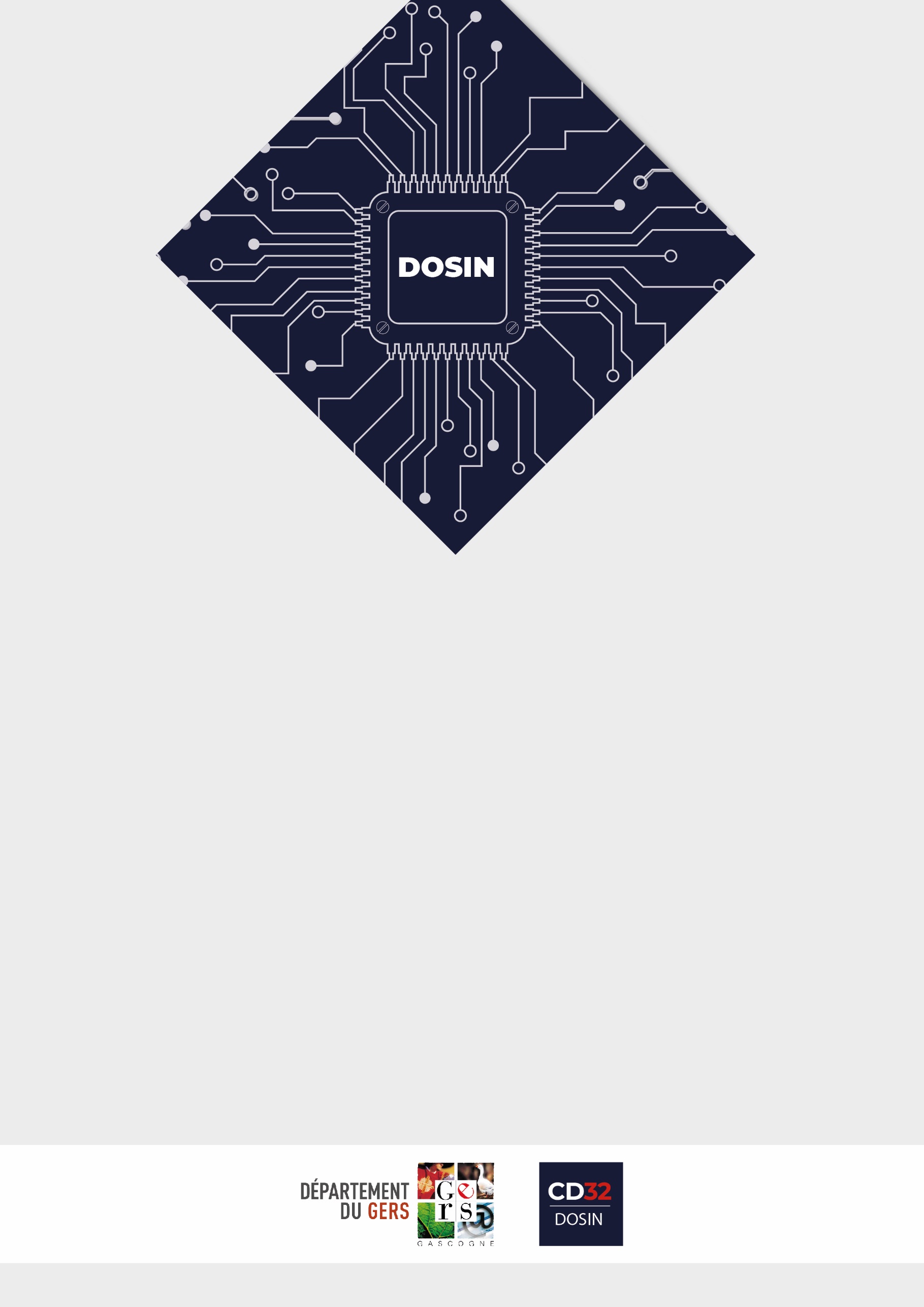
**News'32**



**Guide technique**

| **Propriétés** | |
| --- | --- |
| Objectif |  |
| Service | DOSIN – Service logiciels Métiers Cartographie – Pôle développement |
| Origine | Document2 |
| Créateur | LAFON-C |
| Date de création | 24/06/2022 |
| Date de dernière mise à jour | 22/07/2024 16:50:00 |
| Nombre Total de Pages | 5 |
| Statut | Brouillon / Validé |

