

Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

BTS Services informatiques aux organisations Session 2020-2021

E6 – Parcours de professionnalisation
Coefficient

Description d'une situation professionnelle

Épreuve ponctuelle

~~Contrôle en cours de formation~~

~~Parcours SISR~~

Parcours SLAM

NOM et prénom du candidat : KERN Louis

N° candidat : 0319811520

Contexte de la situation professionnelle

FM Logistic cherche à améliorer son système de tickets d'incidents, ils cherchent à pouvoir passer d'un stockage en format Excel à une vérifications du formats de données via un formulaire php. Avec un affichage sous forme d'interface graphique de ses tickets

Intitulé de la situation professionnelle

Développement d'un système de traitements de données en PHP

Période de réalisation : 3^{ème} semestre 2021

Lieu : Phalsbourg

Modalité : ~~Equipe~~ Individuelle

Conditions de réalisation

Finalisation du programme de gestion des demandes liées au Master Data de notre outil de Transport
Mettre en place une restitution graphique des logs traitement jobs Talend

Productions associées

Le travail sera suivi par la Master data Manager et ses associés pour s'assurer qu'il respecte le cahier des charges, et une fois validé, il sera ensuite mis en production sur les serveurs du réseau FM Logistic.

Compétences mises en œuvre : A.1.1.1 à 3, A.1.2.2 à 5, A.1.3.1, A.1.3.3 et 4, A.1.4.1 et 2, A.2.1.1 et 2, A.2.2.1, A.2.3.2, A.3.2.2, A.4.1.1 à 8, A.1.10, A.4.2.1 et 3, A.1.2, A.5.2.1 et 3 et 4.

Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

Introduction :

La Master Data Manager cherche à optimiser et faciliter son travail au sein de l'entreprise, en effet, elle reçoit tous les jours des requêtes et diverses demandes venant des clients, mais le problème est que souvent, les informations sont incomplètes, erronées ou tout simplement au mauvais format, cet outil a donc pour but de remédier à ce problème en normalisant le format des données à envoyer et à permettre au MDM de pouvoir gérer plus facilement les demandes qui lui sont adressées.

Mon objectif est de finir un projet de formulation de demandes envers un Master Data Manager (MDM) et de d'en améliorer l'ergonomie ainsi que d'ajouter des fonctionnalités supplémentaires.

Le but étant de faire régulièrement des points avec la Master Data Manager afin de s'assurer du bon format des demandes, que les données demandées soient claires et ordonnées afin de faciliter le travail du Master Data Manager, en effet ce site ne sert qu'à formuler et à afficher de manière lisible des demandes, le traitement est effectué ensuite par la Master Data Manager.

Un autre défi du projet est de devoir gérer les accès et possibilités d'action du côté client mais aussi du côté MDM, par exemples, un client peut formuler une demande et performer quelques actions dessus, en revanche, le MDM ne peut que les consulter, mais peut performer plus d'actions dessus.

Beaucoup de fonctionnalités du site sont affectées par rapport à quel type d'utilisateur est connecté (client ou MDM).

Condition de réalisation :

Environnement de développement

Logiciels : Visual Studio Code, MySQL Workbench

Langages : PHP, HTML, CSS, SQL

Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

Formulaire de demandes :

Le travail le plus important de ce projet est la présentation, le contenu, le format et l'affichage des différentes demandes envers la Master Data Manager, pour illustrer le processus, nous allons nous connecter en tant que client :

Serveur test !

Master Data Management

Interface Utilisateur

Tableau de bord

Nouvelle Demande

Mes Demandes

Demandes agence

Aide

FMLOGISTIC

1 Notifications

louis kern

Mes Demandes

En Attente

0

Voir

En Cours

0

Voir

Actions Requises

0

Voir

Traitées

0

Voir

Demandes de mon Agence

En Attente

0

Voir

En Cours

1

Voir

Traitées

0

Voir

Nous sommes sur la page tableau de bords, contenant une vue globale des différentes demandes déjà faite par l'utilisateur et son agence et leur état actuel.

