PHP : Les mails

Envoyer un mail est chose courante dans une application web.

Notez bien que cela fonctionne dans les 2 sens :

- l'internaute prend contact avec le propriétaire d'un site à partir d'un formulaire de contact (ou d'inscription)
- le propriétaire d'un site envoie des mails aux internautes :
  - confirmation d'inscription
  - confirmation de commande d'un achat en ligne
  - newsletter
  - rappel d'identifiant, réinitialisation d'un mot de passe
  - pièce jointe : facture, documentation...

Un mail est structuré en 2 parties :

- Une partie entête (= *headers*) indiquant des données techniques. Comme pour une page HTML, cette partie n'est pas affichée au lecteur du mail mais reste visible dans le code source.
- Une partie corps (= *body*) qui comprend le message et représente donc la partie visible affichée par les logiciels de messagerie.

> **Important :** l'envoi d'un mail ne relève pas que du langage informatique utilisé. Plusieurs normes informatiques (RFC 2822, RFC 2047...), logiciels serveurs et protocoles de communication (SMTP, IMAP etc.) entrent en jeu, ainsi que la configuration des serveurs d'envoi et de réception, des hébergeurs ou encore des fournisseurs de messagerie (Gmail, Yahoo, Orange, Free...), des logiciels de lecture clients (Outlook, Thunderbird...) et filtres antispam. Autant de points qui rendent compliqué de s'assurer de la bonne réception d'un courriel par un destinataire.

Pré-requis : configuration d'un serveur mail

Par défaut, Wamp ne permet pas l'envoi de mail.

Il est nécessaire d'installer un émulateur pour remplacer les serveurs de mails (*sendmail*, *postfix*) que l'on trouve sur les hébergements Linux.

Installation d'un serveur mail

[Mailhog]() permet d'intercepter les mails envoyés dans un environnement local sous Windows, tel que Wamp.

Procédure d'installation :

- Téléchargez ici le fichier *MailHog_windows_amd64.exe* et déplacez-le dans *C:/wamp/mailhog/* (répertoire *mailhog* à créer).
- Renommez le fichier *mailhog.exe*.

Configuration de Wamp

- Ouvrez le fichier *php.ini* (fichier de configuration de PHP) : clic gauche sur l'icône Wamp > *PHP* > *php.ini*.
- Recherchez la section *[mail function]*

- Renseignez les valeurs suivantes :

```
SMTP = 127.0.0.1
smtp_port = 1025
sendmail_path = "C:/wamp/mailhog/mailhog.exe sendmail"
mail.log = "C:/wamp/logs/mails.log"
```

- Redémarrez Wamp.

> Pour activer une variable dans *php.ini*, il faut éventuellement la décommenter en supprimant le `;` en début de ligne.

Lancement de maillhog

- Exécutez **en tant qu'administrateur** le fichier *C:/wamp/mailhog/mailhog.exe*
- Ouvrez l'adresse http://localhost:8025 dans votre *navigateur*

> Attention : Mailhog n'est qu'un émulateur qui ne fait qu'intercepter et afficher les mails sur le SMTP local, il ne les envoient pas réellement. C'est un outil de mise au point des mails en phase de développement et non pas un serveur de mails complet comme *sendmail*.

la fonction `mail()`

Pour expédier un mail, PHP fournit la fonction `mail()` :

```
mail(destinataire, objet, message, entêtes, paramètres)
```

Cette fonction retourne un booléen : `TRUE` si exécutée correctement, `FALSE` si échec.

Les arguments de la fonction sont, dans l'ordre :

- Destinataire : l'adresse mail de la personne à laquelle on veut envoyer le mail
- Objet : le sujet du mail, tel qu'il apparaîtra dans les clients de messagerie
- Message : le contenu principal du mail, tel qu'il apparaîtra dans les clients de messagerie
- Entêtes (= headers) : des informations techniques facultatives.
- Paramètres : d'autres informations techniques facultatives.

Exemple :

```
<?php
mail("dave.loper@afpa.com", "Confirmation d'inscription", "Bienvenue sur JardiXou ! Ta peux y acheter des tomates cerises pour 1 apéro et une brouette pour les transporter. Sans elle tes
```

Pour éviter le spam, la plupart des hébergeurs ne permettent pas l'utilisation de cette fonction. Ils la désactivent ou en limitent l'usage (quota de nombre de mails sur une période donnée). Certains proposent une alternative : arguments limités, surcharge par d'autres fonctions... Il faut en tenir compte dans les critères de choix d'un hébergeur.

