

Test email en local avec mailTrap et PHP Mailer

- 1) MAIL TRAP : <https://mailtrap.io/>
- 2) Créer un compte et rendez-vous sur votre Dashboard
- 3) Créer un projet -> puis myInbox
- 4) Cliquer sur la roue cranté (en haut à gauche) -> open myInbox Setting
- 5) Dans la fenêtre déroulante « Intégrations » sélectionner Zend Framework
- 6) Vous avez accès à vos settings pour PHP Mailer

PHP Mailer :

- 1) Rendez-vous à l'adresse suivante :
<https://github.com/PHPMailer/PHPMailer>

- 2) Installer composer :
<https://getcomposer.org/download/>

Puis cliquer sur le lien :

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

- 3) Ouvrir une console ligne de commande à la racine de votre projet (Alt + F12 sur PHP Storm) et entrer la commande suivante :

Composer init

- 4) Répondez aux questions

- 5) Ceci a pour effet de vous créer un fichier à la racine

composer.json

- 6) Ce dernier permet d'installer et d'utiliser des dépendances pour votre projet. Ex : PHP Mailer, Twig, etc...
- 7) Rendez-vous sur <https://packagist.org/> pour explorer les packages composer disponibles pour PHP
- 8) Exemple de composer.json

```
{
  "name": "QiwoGames/bon_coin_mic",
  "description": "Site fullstack annonce type le bon coin",
  "require": {
    "automattic/jetpack-autoloader": "^2.10",
    "carlosas/simple-event-dispatcher": "^0.1.1",
    "phpmailer/phpmailer": "^6.3",
    "ext-pdo": "*"
  },
  "authors": [
    {
      "name": "micpiwo",
      "email": "micpiwo@hotmail.fr"
    }
  ]
}
```

9) Installer PHP mailer via la ligne de commande :

composer require phpmailer/phpmailer

10) Les dépendances sont installées dans un dossier vendor
-> phpMailer

De nombreux développeurs qui écrivent des applications orientées objet créent un fichier source par définition de classe.

Un des plus gros inconvénients de cette méthode est d'avoir à écrire une longue liste d'inclusions de fichier de classes au début de chaque script : une inclusion par classe.

La fonction `spl_autoload_register()` enregistre un nombre quelconque de chargeurs automatiques, ce qui permet aux classes et aux interfaces d'être automatiquement chargées si elles ne sont pas définies actuellement.

En enregistrant des autochargeurs, PHP donne une dernière chance d'inclure une définition de classe ou interface, avant que PHP n'échoue avec une erreur.

11) Modifier les infos PHP Mailer avec les infos mailTrap Zend Framework

12) Dans votre classe, charger la classe autoload.php créer par composer init

require 'vendor/autoload.php';

13) La capacité de faire référence à un nom absolu avec un alias ou en important un espace de noms est stratégique. C'est un avantage similaire aux liens symboliques dans un système de fichiers.

PHP peut aliaser(/importer) les constantes, fonctions, classes, interfaces, et les espaces de noms.

Un alias est créé avec l'opérateur use. Voici un exemple qui présente les cinq types d'importation :

```
use PHPMailer\PHPMailer\PHPMailer;
```

```
use PHPMailer\PHPMailer\SMTP;
```

```
use PHPMailer\PHPMailer\Exception;
```

14) Utiliser l'exemple de classe simple depuis l'adresse suivante : <https://github.com/PHPMailer/PHPMailer>

15) Ajouter un appel à votre base de données et effectués des requête pour récupérer et affiche des données dans votre email

15) Créer un squelette HTML5 classique pour le contenu de votre email, vous pouvez utiliser des <table> ou autres.

16) Votre email ne peut pas inclure de fichier .css vous devez donc mettre vos styles soit directement dans les balises HTML5 ou à l'intérieur d'une balise <style> ici le css </style>

17) Dans un formulaire appeler votre classe phpMailer

```
require "classes/Reservation.php";
```

18) Créer une instance de la classe :

```
$emailValidation = new Reservation();
```

19)Créer un condision if{}else{} et appeler la methode conernée.

```
if (isset($_POST['validReservation'])) {  
    $emailValidation->reserverGite();  
    echo "<h3 class='alert-success p-3 text-danger'>Un email  
viens de vous etre envoyé, merci de verifié votre boite mail  
pour confirmer votre resevation</h3>";  
} else {  
    echo "<p class='alert-warning p-3'>Merci de remplir le  
formulaire avec votre email</p>";  
}
```

20)Pour rappel \$_POST['bouton'] est egale à :

```
<button type="submit" value="Reserver"  
name="validReservation" class="btn btn-outline-  
info">Envoyer</button>
```