

Groupe : Valentino N’ZITA, Hakim, Billy

Vetux Line

Niveau : Deuxième année de BTS SIO SLAM

Projet

Dans ce projet nous travaillons pour l'entreprise Vetux-Line qui est une entreprise de création de vêtements. Vetux-Line reçoit tous les deux mois 2 fichiers client Csv. Ce projet a pour but de créer une application pour l'entreprise qui va nous permettre de fusionner deux fichiers CSV en un seul. Pour effectuer ceci notre entreprise nous transmet deux fichiers Csv qui se nomment: french-client.csv (avec 3000 clients) et un german-client.csv (avec 2000 clients)

Première Partie

Pour cette mission nous avons pour but de :

- Créer un formulaire d'authentification
- Upload les deux fichiers csv
- Filtrer les données renvoyées (genre, titre, nom, prénom, email, date de naissance, num tel, CCType, CCNumber, CVV2, CCExpaires, adresse physique (plusieurs colonnes), taille, poids, véhicule, coordonnées GPS.)
- Fusionner les deux fichiers Csv
- Une fois les deux fichiers fusionnés les télécharger

Pour cette Partie 1 nous avons réduit le volume des deux fichiers csv et n'avons pas modifié leur nom, small-french-client.csv (12 clients) et small-german-client.csv (8 clients)

Première tâche à effectuer : Login Register

Nous avons créé un formulaire d'authentification qui nous a permis de nous inscrire et de nous connecter en tant qu'admin ou en tant que simple membre ou bien de nous déconnecter, l'Admin après s'être connecté va pouvoir uploader deux fichiers Csv. Tandis qu'un utilisateur va pouvoir modifier son compte c'est-à-dire modifier son mot de passe ou son nom d'utilisateur ou supprimer son compte.

Deuxième Tâche

Pour commencer cette partie nous avons décidé de créer un formulaire pour upload pour cela dans le service.yaml nous avons défini un paramètre qui contient le nom du répertoire où les images doivent être téléchargées ; upload_dir: '../public/uploads'

nous avons rajouté :

bind:

\$uploadDir: '%upload_dir%'

dans le service.yaml aussi juste en dessous de

autowire: true # Automatically injects dependencies in your services.

autoconfigure: true # Automatically registers your services as commands, ...

Nous avons par la suite créé un contrôleur avec la commande :

php bin/console make:controller HomeController

Ce formulaire va permettre d'envoyer une vue contenant le formulaire Web à l'utilisateur.

Nous avons ensuite cree un controleur qui repond a la soumission du formulaire avec la commande :
php bin/console make:controller UploadController

```
$token = $request->get("token");
```

```
if (!$this->isCsrfTokenValid('upload', $token))  
{  
    $logger->info("CSRF failure");
```

```
    return new Response("Operation not allowed", Response::HTTP_BAD_REQUEST,  
        ['content-type' => 'text/plain']);  
}
```

Nous récupérons le jeton et le validons avec la méthode. Si la validation échoue, nous enregistrons l'incident et envoyons une réponse simple « Opération non autorisée » et un code de réponse. isCsrfTokenValid()Response::HTTP_BAD_REQUEST

```
$file = $request->files->get('myfile');
```

```
if (empty($file))  
{  
    return new Response("No file specified", Response::HTTP_UNPROCESSABLE_ENTITY,  
        ['content-type' => 'text/plain']);  
}
```

Nous vérifions si l'utilisateur a spécifié un fichier dans le formulaire avec la méthode. Si le champ de saisie est vide, nous renvoyons un texte brut « Aucun fichier spécifié » au client avec le code de réponse. empty()Response::HTTP_UNPROCESSABLE_ENTITY

```
$filename = $file->getClientOriginalName(); (ligne de code)
```

Nous obtenons le nom du fichier avec . getClientOriginalName()

```
$uploader->upload($uploadDir, $file, $filename); (ligne de code )
```

Nous appelons la méthode de service de téléchargement, qui déplace le fichier vers le répertoire choisi. Nous passons à la méthode le nom du répertoire, les données du fichier et le nom du fichier. upload()

```
et nous finissons par un return : return new Response("File uploaded", Response::HTTP_OK,  
    ['content-type' => 'text/plain']);
```

suite a ca nous avons crée un dossier dans src nomme Service puis dans ce dossier nous avons cree FileUploader.php

ce service va permettre de déplacer les fichiers vers le répertoire de téléchargement.

```
<h1>Troisième tâche  
</h1>
```

```

```

Les deux fichiers uploader ont été fusionnés et triés grâce à cette ligne qui nous a permis de sélectionner certaines catégories :

```
$tabName = ["Number", "GivenName", "Surname", "Birthday", "StreetAddress", 'NameSet', "EmailAddress",  
    "TelephoneNumber", "Kilograms", "CCType", "CCNumber", "CVV2", "CCEXpires", "Vehicle"];
```

```
<h1>Quatrième partie </h1>
```

```
$response = new BinaryFileResponse('../public/csv/output.csv');
```

cette ligne qui va permettre de dire ou le fichier va être téléchargé

```
$response = new BinaryFileResponse('../public/csv/output.csv');  
$response->headers->set('Content-Type', 'text/csv');  
$response->setContentDisposition(  
ResponseHeaderBag::DISPOSITION_ATTACHMENT,  
'fusion.csv'  
);  
return $response;
```

ce code va nous permettre de télécharger le nouveau fichier fusionner

<h2>Problème rencontré</h2>

Dans cette partie notre principal problème a été la fusion difficile de trouver des aides sur internet on n'a donc reçu l'aide de nos camarades de classe pour effectuer cette fusion

<h1>Conclusion</h1>

Ce projet était intéressant il nous a permis de développer certains acquis et malgré que nous n'ayons pas pu réussir jusqu'à la fin il nous a permis de nous familiariser avec le framework symfony# Vetux-line