Les paramètres de la fonction `mail()`

Le(s) destinataire(s)

Le destinataire peut être écrit sous différentes formes, mais le formatage de cette chaîne doit être conforme à la norme RFC 2822.

Formes les plus courantes :

- `dave.loper@afpa.fr`
- `Dave Loper <dave.loper@afpa.fr>`

Il est possible d'indiquer plusieurs destinataires (attention, les filtres antispam n'aiment pas ça), dans ce cas les noms doivent être séparés par une virgule :

```
<?php
$destinataire = "Dave Loper <dave.loper@afpa.fr>, jessica.pikatchin@laposte.net, alain.terieur@gmail.com";
```

L'objet (sujet)

L'objet est le sujet du mail. Il doit lui aussi répondre à la norme RFC 2822 et respecter des bonnes pratiques de rédaction, sous peine d'être classé comme spam, par conséquent cela relève plus du marketing.

L'objet ne doit pas être trop long, sous peine d'être tronqué à la lecture.

Le corps

Le corps est la partie qui comprend le contenu du mail, c'est-à-dire la partie à afficher.

Celui-ci peut-être au format texte ou HTML; il est possible de proposer pour un même mail une configuration prenant en compte ces 2 alternatives.

Les entêtes

Les entêtes - **headers** - sont des informations techniques additionnelles, telles que :

- le format (texte ou HTML)
- l'encodage (UTF-8 en général)
- les destinataires en copie
- les pièces jointes

Ces informations sont facultatives mais dans la réalité certaines sont exigées par les services de messageries (Gmail, Outlook, Yahoo!...) sinon le mail est considéré comme spam.

Les entêtes ne sont pas visibles lorsque vous visualisez un mail : il faut afficher le code source du mail (dans les logicles de messagerie) pour les voir.

Les entêtes peuvent être indiquées :

- avec une concaténation de chaînes, ce qui nécessite d'ajouter des caractères de retour chariot (`\n`, `\r` ou `\n\r`) pour séparer chaque entête (cf. exemple 1)
- soit, depuis la version 7.2.0 de PHP, sous forme de tableau PHP (cf. exemple 2). **Cette deuxième forme est désormais recommandée**, pour des raisons de lisibilité et de sécurité.

> Avec la concaténation, une faille de sécurité majeure appelée injection d'entêtes était possible; celle-ci a été corrigée depuis les versions 5.4.42 et 5.5.27 de PHP.

Exemple 1 : déclaration d'entêtes par une chaine concaténée

> Cette syntaxe n'est plus recommandée depuis la version 7.2.0 de PHP.

```
$eheaders  = 'MIME-Version: 1.0' . "\r\n";
$eheaders .= 'Content-Type: text/html; charset=utf-8' . "\r\n";
$eheaders .= 'From: Dave Loper <dave.loper@afpa.fr>' . "\r\n";
$eheaders .= 'Reply-To: Service commercial <commercialsw@jardiXou.com>' . "\r\n";
$eheaders .= 'X-Mailer: PHP/' . phpversion();  "\r\n";
```

Exemple 2 : déclaration d'entêtes par un tableau PHP

> Cette syntaxe est recommandée depuis la version 7.2.0 de PHP.

```
$eheaders = array('MIME-Version' => '1.0',
                  'Content-Type' => 'text/html; charset=utf-8',
                  'From' => 'Dave Loper <dave.loper@afpa.fr>',
                  'Reply-To' => 'Service commercial <commercialsw@jardiXou.com>',
                  'X-Mailer' => 'PHP/' . phpversion()
                  );
```

Liste des entêtes courantes :

> Respecter la casse des noms d'entête.

Valeur	Description
BCC	*Blind Carbon Copy*, ou copie carbone cachée : adresses mail des personnes recevant une copie du message; ces adresses sont masquées par le destinataire. Attention, les logiciels antispam n'aiment pas.
CC	*Carbon Copy*, ou copie carbone : adresses mail des personnes recevant une copie du message; ces adresses sont visibles par le destinataire. Attention, les logiciels antispam n'aiment pas.
Content-Type	Type de contenu du mail, c'est-à-dire le format.
From	Expéditeur du mail.
MIME-Version	Version du type MIME, toujours la valeur *1.0*.
Reply-To	Adresse mail de réponse au mail. Si non indiquée, cette adresse sera celle de l'expéditeur spécifiée dans *From*.
X-Mailer	Indique le logiciel, service ou langage (par exemple la version de PHP) utilisé pour envoyer le mail.

Format texte et format HTML

Un mail peut être envoyé soit au format texte, soit au format HTML, ou les 2 en même temps.