| **Suivi des Versions** | | | | | |
| --- | --- | --- | --- | --- | --- |
| Version | Date | Rédacteur | Approbateur | Date Approbation | Modification |
| vf 1.00 | 24/06/2022 | C. LAFON |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

| **Documents Associés** | |
| --- | --- |
| Nom | Descriptif |
|  |  |
|  |  |
|  |  |

| **Diffusion** | | |
| --- | --- | --- |
| Structure \ Direction \ Service | Destinataires | Actions à Mener Pour Diffusion |
| CD32 |  | Conversion PDF et Enregistrement sous ? |
|  |  |  |

# Introduction

Le but de l'application est de faciliter et automatiser la gestion des bases de contacts utilisés dans les différents comptes SendinBlue.

Elle permet la création de listes de contacts suivant différents critères. Ces listes sont ensuite synchronisées tous les jours.

# Carte d'identité

|  |  |
| --- | --- |
| Nom | News'32 |
| Gitlab | newsletter (http://gitlab.cg32.local/dosin-poledev/newsletter) |
| Framework | Symfony 5 |
| Production | |
| Url | http://srv-php-appp.cg32.local:8094/accueil/ |
| Serveur | srv-php-appp |
| Port | 8094 |
| Base de données | srv-php-appp |
| Test | |
| Url | <http://srv-php-appt.cg32.local:8094> |
| Serveur | srv-php-appt |
| Port | 8094 |
| Base de données |  |

# Contexte technique

L'application est basée sur le framework Symfony 5.

Le lien avec Sendinblue se fait via les API Sendinblue

## API Sendinblue

Documentation de l'API : <https://developers.sendinblue.com/reference/>

Tous les appels à l'API sont centralisés dans la classe Service\CallApiService.php

Chaque compte sendinblue a un token d'identification, ce token est à utiliser lors de chaque appel API

**Problème de certificat SSL en local :**

Aller sur la page : <http://curl.haxx.se/ca/cacert.pem>

Enregistrer le certificat sur le poste

Dans le fichier php.ini, décommenter la ligne curl.cainfo et renseigner le chemin vers le certificat :

curl.cainfo = "c:/wamp64/www/cacert.pem"

Redemarrer symfony

Le chemin doit apparaitre dans le phpinfo

Texte de remplacement généré par une machine :
CURL support 
CURL Information 
Age 
Features 
AsynchDNS 
CharConv 
Debug 
GSS-Negotiate 
ION 
IPv6 
krb4 
Largefile 
libz 
NTLMWB 
SPNEGO 
SSL 
SSP' 
TLS-SRP 
HTTP2 
GSSAPI 
KERBEROS5 
UNIX SOCKETS 
PSL 
HTTPS PROXY 
MULTI SSL 
BROTLI 
Protocol s 
Host 
SSL Version 
ZLib Version 
libSSH Version 
curl.cainfo 
curl 
enabled 
7.700 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
Yes 
dict, file, ftp, tps, gopher, http, https, imap, imaps, Idap, Idaps, pop3, pop3s, rtsp, scp, sftp, smb, smbs, smtp, 
smtps, telnet tfip 
x86 ô4-pc-win32 
openSSL/1.1.1g 
libssh2Y1 go 
pem 
pem 

Si ce n'est pas le cas s'assurer qu'on est sur le bon php.ini

Texte de remplacement généré par une machine :
PHP Version 7.4.9 
System 
Build Date 
Compiler 
Architecture 
Configure Command 
Server API 
Virtual Directory Support 
Configuration File (php.ini) Path 
Loaded Configuration File 
php 
Windows NT 16-010901 100 build 19044 (Windows 10) AMD64 
Aug 4 2020 11 45:36 
Visual 2017 
xô4 
cscript Inologo /e:jscript configure.js "—enable-snapshot-build" "—enable-debug-pack' 
builcfideps 12 Ilsdk,shared" 
builffdeps 12 "—enable-com- 
dotnet:shared" —without-anatyzer. "—with-pgo" 
CGI/FastCGl 
enabled 
C:kWlNDOWS 
C:iwamp64biniphpIphp7 4 