En cliquant soit sur « mes demandes », soit « Demande agence » l'utilisateur arrive sur cette fenêtre :

| # | Sujet | Type | Auteur | Site | Etat | Créée le |
|------|--|---------------------------------------|------------|------|----------|------------------|
| #160 | Création de Tarif Transport Référencé : Hubert corporate | Création de Tarif Transport Référencé | louis kern | LPO | Nouveau | 04/02/2021 11:24 |
| #159 | Création de Tracteur FM eee | Création de Tracteur | louis kern | LPO | Annulée | 01/02/2021 11:27 |
| #158 | Création de Tarif Client : e | Création de Tarif Client | louis kern | LPO | Annulée | 01/02/2021 09:52 |
| #157 | Transfer de Tracteur e | Transfert de Tracteur | Louis Kern | LPO | Annulée | 29/01/2021 16:09 |
| #156 | Transfer de Tracteur ee | Transfert de Tracteur | Louis Kern | LPO | En Cours | 29/01/2021 15:31 |
| #155 | Transfer de Tracteur eee | Transfert de | Louis Kern | LPO | Annulée | 29/01/2021 15:31 |

Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

Ceci affiche le listing de toutes les demandes avec leur numéro, leur sujet, type, auteur, etc...

Quand l'utilisateur veut formuler une nouvelle demande, il clique sur « Nouvelle demande » et se retrouve sur cette page :

The screenshot shows the 'Nouvelle Demande' (New Request) page. On the left is a dark sidebar with the title 'Serveur test ! Master Data Management Interface Utilisateur' and a menu with items: 'Tableau de bord', 'Nouvelle Demande' (highlighted), 'Mes Demandes', 'Demandes agence', and 'Aide'. The main content area has the heading 'Choisissez une catégorie:' and four buttons: 'Comptes' (Modification de comptes client ou transporteur), 'Tarifs' (Création ou modification de tarifs client ou transporteur), 'Moyens Propres' (Création ou modification de chauffeurs, tracteurs ou remorques), and 'Lieux' (Création ou modification de lieux).

Après avoir sélectionné une catégorie, il se retrouve ensuite sur cette page :

The screenshot shows the 'Nouvelle Demande - Tarifs' (New Request - Rates) page. The sidebar is identical to the previous screenshot. The main content area has the heading 'Nouvelle Demande - Tarifs' and the question 'Que souhaitez vous faire ?'. Below this are seven buttons: 'Créer un Tarif Client', 'Modifier un Tarif Client', 'Créer un Tarif Transp. Référencé', 'Modifier un Tarif Transp. Référencé', 'Créer un Tarif Sous Traitant', 'Modifier un Tarif Sous Traitant', and 'Reconduire une grille tarifaire'.

Depuis là il peut choisir quel formulaire de demande l'utilisateur veut remplir. Il est important de noter que les modifications de demandes créées de nouvelles demandes au même titre que les demandes de créations, la différence étant que le sujet est différent de leur homologue « création », et que seule une information permettant d'identifier quel élément sera affecté par le MDM et les informations à modifier sont demandés (le formulaire peut être envoyé incomplet).

Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

Voici un exemple de formulaire pour une création de Tarif Client référencé :

The screenshot shows a web application interface for 'Master Data Management'. On the left is a dark sidebar with a red header 'Serveur test !' and the title 'Master Data Management' with the subtitle 'Interface Utilisateur'. The sidebar contains a menu with items: 'Tableau de bord', 'Nouvelle Demande', 'Mes Demandes', 'Demandes agence', and 'Aide'. The main content area is a light gray form titled 'Interface Utilisateur'. The form contains several sections: 1. 'Code Activité' (dropdown with 'Distribution') and 'Indice Gazoil' (dropdown with 'Oui'). 2. 'Niveau Service' (text input with 'Maximum'). 3. 'Point de départ' (text input with 'Sarrebouurg'), 'Point de livraison' (text input with 'Strasbourg'), and 'Groupage' (dropdown with 'Zone (FR)'). 4. 'Type de tarif' (dropdown with 'Forfaitaire') and 'Date de mise en place' (calendar icon with '04/02/2021'). 5. 'Grille tarifaire (obligatoire)' section with a 'Choisir un fichier' button and a file named 'test_grille.xlsx'. 6. Two links: 'Exemple de grille tarifaire séquentielle pour confrère' and 'Exemple de grille tarifaire forfaitaire pour confrère'. At the bottom is a large blue 'Valider' button.