Le format texte

Au format, texte (*plain text* en anglais), le mail sera affiché à la lecture en texte brut sans aucune mise en forme (couleurs, polices...), un peu comme dans le bloc-note donc.

Il s'agit du format par défaut, il est donc inutile de le préciser dans les entêtes, c'est donc que vous avez vu au paragraphe *la fonction `mail()`*.

Le format texte permet de s'assurer que le mail pourra être lu par tous les dispositifs d'affichage et tous les logiciels. Ceci est de moins en moins vrai car la plupart des clients de messagerie modernes sont capables d'interpréter le format HTML.

Exemple

```
$eheaders[] = "Content-Type: text/plain; charset=utf-8";
```

Le format HTML

Dans le format HTML, un mail est affiché à l'internaute comme une véritable page web avec une structure en HTML, une mise en forme via CSS, ajout de liens, d'images etc., on peut même y adjoindre des interactions Javascript.

Pour indiquer le format HTML, il faut ajouter les 2 entêtes suivantes :

```
$eheaders[] = "MIME-Version: 1.0";
$eheaders[] = "Content-Type: text/html; charset=utf-8";
```

Le message sera lui codé en HTML.

Exemple (avec Bootstrap)

```
$message = "<!DOCTYPE html>
<html lang='fr'>
<head>
<meta charset='utf-8'>
<title>Mon premier mail HTML</title>
<meta name='viewport' content='width=device-width, initial-scale=1, shrink-to-fit=no'>
<link rel='stylesheet' href='https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css' integrity='sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T' c
<style>
html
{
    font-size: 10PX;
}

body
{
    font-size: 1rem; /* si html fixé à 10PX, 1rem = 10px = taille par défaut de police de Firefox ou Chrome */
}
</style>
</head>
<body>
<div class='container'>
    <div class='row'>
        <div class='col-12'>
            <h1>Mon premier mail HTML</h1>
        </div>
    </div>
    <div class='row'>
        <div class='col-12'>
            Ouah c'est trop génial ! On peut même mettre une image.
        </div>
    </div>
    </div>
    <div class='row'>
        <div class='col-12'>
            <img src='jardixou_logo.jpg' title='Logo' alt='Logo' class='img-fluid'>
        </div>
    </div>
</div>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src='https://code.jquery.com/jquery-3.3.1.slim.min.js' integrity='sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo' crossorigin='anonymous'></script>
<script src='https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js' integrity='sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM' crossorigin='anonym
</body>
</html>";
```

Bien évidemment, on peut mettre dans le code de la page des variables PHP : données postées d'un formulaire (de contact par exemple) ou extraites d'une base de données.

Formats combinés

Il est recommandé en termes de bonnes pratiques de proposer un mail dans les 2 formats.

Dans ce cas, le code source du mail est alors scindé en 2 parties distinctes séparées par une "frontière" (*boundary* en anglais). On retrouve cette technique dans les mails avec pièce jointe.

Mail avec pièce jointe

+++ TODO [19/08/2019] +++

Librairies, frameworks, CMS et services d'emailing

Pour faciliter le codage des mails, des librairies externes telles que SwiftMailer (écosystème Symfony) ou encore PHPMailer](https://github.com/PHPMailer/PHPMailer) ont vu le jour.

Les frameworks et les CMS possèdent eux des outils natifs ou des plugins, par exemple la librairie Email dans CodeIgniter.

Enfin, il existe aussi des services d'envoi de mails en masse (campagne emailing, newsletters) en ligne tels que Mailchimp ou Sarbacane, en partie payants et qui assurent une délivrabilité optimale des campagnes d'emailing avec un suivi statistique (taux d'ouverture, de désinscription etc.).

Bonnes pratiques

L'envoi de mails doit implémenter certaines bonnes pratiques, sous peine de votre classement comme spammeur :

- balisage HTML correct
- définition de l'encodage (UTF-8)
- HTML responsive pour qu'il puisse être lisible aussi bien sur PC que sur smartphones ou tablettes
- Présence obligatoire d'un lien de désinscription de l'utilisateur
- Lien de redirection du mail vers une page web permettant l'affichage correct du mail (dans le cas où le client de messagerie ne le permet pas)

> Votre serveur (adresse IP) pourrait être blacklisté comme spammeur pour longtemps (il existe des bases de données sur lesquelles se fondent les logiciels antispam). Vous pouvez cependant vous retrouver avec un mail classé en spam suite à une mauvaise configuration du serveur ou un contenu mal interprété.

Ressources

- arobase.org