[How to fix cURL error 60: SSL certificate problem | by Narendra Vaghela | Medium](https://medium.com/@narendravaghela/how-to-fix-curl-error-60-ssl-certificate-problem-80e7dafafa57)

## Logs

composer require symfony/monolog-bundle

Création du fichier config/packages/monolog.yaml

|  |
| --- |
| monolog:  channels: ['db']  handlers:  dbHandler:  channels: ['db']  type: service  id: dbHandler |

Dans service.yaml, déclarer le handler et le processor :

|  |
| --- |
| monolog.dbHandler:  class: App\Utility\DbHandler  monolog.dbProcessor:  class: App\Utility\DbProcessor  tags:  - {name: monolog.processor, handler: dbHandler} |

Création de la classe src\Utility\DbProcessor qui va s'executer avant le handler. C'est dans le processor qu'on peut récupérer des informations de la requête http (ip, utilisateur…) :

|  |
| --- |
| <?php  namespace App\Utility;  use Symfony\Component\HttpFoundation\RequestStack; use Symfony\Component\Security\Core\Security;  class DbProcessor {  private $request;  private $security;   //RequestStack : renvoie l'ensemble des requêtes qui ont été émises  public function \_\_construct(RequestStack $request, Security $security)  {  $this->request = $request->getCurrentRequest();  $this->security = $security;  }   public function \_\_invoke(array $record)  {  //on modifie le $record pour ajouter nos infos  $record['extra']['clientIp'] = $this->request->getClientIp();  $record['extra']['utilisateur'] = $this->security->getUser();  return $record;  } } |

Création de la classe src\Utility\DbHandler qui va se charger d'enregistrer les logs en Bdd :

|  |
| --- |
| <?php  namespace App\Utility;  use App\Entity\Log; use Doctrine\ORM\EntityManagerInterface; use Monolog\Handler\AbstractProcessingHandler; use Monolog\Logger;  class DbHandler extends AbstractProcessingHandler {  private $manager;   public function \_\_construct($level = Logger::*DEBUG*, bool $bubble = true, EntityManagerInterface $manager)  {  parent::*\_\_construct*($level, $bubble);  $this->manager = $manager;  }   protected function write(array $record): void  {  //Enregistrement des logs en bdd  $log = new Log();  $log->setContext($record['context']);  $log->setLevel($record['level']);  $log->setLevelName($record['level\_name']);  $log->setMessage($record['message']);   $this->manager->persist($log);  $this->manager->flush();  } } |

https://symfony.com/doc/5.4/logging/monolog\_email.html

## Cron de synchronisation

Lancement des synchronisations à 6h et à 12h

/var/spool/cron/root

|  |
| --- |
| 0 6 \* \* \* sh /var/www/html/news32.sh  0 12 \* \* \* sh /var/www/html/news32.sh |

Contenu du fichier news32.sh :

|  |
| --- |
| php /var/www/html/newsletter/bin/console synchro |

Le script de synchronisation est dans src/Command/Synchronisation.php

Le nom de la commande à lancer (ici "synchro") est défini dans la fonction configure() et le processus métier dans la fonction execute()

Pour le lancer en test dans la console : php bin/console synchro

Documentation Symfony : <https://symfony.com/doc/current/console.html>

# Sécurité

## SSO

La configuration pour le SSO se fait dans le fichier config > packages > security.yaml