Une fois le formulaire remplis, lors de l'envoi on récupère le nom du formulaire (ex creaTC) et on construit le sujet de la demande avant d'envoyer en base de données le formulaire avec ses données.

```
<form action="{{route('submit')}}" method="POST" class="needs-validation card pl-3 pr-3 pt-2 pb-2" style="@csrf
<input name="auteur" value="{{ Auth::user()->id }}" hidden>
<input name="type" value="t_creaTC" hidden>
<div class="form-row">
  <div class="col-md-6 mb-3">
    <label for="f_chez">Chez</label>
    <input type="text" class="form-control form-control-sm" name="f_chez" placeholder="Chez" required/>
  </div>
</div>
```

Nom du formulaire dans un input nommé « type » caché, le but est de pouvoir récupérer le nom du « type » de formulaire (dans cet exemple « t_creaTC ») avec t_ qui indique que c'est un formulaire de la catégorie « tarif », « crea » qui indique que c'est un formulaire de création et, enfin, « TC » qui est les initiales de « Tarif Client », Avec cette nomenclature, nous pouvons facilement identifier de quel formulaire est utilisé et pouvoir construire le sujet grâce à cette information.

```
<div class="form-row">
  <div class="Immatriculation col-md-4 mb-3" style="display: none;">
    <label for="f_immatric">Immatriculation de la Remorque</label>
    <input type="text" class="form-control form-control-sm" name="f_immatric" placeholder="Immatriculati
  </div>
  <div class="tractST col-md-4 mb-3" style="display: none;">
    <label for="f_searchname">Searchname du sous-traitant</label>
    <input type="text" class="form-control form-control-sm" name="f_searchname" placeholder="Searchname
  </div>
</div>
```

Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

On récupère ensuite la valeur des inputs (cette image montre un code venant d'un autre formulaire) avec chacun leur tag propre.

```
$req->c_searchname = $request->input('f_searchname');
$req->c_typeclient = $request->input('f_typeclient');
$req->c_chez = $request->input('f_chez');
$req->c_pour = $request->input('f_pour');
$req->c_codegestion = $request->input('f_codegestion');

$req->c_adresse = $request->input('f_adresse');
$req->c_codepostal = $request->input('f_codepostal');
$req->c_ville = $request->input('f_ville');
$req->c_pays = $request->input('f_pays');

$req->c_tva = $request->input('f_tva');
$req->c_telephone = $request->input('f_tel');
$req->c_fax = $request->input('f_fax');
$req->c_mail = $request->input('f_mail');

$req->t_typemodif = $request->input('f_typemodif');
$req->t_codeactivite = $request->input('f_codeactivite');
$req->t_niveauservice = $request->input('f_niveauservice');
$req->t_indicegazole = $request->input('f_indicegazole');

$req->t_depart = $request->input('f_depart');
$req->t_livraison = $request->input('f_livraison');
$req->t_groupage = $request->input('f_groupage');

$req->t_type = $request->input('f_type');
$req->t_miseenplace = $request->input('f_miseenplace');
```

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Request extends Model
{
    protected $fillable = [
        'etat',
        'auteur',
        'sujet',
        'type',
        'commentaire',
        'c_searchname',
        'c_typeclient',
        'c_chez',
        'c_pour',
        'c_codegestion',
        'c_adresse',
        'c_codepostal',
        'c_ville',
        'c_pays',
        'c_tva',
        'c_telephone',
        'c_fax',
        'c_mail',
        't_numtarif',
        't_codeactivite',
        't_indicegazole',
    ];
}
```

Dans un fichier RequestController (à gauche), qui s'occupe de formater les données avant l'envoi dans la base de données, on récupère toutes données possibles venant d'un formulaire, le programme ne fait pas de distinction, les données manquantes seront simplement *null*. On envoie les variables dans le fillable du RequestBuilder. Puis ils les postent en base de données.

Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

Mais avant ça on doit encore construire le sujet de la demande :

