

DataXchange

...

Back-office

Sommaire



I. Introduction

II. Présentation de la mission

III. La mission

IV. Résultats

V. Conclusion

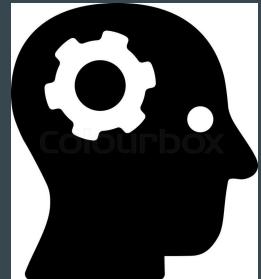
I. Introduction



II. Présentation de la mission

A. Zento

- Fondée par Yoann Amsellem et Jérémy Amsellem en 2016
- Conseil en systèmes et logiciels informatiques
- R&D : Plateforme DataXchange

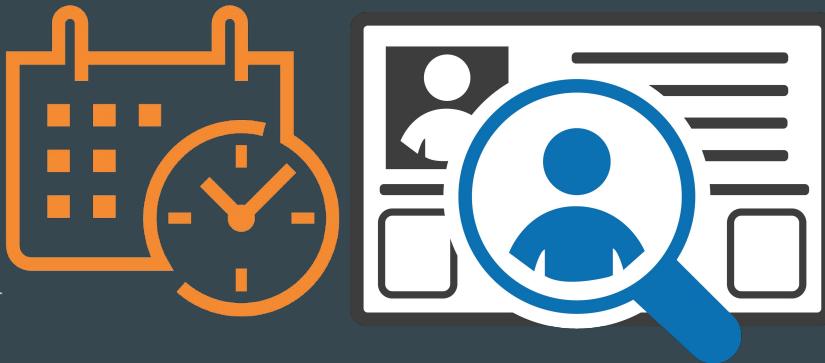


II. Présentation de la mission

A. Besoins clients

1. Objectifs premiers

- Création d'un back-office supervision
- Gestion de logs : Exceptions Java
- Lisibilité : Répartition des erreurs
- Traçabilité : Date d'insertion selon les bases de données
- Maintenabilité : Sous-traitance prévue



II. Présentation de la mission

A. Besoins clients

2. Technologies

- Serveur de prod : API JAVA
- Framework : Symfony 4
- Base de données : MySQL
- Front : Bootstrap / CSS



