

Workshop n°5

SwiftmailerBundle: configuration et utilisation

Objectif

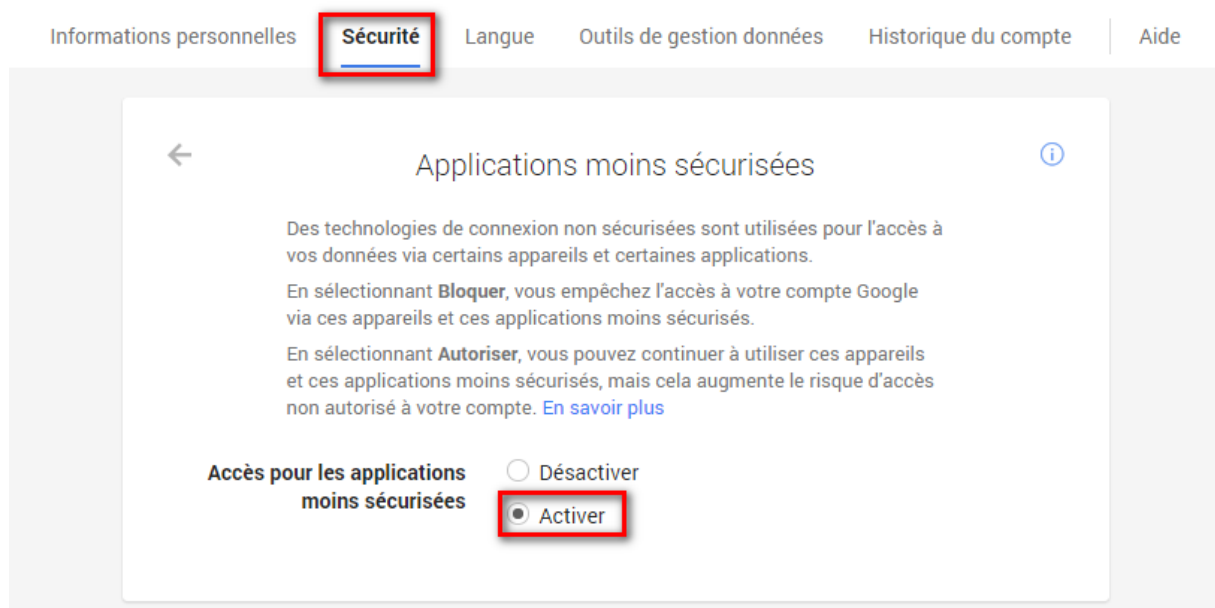
Le but de ce workshop est de configurer et d'utiliser le bundle Swiftmailer pour envoyer des e-mails

Création d'une boîte mail

Avant de commencer notre Workshop, vous devez créer une boîte gmail puis cliquer sur ce lien :

<https://www.google.com/settings/security/lesssecureapps>

Puis Configurer votre adresse mail afin d'être accéder par votre application et ceci en sélectionnant « activer »



Configuration de SwiftmailerBundle

Tout d'abord nous allons créer un nouveau projet intitulé WorkshopSwiftMailer puis nous créons un nouveau bundle intitulé MyAppMailBundle. Par la suite nous générons le contrôleur de ce bundle qui est MailController.

Ps : N'oubliez pas d'activer le bundle dans votre Kernel avant de l'utiliser

```

class AppKernel extends Kernel
{
    public function registerBundles()
    {
        $bundles = array(
            new Symfony\Bundle\FrameworkBundle\FrameworkBundle(),
            new Symfony\Bundle\SecurityBundle\SecurityBundle(),
            new Symfony\Bundle\TwigBundle\TwigBundle(),
            new Symfony\Bundle\MonologBundle\MonologBundle(),
            new Symfony\Bundle\SwiftmailerBundle\SwiftmailerBundle(),
            new Symfony\Bundle\AsseticBundle\AsseticBundle(),
            new Doctrine\Bundle\DoctrineBundle\DoctrineBundle(),
            new Sensio\Bundle\FrameworkExtraBundle\SensioFrameworkExtraBundle(),
            new Nomaya\SocialBundle\NomayaSocialBundle(),
            new MyApp\MailBundle\MyAppMailBundle(),
        );
    }
}

```

Après la phase de création du bundle, nous allons maintenant le configurer afin d'utiliser le SwiftMailerBundle. Allons-nous vers : [#app/config/parameters.yml](#) puis, Copiez la configuration nécessaire :

```

swiftmailer:
  transport: gmail
  host:      smtp.gmail.com
  username:  mail_crée_précédement
  password:  your_password
  port: 465
  disable_delivery: false
  encryption: ssl
  auth_mode: login

```

PS : le host change selon votre serveur mail.

Création d'un formulaire de contact

Nous allons créer un formulaire de contact par lequel un visiteur peut nous contacter. Le résultat de validation du formulaire est un mail d'accusé de réception envoyer par notre application.