|  |
| --- |
| security:  password\_hashers:  App\Entity\Utilisateur:  algorithm: auto  role\_hierarchy:  ROLE\_ADMIN: [ ROLE\_USER ]  # https://symfony.com/doc/current/security.html#where-do-users-come-from-user-providers  # providers : où sont récupérées les données des utilisateurs  providers:  # users\_in\_memory: { memory: null }  # Pour connexion LDAP  my\_ldap:  ldap:  service: Symfony\Component\Ldap\Ldap  base\_dn: OU=CD32,DC=cg32,DC=local  search\_dn: "civitas\_ldap"  search\_password: JslmdpLRH!  default\_roles: ROLE\_USER  # Fin pour connexion LDAP  users:  entity:  class: App\Entity\News\Utilisateur  property: login  users\_sso:  entity:  class: App\Entity\News\Utilisateur  property: sso  firewalls:  dev:  pattern: ^/(\_(profiler|wdt)|css|images|js)/  security: false  main:  # Pour connexion LDAP  # anonymous: true  # provider: users  # form\_login\_ldap:  # login\_path: /login  # check\_path: /login  # search\_dn: "civitas\_ldap"  # search\_password: 'JslmdpLRH!'  # service: Symfony\Component\Ldap\Ldap  # dn\_string: 'dc=cg32, dc=local'  # query\_string: '(&(objectClass=user)(samAccountName={username})(cn=\*))'  # success\_handler: redirect.after.login  # Fin pour connexion LDAP  #user\_checker : vérifie que l'utilisateur est bien actif lors de la connexion  user\_checker: App\Security\UserChecker  # Pour connexion SSO  entry\_point: App\Security\AuthenticationEntryPoint  remote\_user:  provider: users\_sso  user: REMOTE\_USER  # Fin pour connexion SSO  logout:  path: app\_logout  # Pour connexion LDAP  # target: app\_login  # Fin pour connexion LDAP  # Easy way to control access for large sections of your site  # Note: Only the \*first\* access control that matches will be used  access\_control:  - { path: ^/login, roles: IS\_AUTHENTICATED\_ANONYMOUSLY }  - { path: ^/noacces, roles: IS\_AUTHENTICATED\_ANONYMOUSLY }  # - { path: ^/param/, roles: ROLE\_ADMIN }  - { path: ^/, roles: ROLE\_USER } |

Sur le serveur :

etc/http/conf.d/default.conf

|  |
| --- |
| Listen 8094  <VirtualHost \*:8094>  ServerName newsletter  DocumentRoot /var/www/html/newsletter/public  <Directory /var/www/html/newsletter/public>  AllowOverride All  Order Allow,Deny  Allow from All  <IfModule mod\_rewrite.c>  Options -MultiViews  RewriteEngine On  RewriteCond %{REQUEST\_FILENAME} !-f  RewriteRule ^(.\*)$ app.php [QSA,L]  </IfModule>  </Directory>  ErrorLog /var/log/newsletter\_error.log  CustomLog /var/log/newsletter\_access.log combined    <Location />  AuthType Kerberos  AuthName "Auth. required."  KrbAuthRealms CG32.LOCAL  KrbServiceName HTTP/srv-php-appt.cg32.local@CG32.LOCAL  Krb5Keytab /etc/httpd/conf/keytabs/sso-php-t.keytab  require valid-user  </Location>  </VirtualHost> |

Si l'utilisateur n'est pas connu par l'application :

Création d'une page no\_acces accessible aux personnes non authentifiées

C'est la fonction start() de la classe App\Security\AuthenticationEntryPoint qui gère la redirection en cas de problème d'authentification

