
TP : Envoi de Mail avec PHP

Introduction

Ce tutoriel explique comment envoyer des e-mails en utilisant PHP. Nous couvrirons deux parties distinctes :

1. **Envoi d'e-mail simple** avec PHP et sécurisation des entrées utilisateur.
 2. **Utilisation de PHPMailer** pour un envoi avancé avec **SMTP** et **pièces jointes**.
-

Première Partie : Envoi d'un e-mail simple en PHP

1. Configuration des fichiers nécessaires

Avant d'envoyer un e-mail en PHP, vous devez configurer certains fichiers pour permettre la communication avec un serveur SMTP.

Qu'est-ce que **php.ini** et pourquoi le configurer ?

Le fichier **php.ini** est le fichier de configuration principal de PHP. Il permet de définir les options du langage, notamment celles liées à l'envoi d'e-mails. Sans une bonne configuration, PHP ne pourra pas envoyer d'e-mails en local.

Modifier **php.ini**

- Ouvrez **XAMPP Control Panel** → **Config** sous **Apache** → Sélectionnez **PHP (php.ini)**.
- Recherchez **[mail function]** et assurez-vous que ces lignes sont configurées correctement :

```
[mail function]
SMTP=smtp.gmail.com
smtp_port=587
sendmail_from = your_email@gmail.com
sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"
```

- Sauvegardez et fermez le fichier.

Qu'est-ce que **sendmail.ini** et pourquoi le configurer ?

Le fichier **sendmail.ini** est utilisé pour configurer l'outil Sendmail, qui permet à PHP de communiquer avec un serveur SMTP. Si ce fichier n'est pas correctement configuré, les e-mails ne seront pas envoyés.

Configurer **sendmail.ini**

- Accédez au dossier **C:\xampp\sendmail**.
- Ouvrez (ou créez) le fichier **sendmail.ini** et ajoutez :

```
[sendmail]
smtp_server=smtp.gmail.com
smtp_port=587
smtp_ssl=auto
auth_username=your_email@gmail.com
auth_password=your_password
force_sender=your_email@gmail.com
```

- Utilisez un mot de passe d'application si 2FA est activé.

Qu'est-ce que 2FA et comment l'activer ?

L'authentification à deux facteurs (2FA) est une couche de sécurité supplémentaire qui protège votre compte en demandant une seconde preuve d'identité. Si 2FA est activé sur votre compte Gmail, vous devez générer un **mot de passe d'application** pour permettre à PHP d'envoyer des e-mails.

Comment générer un mot de passe d'application Gmail ?

1. Accédez à votre compte Google : [Google My Account](#).
2. Allez dans "Sécurité".
3. Dans la section "Connexion à Google", cliquez sur **Mots de passe d'application**.
4. Générez un mot de passe d'application spécifique à votre script PHP.
5. Remplacez votre mot de passe dans **sendmail.ini** par ce mot de passe d'application.

2. Envoi d'e-mail simple avec `mail()`

PHP propose une fonction `mail()` pour envoyer des e-mails directement. Cependant, cette méthode nécessite un serveur SMTP fonctionnel.

Exemple de script simple

Créer un fichier `mail_test.php` et ajouter :

```
<?php  
  
$to = "recipient@example.com";  
  
$subject = "Test Email from PHP";  
  
$message = "Bonjour, ceci est un e-mail de test envoyé depuis PHP !";  
  
$headers = "From: your_email@gmail.com";  
  
if(mail($to, $subject, $message, $headers)){  
    echo "E-mail envoyé avec succès !";  
} else {  
    echo "Échec de l'envoi de l'e-mail.";  
}  
  
?>
```

Redémarrer Apache

1. Ouvrez **XAMPP Control Panel**.
2. Cliquez sur **Stop** sur **Apache**, puis **Start**.
3. Testez l'envoi d'un e-mail.

Sécurisation des entrées utilisateur (Sanitizing)- Voir le cours

Lors de l'envoi d'e-mails en PHP, il est crucial de sécuriser les entrées utilisateur pour éviter les attaques par injection. Cette approche protège contre les attaques d'injection d'e-mail et garantit que seule une adresse valide est utilisée.