Java™



CSS



III. La mission

A. Analyse de l'existant

1. Etude du fonctionnement du site
 - Technologie Java
 - Architecture 3-Tiers
 - Génération de logs en fichiers texte



III. La mission

A. Analyse de l'existant

2. Les informations à traiter

- Logs de différents niveaux de criticités
- Exceptions de différents types

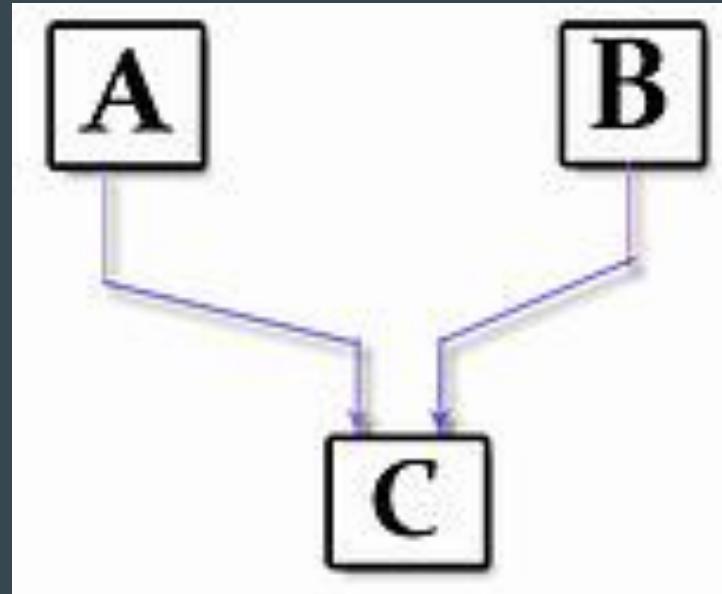


III. La mission

A. Analyse de l'existant

3. Les entités

- Entités imbriquées
- Total de 15 entités



III. La mission

A. Analyse de l'existant

4. Challenges rencontrés
- Gestion des clés étrangères
 - Gestion des “IncludedException”



III. La mission

B. Approche technique et organisation projet

1. Méthode de travail Agile
 - Sprint de 5 jours
 - Rédaction d'un backlog
 - Création d'un planning de Gantt



	ID	Titre	Description	Quand ? ou après quoi ?	Etat
2	1	Enlever les div des vues	Enlever les div des pages de vues	Après le parsing	A faire
3	2	Scripts Popper Bootstrap	Changer les chemins de scripts par ceux des dépendances téléchargées	Avant la mise en forme de toutes les vues	A faire
4	3	Creation de la BDD	Creation de la BDD et des tables avec les FK	Avant le Parsing	Terminé
5	4a	Parsing	Creation de la fonction de parsing	Avant l'enlèvement des div dans les vues	En cours
6	4b	Ecriture en BDD	Insertion des infos parsées en BDD	Après Parsing	Terminé
7	5	Appel des infos global	Creation des méthodes pour récupérer les infos insérées dans la BDD grâce au Parsing	Après l'enlèvement des div dans les vues	Terminé
8	19	Filtres	Creation des filtres et barre de recherche Symfony	Après Scripts Pooper Bootstrap	A faire
22	20	Mise en forme des vues	Creation de la mise en forme des vues en .scss	Après Filtres	A faire
23	21	Authentification	Creation du login / mdp	Avant la mise en Prod	Terminé
24	22	Formatage à changer	ATTENTION aux éléments de ponctuation permettant de séparer les différents éléments	Avant Ecriture en BDD	Terminé
25	23	Validation	Valider les tests de parsing de données brut et insertion en base	Après les tests de la BDD	Terminé
26	24	Finder	Trouver le moyen de détecter un nouveau fichier dans un répertoire donné		Terminé
27	25	Finder	Puis trouver le moyen de lire son contenu et l'envoyer au parser et l'insérer en base		Terminé
28	26	Cron	Moyen d'exécuter le détection toutes les minutes		Terminé
29	27				

