé avec une Haute Priorité
$email->SetIsImportant(true); // Flag important

// Envoi de l'email
if( $email->EnvoiMail() )
{
    echo "Email Envoyé";
}
else {
    echo "Erreur lors de l'envoi de l'email";
}

?>

```

Utilisation de la classe pour l'envoi d'un email au format HTML :

L'envoi d'un email au format HTML ne change pas vraiment de l'envoi au format TEXT, il suffit d'utiliser la méthode **Email::SetMessageHTML()**.

```

<?php

//Inclusion de la classe
Include('?email.class.php?');

// Initialisation de la classe
$email = new Email();

/* Obligatoire */
// Votre nom ou celle de votre site web (Nom qui apparaîtra dans la messagerie de votre destinataire)
$email->SetNomSite("Mon site web");
// Email utilisé pour l'envoi.
$email->SetEmetteur("emetteur@domain.tld");
// Email à utiliser pour la réponse de votre correspondant.
$email->SetReturnPath("mon.email.de.retour@domain.tld");

// Email du destinataire

```

```

$email->SetDestinataire("email@domain.tld");

// Sujet de votre email
$email->SetSujet("Sujet du mail");

// Message au format TEXT/plain
$email->SetMessageTexte("Message au format text/plain");

// Message au format HTML
$email->SetMessageHTML("<h1>Message au format text/HTML</h1>");

/* Options facultative */
// L?email est envoyé avec une Haute Priorité
$email->SetIsImportant(true); // Flag important

// Envoi de l'email
if( $email->EnvoiMail() )
{
    echo "Email Envoyé";
}
else {
    echo "Erreur lors de l'envoi de l'email";
}

?>

```

Utilisation de la classe pour l'envoi d'un email au format HTML en utilisant un TEMPLATE HTML :

Cette fonctionnalité va vous permettre d'envoyer des emails HTML en utilisant un Template (par exemple le template de votre site), ce qui est très appréciable pour des e-mailing.

Création du template HTML

Créer le template que vous allez utiliser pour l'envoi de l'email.

Deux variables sont a votre disposition :

- %url% : sera remplacé par l'url de votre choix.
- %texte% : sera remplacé par le texte HTML de votre email.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<style type="text/css">
    body {background : #FFFFFF;}
    #container {border : 1px solid #000000;}
</style>
</head>

```

```

<body>
  <div id="container">
    <div id="head">
      <img src%url%/logo-site.gif" />
    </div>

    <div id="content">
      %texte%
    </div>
  </div>
</body>
</html>

```

Utilisation de la classe

Enfin l'utilisation de la classe en utilisant désormais les méthodes **Email::SendMessageHTML()** et **Email::TemplateHTML()**.

```

<?php

//Inclusion de la classe
Include('email.class.php') ;

// Initialisation de la classe
$email = new Email();

/* Obligatoire */
// Votre nom ou celle de votre site web (Nom qui apparaîtra dans la messagerie de votre destinataire)
$email->SetNomSite("Mon site web");
// Email utilisé pour l'envoi.
$email->SetEmetteur("emetteur@domain.tld");
// Email à utiliser pour la réponse de votre correspondant.
$email->SetReturnPath("mon.email.de.retour@domain.tld");

// Email du destinataire
$email->SetDestinataire("email@domain.tld");

// Sujet de votre email
$email->SetSujet("Sujet du mail");

// Message au format TEXT/plain
$email->SetMessageTexte("Message au format text/plain");

// Message au format HTML
$email->SendMessageHTML("<h1>Message au format text/HTML</h1>");

// Choix du template HTML
// $urlDeVotreSite : remplacera %url% par le contenu de cette variable
// $pathToTemplate : chemin vers le template créé précédemment
// La variable %texte% dans votre template sera remplacé par le contenu de la attribut :

```

```

$messageTexte
$email->TemplateHTML($urlDeVotreSite, $pathToTemplate);

/* Options facultative */
// L'email est envoyé avec une Haute Priorité
$email->SetIsImportant(true); // Flag important

/* Envoi de l'email */
if( $email->EnvoiMail() )
{
    echo "Email Envoyé";
}
else {
    echo "Erreur lors de l'envoi de l'email";
}

?>