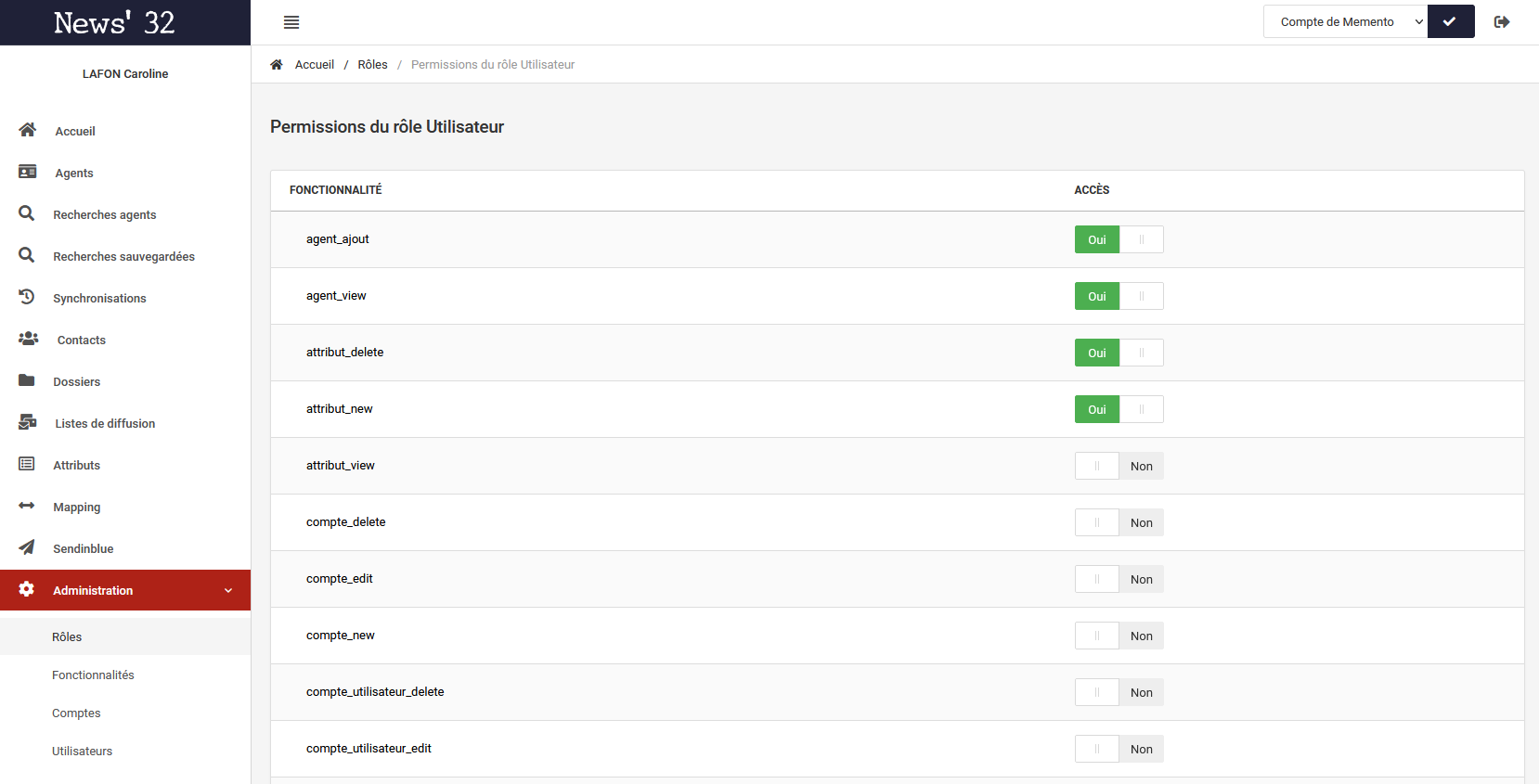
## Droits d'accès

En base de données :

- la table "fonctionnalite" liste toutes les fonctionnalités de l'application (ex : contact\_new -> ajout d'un contact)

- la table "role", liste tous les rôles de l'application (ROLE\_ADMIN, ROLE\_USER…)