Pour la phase de création du formulaire, nous irons créer :

- Une entité « Mail » dans le dossier Entity.
- Une classe « MailType » dans le dossier Form.
- Une vue « new.html.twig » dans le dossier ressources/views de notre bundle : cette vue permet l'affichage du formulaire.
- Une vue « mail.html.twig » » dans le dossier ressources/views de notre bundle : cette vue permet l'affichage d'un message de succès d'envoi.

1- Création de l'entité Mail

[# MyApp/MailBundle/Entity/Mail.php](#)

```

namespace MyApp\MailBundle\Entity;
class Mail {
    private $nom;
    private $prenom;
    private $tel;
    private $email;
    private $text;}

```

NB : N'oubliez pas de générer les getters et les setters.

2- Création du classe MailType

MyApp/MailBundle/Form/MailType.php

```
namespace MyApp\MailBundle\Form;

use Symfony\Component\Form\AbstractType;
use Symfony\Component\Form\Extension\Core\Type\IntegerType;
use Symfony\Component\Form\FormBuilderInterface;
use Symfony\Component\OptionsResolver\OptionsResolver;
use Symfony\Component\Form\Extension\Core\Type\TextType;
use Symfony\Component\Form\Extension\Core\Type\IntegerType;
use Symfony\Component\Form\Extension\Core\Type\EmailType;
use Symfony\Component\Form\Extension\Core\Type\TextareaType;
use Symfony\Component\Form\Extension\Core\Type\SubmitType;
class MailType extends AbstractType
{
    public function buildForm(FormBuilderInterface $builder, array $options)
    {
        $builder
            ->add('nom', TextType::class)
            ->add('prenom', TextType::class)
            ->add('tel', IntegerType::class)
            ->add('email', EmailType::class)
            ->add('text', TextareaType::class)
            ->add('valider', SubmitType::class);
    }

    public function getName()
    {
        return 'Mail';
    }
}
```

3- Ajout des actions dans notre contrôleur

Nous allons ajouter dans notre contrôleur quatre actions qui sont :

- index : qui permet d'afficher le formulaire de contact et envoyer un accusé de réception.
- success: qui permet d'afficher un message de succès si le mail est envoyé

a- Ajout de la fonction index

```
public function indexAction(Request $request)
{
    $mail = new Mail();
    $form= $this->createForm(MailType::class, $mail);
    $form->handleRequest($request);
    if ($form->isValid()) {
        $message = \Swift_Message::newInstance()
            ->setSubject('Accusé de réception')
            ->setFrom('espritplus2016@gmail.com')
            ->setTo($mail->getEmail())
            ->setBody(
                $this->renderView(
                    'MyAppMailBundle:Default:email.html.twig',
                    array('nom' => $mail->getNom(), 'prenom'=>$mail->getPrenom())
                ),
                'text/html'
            );
        $this->get('mailer')->send($message);
        return $this->redirect($this->generateUrl('my_app_mail_accuse'));
    }
}
```

```

    }

    return $this->render('MyAppMailBundle:Default:index.html.twig', array('form'=>$form
->createView()));
}

```

b- Ajout de la fonction success

```

public function successAction(){
    return new Response("email envoyé avec succès, Merci de vérifier votre adresse mail
.");
}

```

NB : N'oubliez pas d'ajouter « use Swift_Message; » «use
Symfony\Component\HttpFoundation\Request; » en haut du contrôleur.

4- Création des vues

Dans cette partie, nous allons créer deux vues qui sont :

- index.html.twig : c'est la vue qui affiche notre formulaire.
- mail.html.twig : c'est la vue qui affiche le message de succès.

MyApp/MailBundle/ressources/views

a- Création de la vue index.html.twig

```

<body>
<h2> <strong> Formulaire De Contact</strong> </h2>
<hr>
<h3><p> Contacter nous </p></h3>
{{ form_start (form) }}
<table>
    <tr>
        <td> Nom </td>
        <td> {{ form_widget (form.nom) }} </td>
    </tr>
    <tr>
        <td> Prenom </td>
        <td> {{ form_widget (form.prenom) }} </td>
    </tr>
    <tr>
        <td> Téléphone </td>
        <td> {{ form_widget (form.tel) }} </td>
    </tr>
    <tr>
        <td> Email </td>
        <td> {{ form_widget (form.email) }} </td>
    </tr>
    <tr>
        <td> Text </td>
        <td> {{ form_widget (form.text) }} </td>
    </tr>
</table>
{{ form_end (form) }}
</body>

```

a- Création de la vue email.html.twig

```
Bonjour Mr/Mme {{ nom }} {{ prenom }}  
<br>  
Merci d'avoir nous contacter
```

5- Configuration des routes

```
my_app_mail_homepage:  
  path:      /  
  defaults: { _controller: MyAppMailBundle:Mail:index }  
  
my_app_mail_accuse:  
  path:      /success  
  defaults: { _controller: MyAppMailBundle:Mail:success }
```

Vérification de l'envoi d'email

Vous pouvez vérifier votre boîte mail ! (boîte mail de réception)