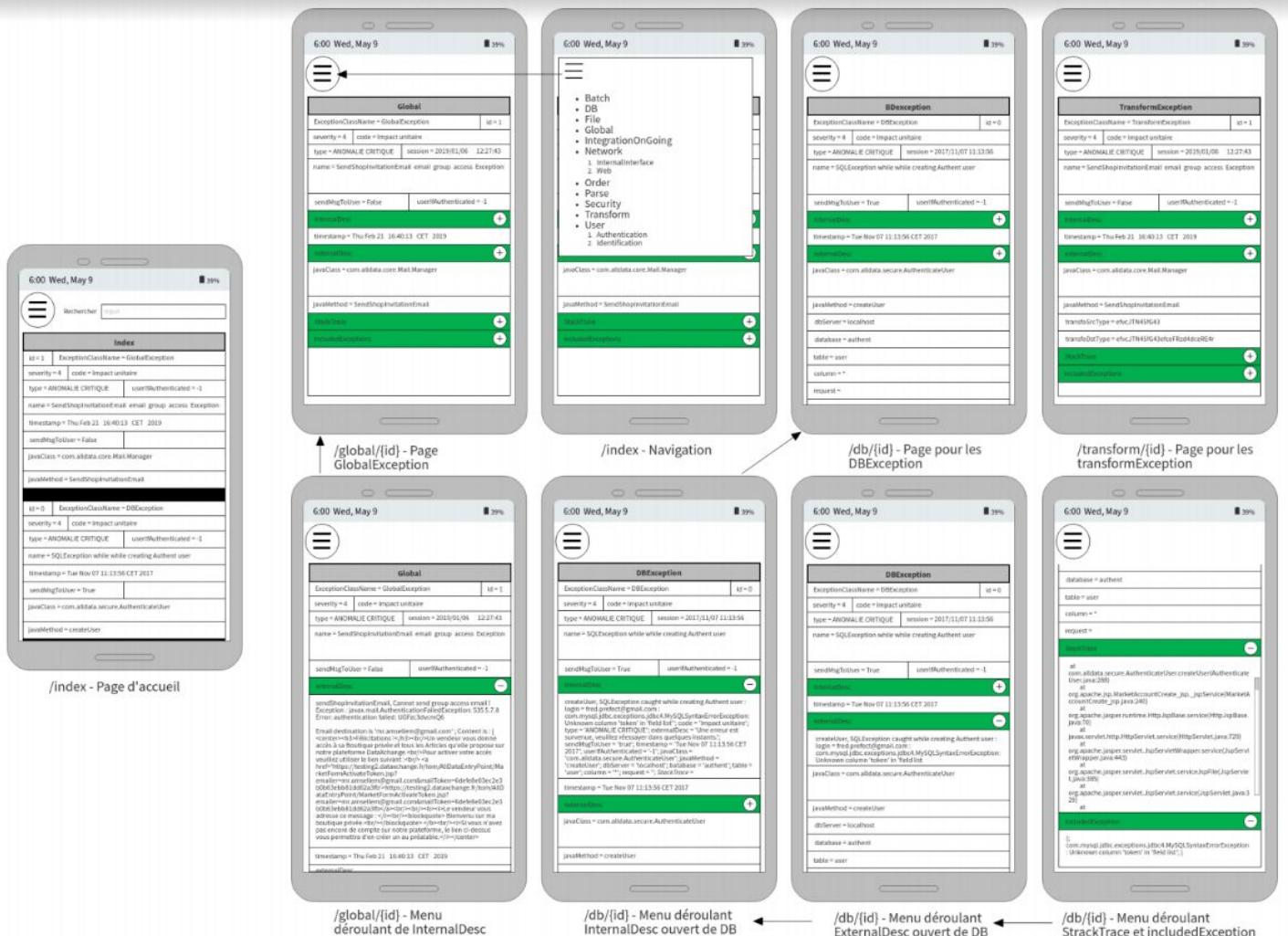
III. La mission

B. Approche technique et organisation projet

2. Projet Gantt

- Réalisation en cascade
- Modélisation
- Maquette





III. La mission

B. Approche technique et organisation projet

3. Modélisation Merise

- Classe générale “Global”
- Classe filles de “Global”
- Classe fille et de “InternalInterface”
- Clés récursives “Included”



Included N°1		Global	
id	1	id	1
previd_id	NULL	ExceptionClassName	GlobalException
nextid_id	2	code	STR NN
finalrow	NULL	name	text NN
RefGlobalException	1	type	str NN
isGlobalException	TRUE	session	datetime NN
Included N°2		userifauthenticated	int NN
id	2	sendmsgtouser	BOOL NN
previd_id	1	timestamp	datetime NN
nextid_id	NULL	internaldesc	text NN
finalrow	TRUE	externaldesc	text NN
RefGlobalException	1	javaclass	text NN
isGlobalException	TRUE	javamethod	text NN
		insertdatetime	datetime NN

0:1

0:n

0:1

0:1

III. La mission

B. Approche technique et organisation projet

4. Git

- Lien vers GitHub
- Un seul collaborateur



III. La mission

C. Choix techniques

1. Back-end

a. Le langage de développement PHP 7

- Utilisation de fonctions natives
- b. Le framework Symfony 4
- Paquet website-skeleton
- Les services



III. La mission

c. L'ORM Doctrine

- i. Lien à la base et création des entités
- Connexion à la BDD
- Création d'entités
- Création des fichiers de migrations
- Migration des entités dans la BDD



```
DATABASE_URL=mysql://root:████████@127.0.0.1:3306/zentoalldata?serverVersion=5.7
```

```
$ php bin/console make:entity

Class name of the entity to create or update:
> Product

New property name (press <return>          to stop adding fields):
> name

Field type (enter ? to see all types) [string]:
> string

Field length [255]:
> 255

Can this field be null in the database (nullable) (yes/no) [no]:
> no

New property name (press <return>          to stop adding fields):
> price

Field type (enter ? to see all types) [string]:
> integer

Can this field be null in the database (nullable) (yes/no) [no]:
> no

New property name (press <return>          to stop adding fields):
>
(press enter again to finish)
```

```
$ php bin/console make:migration
```

```
$ php bin/console doctrine:migrations:migrate
```

III. La mission

c. L'ORM Doctrine

- ii. Gestion des insertions en BDD
 - Utilisation de l'entityManager
 - Setter des champs des tables
 - Informations spécifiques liées au typage



```
/**  
 * @var EntityManagerInterface  
 */  
private $entityManager;  
  
public function __construct(EntityManagerInterface $entityManager)  
{  
    // parent::__construct($entityManager);  
    $this->entityManager = $entityManager;  
}  
  
public function dbInsertion($result)  
{  
    $e = new GlobalException();  
    $e->setExceptionclassname($result['ExceptionClassName']);  
    $e->setSession(\DateTime::createFromFormat('Y/m/d H:i:s', $result['session']));  
    $e->setExceptionid($result['id']);  
    $e->setSeverity($result['severity']);  
    $e->setName($result['name']);  
    $e->setInternaldesc($result['internalDesc']);  
    $e->setCode($result['code']);  
    $e->setType($result['type']);  
    $e->setExternaldesc($result['externalDesc']);  
    $e->setSendmsgtouser($result['sendMsgToUser']);  
    $e->setTimestamp($result['timestamp']);  
    $e->setUserifauthenticated($result['userIfAuthenticated']);  
    $e->setJavaclass($result['javaClass']);  
    $e->setJavamethod($result['javaMethod']);  
    $e->setStacktrace($result['StackTrace']);
```

```
switch ($result['ExceptionClassName'])  
{  
  
    case "InternalInterfaceException":  
        $iI = new InternalInterfaceException();  
        $iI->setRefGlobalException($e);  
        $iI->setIpsrc($result['ipSrc']);  
        $iI->setIpdst($result['ipDst']);  
        $iI->setPort($result['port']);  
        $iI->setActor1($result['actor1']);  
        $iI->setActor2($result['actor2']);  
  
        $this->entityManager->persist($iI);  
        $this->entityManager->flush();  
        break;  
  
    default :  
        echo "abnormal";  
        break;  
}  
}
```

```
case "WebException":  
    $iI = new InternalInterfaceException();  
    $iI->setRefGlobalException($e);  
    $iI->setIpsrc($result['ipSrc']);  
    $iI->setIpdst($result['ipDst']);  
    $iI->setPort($result['port']);  
    $iI->setActor1($result['actor1']);  
    $iI->setActor2($result['actor2']);  
  
    $web = new WebException();  
    $web->setInternalinterfaceexceptionId($iI);  
    $web->setHttpcode($result['httpCode']);  
    $web->setVerb($result['verb']);  
    $web->setUrl($result['url']);  
    $web->setHeader($result['header']);  
    $web->setBody($result['body']);  
  
    $this->entityManager->persist($web);  
    $this->entityManager->flush();  
    break;
```