- la table "permission", liste les accès pour chaque rôle et fonctionnalité

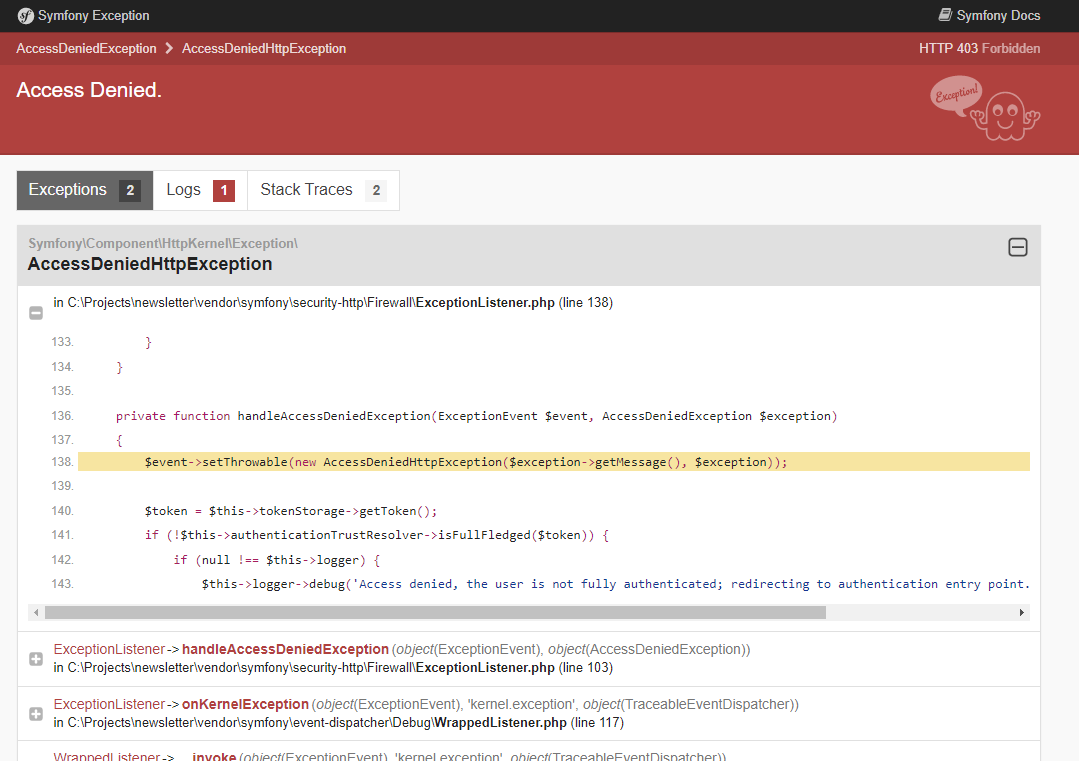


Le contrôle des accès se fait dans Security > Voter > GenericVoter.php

Puis dans chaque méthode des controller qui correspond à une fonctionnalité, via denyAccessUnlessGranted :

*/\*\*  
 \** ***@Route****("/attributs/new", name="attribut\_new", methods={"GET","POST"})  
 \*/*public function new(Request $request, CallApiService $callApiService, TranslatorInterface $translator): Response  
{  
 $this->denyAccessUnlessGranted('attribut\_new');  
 $attribut = new Attribut();  
 $form = $this->createForm(AttributType::class, $attribut);  
 $form->handleRequest($request);

Si un utilisateur essaie d'accéder à une page pour laquelle il n'a pas les droits, une erreur 403 Access Denied est levée



Le rôle administrateur a tous les droits

Documentation Symfony : <https://symfony.com/doc/current/security/voters.html>

## Usurpation d'identité (impersonation)

Si la connexion LDAP est utilisée sur l'application, il est possible d'usurper l'identité d'un autre utilisateur.

Security.yaml

|  |
| --- |
| security:  […]  role\_hierarchy:  ROLE\_ADMIN: [ ROLE\_USER, ROLE\_ALLOWED\_TO\_SWITCH ]  […]  firewalls:  main:  switch\_user: true  […] |

Dans l'url rajouter : ?\_switch\_user=login

Pour quitter ce mode ?\_switch\_user=\_exit

<https://symfony.com/doc/current/security/impersonating_user.html#how-to-impersonate-a-user>

En SSO ce n'est pas possible car l'authentification est "rechargée" à chaque appel au serveur

# Ergonomie / Mise en page

## Pages d'erreur

Les pages d'erreur personnalisées ne sont "activées" qu'en mode production

Elles sont à placer dans templates > bundles > TwigBundle > Exception

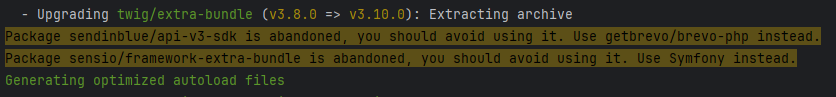
Nommer les pages : "errorNUMERO.html.twig", ex : error404.html.twig

# Migrations

## Migration en 5.4.41

La v5.4.41 est la dernière version de Symfony maintenue avec PHP 7

composer update



### Remplacement du package Sendinblue par Brevo

composer require getbrevo/brevo-php

### Suppression de sensio/framework-extra-bundle

<https://numa-bord.com/miniblog/symfony-se-debarrasser-de-sensio-framework-extra-bundle/>

### Mise à jour de dompdf

Modification de la version dans composer.json puis

composer update

Astuce :

composer audit

Donne les vulnérabilités des dépendances du projet