

# **ADRAR\o/** **PÔLE NUMÉRIQUE**

- > **INFRASTRUCTURES** SYSTÈMES & RÉSEAUX
- > **CYBERSÉCURITÉ** INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > **TRANSFORMATION NUMÉRIQUE DES ENTREPRISES**

[www.adrar-numerique.com](http://www.adrar-numerique.com)

Nous allons utiliser une méthode viable en phase de développement, pour de la production, il faut utiliser le serveur lui-même et le configurer.

Tapez cette commande pour installer une nouvelle dépendance en dev uniquement:

```
$ composer require symfony/google-mailer --dev
```

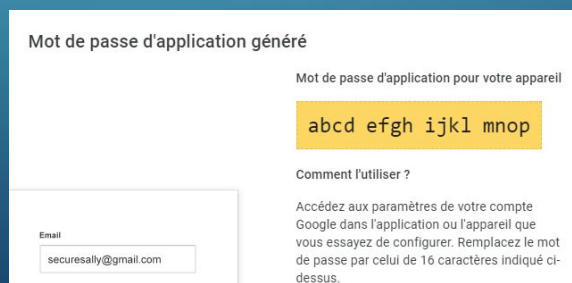
Une fois cela fait, créez un compte dédié à l'envoi de mails sur Google puis [rendez-vous ici](#).

Activez la validation en deux étapes (peu importe la méthode utilisée) puis revenez sur le même lien.

Un nouveau choix est apparu dans la section "Connexion à Google" et cliquez sur "Mots de passe des applications":

Dans la liste, sélectionnez "Autres" et mettez "Adrar classrooms" puis cliquez sur le bouton "Générer".

Copiez le code qui apparaît:



Mot de passe d'application généré

Mot de passe d'application pour votre appareil

abcd efgh ijkl mnop

Comment l'utiliser ?

Accédez aux paramètres de votre compte Google dans l'application ou l'appareil que vous essayez de configurer. Remplacez le mot de passe par celui de 16 caractères indiqué ci-dessus.

Email

seuresally@gmail.com

Ensuite dans votre fichier .env, cherchez la ligne où il y a "MAILER\_DSN" et mettez ceci en-dessous:

```
MAILER_DSN=gmail+smtp://marceaurodrigues@gmail.com:abcdefghijklmnpq@default
```

.env

L'email sur lequel on a  
créé le code

Le code généré plus tôt

Vous allez vous rendre sur cette page [MailHog](#) et récupérer la version *MailHog\_windows\_amd64.exe*.

Ensuite dans votre fichier .env, cherchez la ligne où il y a "MAILER\_DSN" et mettez ceci en-dessous:

```
MAILER_DSN=smtp://localhost:1025
```

.env



Lancez l'application que vous avez installé et rendez-vous dans un de vos contrôleurs pour ajouter une route pour l'expérience. Sur la nouvelle méthode créée/modifiée:

```
$email = (new Email())  
    ->from('symfony6@localhost')  
    ->to('marceau0707@gmail.com')  
    ->subject('Time for Symfony Mailer!')  
    ->text('Sending emails is fun again!')  
    ->html('<p>See Twig integration for better HTML integration!</p>');  
  
$mailer->send($email);  
  
return $this->render('adrrar/index.html.twig', [  
    'controller_name' => 'AdrrarController',  
    'email_sent' => true,  
]);
```

*BiduleController.php*

Vous devrez également ajouter deux **use** en dessous de ceux existants:

```
use Symfony\Component\Mailer\MailerInterface;  
use Symfony\Component\Mime\Email;
```

*BiduleController.php*

Et aussi ajouter ce paramètre à votre méthode:

```
MailerInterface $mailer
```

*BiduleController.php*

Il ne vous reste alors plus qu'à lancer votre application MailHog puis de lancer dans un terminal la commande suivante:

```
$ symfony console messenger:consume async
```

Dès lors, dès que vous actualisez votre page, un faux email sera envoyé.

**NB: MailHog permet de voir que si un vrai DSN était configuré, le mail serait bien partie sans pour autant polluer une boîte mail inutilement**