III. La mission

C. Choix techniques

1. Back-end

c. L'ORM Doctrine

iii. Challenges rencontrés



MySQL®

```
/**  
 * @ORM\Column(name="insertdatetime", type="datetime", options={"default"="CURRENT_TIMESTAMP"})  
 */  
private $insertdatetime;
```

```
/**  
 * @ORM\Entity(repositoryClass="GlobalExceptionRepository")  
 * @ORM\HasLifecycleCallbacks()  
 */  
class GlobalException
```

```
/**  
 * @ORM\PrePersist  
 */  
public function setInsertdatetime(): self  
{  
    $this->insertdatetime = new \DateTime(); //  
  
    return $this;  
}
```

III. La mission

d. Parser PHP

- Plusieurs explosions du contenu
- Répétition de l'action tant qu'il reste des informations
- Suppression des éléments inutiles



```
class TxtParsingController extends AbstractController
{
    public function parseContent(string $txt): array
    {
        $txtexplode = explode( delimiter: ";", string: "$txt");
        $i = 0;
        $t[] = '';

        for ($i; $i < sizeof($txtexplode); $i++) {
            $distinctElem = explode( delimiter: "==>", string: "$txtexplode[$i]");

            if (sizeof($distinctElem) > 1)
            {
                $t[trim($distinctElem[0])] = trim($distinctElem[1], charlist: "\n");
            }
            else
            {
                echo "\n skipping element in parseContent, no Key/value pair !! \n ";
            }
        }
        return $t;
    }
}
```

III. La mission

e. Finder Component

- Différentes manières de sélectionner les fichiers
- Méthode de vérification de présence de fichiers
- Méthode de lecture du contenu



```
/* START FINDER */
echo "<br/><br/>GO FILES FINDER<br/><br/>";
$finder = new Finder();

// find all files in Files directory
$finder->files()->in( dirs: '/home/auzou/Zento/Alldata/src/Files');

// check if there are any search results in Files directory
if ($finder->hasResults())
{
    // Read contents of return files
    foreach ($finder as $file)
    {
        $txt = $file->getContents();
        var_dump($txt);
        echo "<br/><br/>END FILES FINDER<br/><br/>";
    }
    /* END FINDER */
```

III. La mission

f. Déplacement du fichier après traitement

- Changement de l'extension du fichier

Ou

- Changement
des fichiers traités



```
// find all files in Files directory  
$finder->files()->in( dirs: '/home/auzou/Zento/Alldata/src/Files');
```

```
$uploadsDir = '/home/auzou/Zento/Alldata/src/Files/Treated/';  
// $toMove = $finder->files()->in('/home/auzou/Zento/Alldata/src/  
rename( oldname: $file->  
    getPath() . '/' . $file->  
    getFilename(), newname: $uploadsDir . $file->  
    getFilename()  
);
```

III. La mission

g. Exécution périodique du traitement des exceptions

- Création du Cron
- Utilisation de l'entité Command
- Définition de la propriété “app:autoscan”