```

Contenu de la classe Email:

```

<?php
/*
 * -----
 * Author   : Gérald LONLAS
 * Email    : webmaster@dreaminvasion.com
 * Website  : http://www.dreaminvasion.com
 * -----
 */

class Email{

    /* Attributs
    *****/

    var $mime_boundary;
    var $entete;
    var $messageMail;

    var $isImportant = false;
    var $nomSite;

    var $emetteur;
    var $returnPath;
    var $destinataire;

    var $sujet;
    var $messageTexte;

```

```

var $messageHTML;

/* Methodes
*****/

// Entete du mail
function Entete()
{
    $this->entete = "From: ".$this->nomSite."<".$this->returnPath.">\n";
    $this->entete .= "Reply-To: ".$this->nomSite."<".$this->emetteur.">\n";
    $this->entete .= "MIME-Version: 1.0\n";
    $this->entete .= 'Content-Type: multipart/alternative; boundary=$this-
>mime_boundary."'."'\n";
}

// Entete pour les mails important
function Important()
{
    $this->entete .= "X-Priority: 1\n";
    $this->entete .= "Importance: High\n";
    $this->entete .= "X-MSMail-Priority: High\n";
}

function Message()
{
    // Creation du message
    $this->messageMail = "--".$this->mime_boundary."\n";
    $this->messageMail .= "Content-Type: text/plain; charset=iso-8859-16\n";
    $this->messageMail .= "Content-Transfer-Encoding: 8bit\n\n";
    $this->messageMail .= $this->messageTexte."\n\n";

    // Creation du message HTML
    if($this->messageHTML != "")
    {
        $this->messageMail .= "--".$this->mime_boundary."\n";
        $this->messageMail .= "Content-Type: text/html; charset=iso-8859-16\n";
        $this->messageMail .= "Content-Transfer-Encoding: 8bit\n\n";

        $this->messageMail .= $this->messageHTML."\n";
    }

    $this->messageMail .= "--".$this->mime_boundary."--\n\n";
}

// Creation d'un template au format HTML
function TemplateHTML($url, $pathTemplate)
{
    // Lis le fichier template, change les variables persos (%texte%, %url%), et l'affecte une
    variable PHP
    $fic = @fopen($pathTemplate, 'r');

```

```

$template = @fread($fic, filesize($pathTemplate));
@fclose($fic);

$template = str_replace("%texte%", $this->messageHTML, $template);
$template = str_replace("%url%", $url, $template);

$this->messageHTML = $template;

return $template;
}

// Envoi de l'email
function EnvoiMail()
{
    $this->mime_boundary = "----PHPSenderMail----".md5(time());

    // Creation de l'entete
    $this->Entete();

    // Ajout de l'entete important
    if($this->isImportant)
    {
        $this->Important();
    }

    // Ajout du message
    $this->Message();

    return @mail($this->destinataire, $this-> sujet, $this->messageMail, $this->entete);
}

/* Getter & Setter
*****

/**
 * Getter $isImportant
 *
 **/
function GetIsImportant(){
    return $this->isImportant;
}

/**
 * Setter $
 *
 **/
function SetIsImportant($var){
    $this->isImportant = $var;
}

/**

```

```

* Getter $nomSite
*
**/
function GetNomSite(){
    return $this->nomSite;
}

/**
* Setter $nomSite
*
**/
function SetNomSite($var){
    $this->nomSite = $var;
}

/**
* Getter $emetteur
*
**/
function GetEmetteur(){
    return $this->emetteur;
}

/**
* Setter $emetteur
*
**/
function SetEmetteur($var){
    $this->emetteur = $var;
}

/**
* Getter $returnPath
*
**/
function GetReturnPath(){
    return $this->returnPath;
}

/**
* Setter $returnPath
*
**/
function SetReturnPath($var){
    $this->returnPath = $var;
}

/**
* Getter $destinataire
*
**/