Deuxième Partie : Envoi avancé avec PHPMailer et SMTP

1. Qu'est-ce que Composer et pourquoi l'utiliser ?

Composer est un **gestionnaire de dépendances** pour PHP. Il permet d'installer facilement des bibliothèques tierces, comme **PHPMailer**, sans avoir à les télécharger et configurer manuellement.

Installation de Composer

1. Télécharger Composer :

- Rendez-vous sur getcomposer.org et téléchargez l'installateur.
- Pour Linux/macOS, utilisez la commande suivante :

```
curl -sS https://getcomposer.org/installer | php  
sudo mv composer.phar /usr/local/bin/composer
```

2. Installer Composer sur Windows :

- Ouvrez l'installateur et suivez les étapes.
- Vérifiez que **PHP est ajouté au PATH**.

3. Vérifier l'installation :

```
composer --version
```

- #### 4.
- Si Composer est bien installé, cette commande affichera sa version actuelle.

Vérifier que PHP est ajouté au PATH

Lorsque vous exécutez PHP à partir de la ligne de commande, votre système doit savoir où se trouve l'exécutable PHP. Cela est possible grâce à la variable d'environnement **PATH**, qui permet d'accéder à PHP depuis n'importe quel dossier du terminal.

Pourquoi ajouter PHP au PATH ?

- Permet d'exécuter des commandes PHP dans le terminal (ex: `php -v`).
- Nécessaire pour l'installation et l'utilisation de **Composer**.
- Évite les erreurs comme `php n'est pas reconnu comme une commande interne ou externe`.

Comment ajouter PHP au PATH sur Windows ?

1. **Trouvez le dossier PHP :**
 - Si vous utilisez XAMPP, PHP est souvent situé dans `C:\xampp\php`.
 2. **Ajouter PHP au PATH :**
 - Ouvrez **Paramètres Windows** et recherchez **Modifier les variables d'environnement**.
 - Dans **Variables système**, sélectionnez **Path** puis cliquez sur **Modifier**.
 - Cliquez sur **Nouveau** et ajoutez :
`C:\xampp\php`
 - Cliquez sur **OK** pour enregistrer.
 3. **Vérifier l'ajout :**
 - Ouvrez un terminal (CMD ou PowerShell) et tapez :
`php -v`
 - Si PHP est correctement ajouté, la version de PHP installée s'affichera.
-

2. Installation et utilisation de PHPMailer

Installer PHPMailer avec Composer

Ouvrir un terminal et se positionner dans `htdocs` :

```
cd C:\xampp\htdocs
```

1. **Installer PHPMailer :**
`composer require phpmailer/phpmailer`
2. **Vérifier l'installation :** Assurez-vous que le dossier `vendor/` contient **PHPMailer**.

Créer un script d'envoi d'e-mail avec PHPMailer

Créer un fichier `mail_smtp.php` et ajouter :

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'vendor/autoload.php';

$mail = new PHPMailer(true);

try {
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
```

```
$mail->SMTPAuth = true;
$mail->Username = 'your_email@gmail.com';
$mail->Password = 'your_app_password';
$mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
$mail->Port = 587;

$mail->setFrom('your_email@gmail.com', 'Your Name');
$mail->addAddress('recipient@example.com', 'Recipient Name');

$mail->isHTML(true);
$mail->Subject = 'Test Email via PHPMailer';
$mail->Body = 'Ceci est un test d\'envoi d\'e-mail via PHPMailer avec SMTP.';

$mail->send();
echo 'E-mail envoyé avec succès !';
} catch (Exception $e) {
    echo "L'e-mail n'a pas pu être envoyé. Erreur : {$mail->ErrorInfo}";
}
?>
```

Vous pouvez maintenant envoyer des e-mails avec PHP dans XAMPP, avec ou sans pièces jointes

Instructions de soumission

Chaque étudiant doit :

1. Soumettre un **rapport en PDF** contenant :
 - Captures d'écran de la configuration et du code PHP.
 - Captures d'écran de l'e-mail reçu.
2. Envoyer un e-mail avec l'objet : **Nom-Prénom-Filière - The Assignment**.
 - Exemple : **Qassimi-Sara-IRISI 1 - PHP Mail**
3. Joindre une **pièce jointe** à l'e-mail.
4. Envoyer l'e-mail à : **sara.qassimi@uca.ac.ma**

 **Date limite** : Vendredi 28 Février 2025 à 18H.