Cron Job
Runner



```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
# indicating with different fields when the task will be run  
# and what command to run for the task  
#  
# To define the time you can provide concrete values for  
# minute (m), hour (h), day of month (dom), month (mon),  
# and day of week (dow) or use '*' in these fields (for 'any').#  
# Notice that tasks will be started based on the cron's system  
# daemon's notion of time and timezones.  
#  
# Output of the crontab jobs (including errors) is sent through  
# email to the user the crontab file belongs to (unless redirected).  
#  
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
*/1 * * * * /usr/bin/php7.2 /home/auzou/Zento/Alldata/bin/console app:AutoScan
```

```
class AutoScanCommand extends Command//extends ContainerAwareCommand//  
{  
    protected static $defaultName = 'app:autoscan';  
    /**  
     * @var EntityManagerInterface  
     */  
    private $entityManager;  
  
    public function __construct(EntityManagerInterface $entityManager)  
    {  
        parent::__construct();  
        $this->entityManager = $entityManager;  
    }  
  
    protected function execute(InputInterface $input, OutputInterface $output)  
    {  
        $output->writeln([  
            'Try use (new AutoScanCommand($this))->scanNow() with php bin/console  
            =====', // Another line  
            '', // Empty line  
        ]);  
  
        $this->scanNow();  
  
        $output->writeln([  
            =====, // Another line  
            'My Final Symfony command', // A line  
            '', // Empty line  
        ]);  
    }  
}
```

```
function scanNow(): void  
{  
    $g = new FinderFiles($this->entityManager);  
    $g->findFiles();  
}
```

III. La mission

h. Sécurité

- Création du formulaire de Login
- Modification des paramètres de “security.yaml”
- Modification de “Loginformauthenticator”
- Intégration du “loginform” dans la vue
- Modification des routes dans les annotations



```
$ php bin/console make:auth

What style of authentication do you want? [Empty authenticator]:
[0] Empty authenticator
[1] Login form authenticator
> 1

The class name of the authenticator to create (e.g. AppCustomAuthenticator):
> LoginFormAuthenticator

Choose a name for the controller class (e.g. SecurityController) [SecurityController]:
> SecurityController

Do you want to generate a '/logout' URL? (yes/no) [yes]:
> yes

created: src/Security/LoginFormAuthenticator.php
updated: config/packages/security.yaml
created: src/Controller/SecurityController.php
created: templates/security/login.html.twig
```

```
# 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: ^/admin, roles: ROLE_ADMIN }
    # - { path: ^/profile, roles: ROLE_USER }
    - { path: ^/login$, roles: IS_AUTHENTICATED_ANONYMOUSLY }
```

```
firewalls:
    dev:
        pattern: ^/(_(profiler|wdt)|css|images|js)/
        security: false
    main:
        anonymous: true
        guard:
            authenticators:
                - App\Security\LoginFormAuthenticator
        logout:
            path: app_logout
```

```
public function onAuthenticationSuccess(Request $request, TokenInterface $token, $providerKey)
{
    if ($targetPath = $this->getTargetPath($request->getSession(), $providerKey)) {
        return new RedirectResponse($targetPath);
    }

    // For example : return new RedirectResponse($this->urlGenerator->generate('some_route'));
    return new RedirectResponse($this->urlGenerator->generate( name: 'home_index'));
}

protected function getLoginUrl()
{
    return $this->urlGenerator->generate( name: 'app_login');
}
```

```
$ php bin/console make:user
```

```
The name of the security user class (e.g. User) [User]:
```

```
> User
```

```
Do you want to store user data in the database (via Doctrine)? (yes/no) [yes]:
```

```
> yes
```

```
Enter a property name that will be the unique "display" name for the user (e.g.  
email, username, uuid [email])
```

```
> email
```

```
Does this app need to hash/check user passwords? (yes/no) [yes]:
```

```
> yes
```

```
created: src/Entity/User.php
```

```
created: src/Repository/UserRepository.php
```

```
updated: src/Entity/User.php
```

```
updated: config/packages/security.yaml
```


III. La mission

C. Choix techniques

2. Front-end

- a. Le web mobile Bootstrap
- b. Pré-processeur SCSS

```
@import "bootstrap/scss/bootstrap";  
@import "general.scss";  
@import "nav.scss";
```



CSS



III. La mission

C. Choix techniques

3. Tests et validations



```
auzou@auzou-ThinkPad-L450:[~/Zento/Alldata] (master=) $php bin/console app:AutoScan
Try use (new AutoscanCommand($this))->scanNow() with php bin/console app:AutoScan
=====
<br/><br/>GO FILES FINDER<br/><br/>=====
My Final Symfony command
```

Time: 131 ms, Memory: 4.00 MB

OK (1 test, 23 assertions)

IV. Résultats

A. Fin du projet

- La majorité des fonctions prévues à l'origine sont réalisées

B. Axes d'amélioration de l'existant

- Front à faire + tri et filtre

C. Développements futur

- Ajout des logs d'informations + ajout de tables

V. Conclusion