```

```

function GetDestinataire(){
    return $this->destinataire;
}

/**
 * Setter $destinataire
 *
 **/
function SetDestinataire($var){
    $this->destinataire = $var;
}

/**
 * Getter $sujet
 *
 **/
function GetSujet(){
    return $this->sujet;
}

/**
 * Setter $sujet
 *
 **/
function SetSujet($var){
    $this->sujet = $var;
}

/**
 * Getter $messageTexte
 *
 **/
function GetMessageTexte(){
    return $this->messageTexte;
}

/**
 * Setter $messageTexte
 *
 **/
function SetMessageTexte($var){
    $this->messageTexte = $var;
}

/**
 * Getter $messageHTML
 *
 **/
function GetMessageHTML(){
    return $this->messageHTML;
}

```

```
/**
 * Setter $messageHTML
 *
 **/
function SetMessageHTML($var){
    $this->messageHTML = $var;
}
}
?>
```

Class ZipExtract, décompresser (extraire) un fichier zip avec PHP

publié par admin le 2007-07-27 09:00:00



ZipExtract est une classe en PHP que je viens de faire pour décompresser des archives au format Zip.

Ayant eu du mal à trouver ce que je voulais en matière de décompression d'archive zip en PHP (classe trop lourde, ensemble de fonctions à bidouiller), j'ai décidé de faire ma propre classe.

Fonctionnalités de la classe ZipExtract.lib.php

Pour l'instant les fonctionnalités proposées par la classe sont simple et permettent de :

- Vérifier que l'archive zip n'est pas corrompue.
- Décompresser de l'archive.
- Choisir le chemin de destination.
- Supprimer l'archive.
- Liste les fichiers extraits.

Prérequis

Pour utiliser cette classe vous devez avoir au moins les versions de PHP suivantes : PHP 4 >= 4.2.0, PHP 5 >= 5.2.0.

[Pour plus d'informations, le manuel php.net sur ZIP](#)

Téléchargement

- [zipextract.lib.zip](#) (Format Zip : 2 Ko) : [ZipExtract.lib.php version 1.0.01](#)
- [Documentation de la classe : version 1.0](#) (Format PDF 22 Ko)

Utilisation de la classe

```
<?php

// Inclusion de la class ZipExtract
require_once( 'zipextract.lib.php' );
```

```

$filename = 'install.zip'; // Sous unix / linux
$filename = 'c:\tmp\install.zip'; // Sous windows si votre archive se trouve dans le dossier
C:\tmp

// Création de l'objet ZipExtract
$zip = new ZipExtract();

if( $zip->OpenZipFile( $filename ) )
{
    // Parametrage de la classe
    $zip->SetOutputDestination('util');
    $zip->SetIsChangeNameFile(false);

    // Extraction des fichiers
    $zip->Extract();

    // Suppression du fichier ZIP
    if( $zip->DeleteZipFile() )
    {
        echo "<p>Fichier zip supprimé</p>";
    }
    else
    {
        echo "<p>Pb suppression fichier Zip</p>";
    }

    // Affiche les fichiers dans le zip
    echo "Fichiers extraits : <ul>";
    foreach ( $zip->ListFiles() as $nom_fichier )
    {
        echo "<li>".$nom_fichier."</li>";
    }
    echo "</ul>";
}

?>

```

N'hésitez pas à me laisser un commentaire ou à me contacter pour toutes vos questions, critiques, suggestions

Conseils de performance pour MySQL

publié par admin le 2007-07-13 21:00:00



MySQL est un des SGBD les plus utilisés. Nous le voyons souvent dans les applications Web. Rapide, fiable et flexible ce SGBD permet une grande souplesse pour nos applications.

Cependant **le coût de la performance est un point à prendre en compte lorsque nous développons** Vider la mémoire lorsque nos variables ne sont plus utiles, éviter des requêtes trop complexes, alléger les échanges entr e le

client et le serveur, sont des contraintes que nous connaissons tous.

Mais quand est-il de nos questions à propos de notre utilisation de MySQL ?

[Jay Pipes vient de publier quelques conseils sur les performances de MySQL](#). Il répond via un FAQ aux questions que nous nous posons sur le choix des types de données, du choix du moteur de base à utiliser et d'autres questions qui sûrement vous aideront dans l'optimisation des vos applications MySQL.

InnoDB est-il plus performant / meilleur que MyISAM ?

Le type Char est-il plus rapide que Varchar ?

Le changement d'une derivated table vers une view change t'elle les performances ?

[Common Questions and Answers from Performance Tuning Webinars](#)