```
public static function typeName($type)
{
    switch ($type)
    {
        case 'l_creal':
            return 'Création de Lieu';
        case 'l_modifL':
            return 'Modification de Lieu';

        case 'c_modifCT':
            return 'Modification de Client Transporteur';
        case 'c_modifRC':
            return 'Modification ou validation de relation client';
        case 'c_modifT':
            return 'Modification de Transporteur';

        case 't_creaTC':
            return 'Création de Tarif Client';
        case 't_creaTST':
            return 'Création de Tarif Sous-Traitant';
        case 't_creaTTR':
            return 'Création de Tarif Transport Référencé';
        case 't_modifTC':
            return 'Modification de Tarif Client';
        case 't_modifTST':
            return 'Modification de Sous-Traitant';
    }
}
```

La fonction typeName permet de renvoyer des messages prédéfinis en fonction du nom du type de formulaire passé en paramètre, on appelle cette fonction dans save pour pouvoir construire le sujet de la demande.

En fonction du type de formulaire on construit dans la fonction save le sujet par rapport au type de formulaire utilisé et on récupère la valeur qu'a retourné typeName pour le formulaire passé en paramètre ainsi que les valeurs désirées des formulaires.

```
protected function save(Request $request)
{
    $req = new Req;

    $req->etat = 'En Attente';
    $req->auteur = $request->input('auteur');
    $req->type = $request->input('type');
    switch ($request->input('type'))
    {
        case "c_modifT":
            $req->sujet = $this->typeName($request->input('type')) . " : " . $request->input('f_searchname');
            break;
        case "c_modifCT":
            $req->sujet = $this->typeName($request->input('type')) . " : " . $request->input('f_searchname');
            break;
        case "c_modifRC":
            if ($request->input('f_typedmodif') == "Modification")
            {
                $req->sujet = "Modification de relation client" . " : " . $request->input('f_searchname');
            }
            else
            {
                $req->sujet = "Validation de relation client" . " : " . $request->input('f_searchname');
            }
            break;
    }
}
```

Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

Ensuite on ajoute les données spécifiques à la demande comme le nom de l'élément affecté puis on l'ajoute dans fillable. Si une grille tarifaire est présente on l'enregistre dans les fichiers du site, dans le cas d'un renvoi de grille via l'historique on lui attribue un nom spécial et un Id propre pour éviter les doublons de grilles sur la même demande et pouvoir tracer quelle grille correspond à quelle action :

```
protected function downloadGrille($id)
{
    $req = Req::findOrFail($id);

    $path = 'app/' . $req->t_grille;
    return response()
        ->download(storage_path($path));
}

protected function downloadHistfile($id, $hist_id)
{
    $hist = Hist::where('req_id', $id)->where('hist_id', $hist_id)->first();
    $name = preg_replace('(.*/(\d+_?\d+)?)', '', $hist->fichier);
    $path = 'app/' . $hist->fichier;
    return response()
        ->download(storage_path($path) , $name);
}

protected function downloadExempleGrille($name_file)
{
    $path = 'app/public/exemples_grilles/' . $name_file . '.xlsx';
    echo $path;
    return response()->download(storage_path($path));
}
```

Puis on récupère le nom du fichier et on y ajoute le numéro de la demande associé pour éviter les doublons si jamais une grille correspondant à une autre demande ayant le même nom serait amené à être uploadé sur le site :

```
case "t_creaTTR":
    $req->sujet = $this->typeName($request->input('type')) . " : " . $request->input('f_searchname');
    if ($request->hasFile('f_grille'))
    {
        $fileName = Req::max('id') + 1 . "_" . $request->file('f_grille')
            ->getClientOriginalName();
        $req->t_grille = $request->file('f_grille')
            ->storePubliclyAs('public/grilles', $fileName);
    }
}
```


Fiche PP 3 : Formulation de demandes en PHP

Kern Louis

Une fois la demande créée on l'affiche dans une nouvelle fenêtre :

Votre ticket a été créé avec succès !

Demande #160

🕒 En Attente

Création de Tarif Transport Référencé : Hubert corporate

⚙️ Actions

Informations

Identifiant: #160

Type: Création de Tarif Transport Référencé

Demandeur: louis kern

E-Mail: louis kern@mail.com

Agence: LPO

Dates

Date de Création: 04/02/2021 11:24

Dernière Modification: 04/02/2021 11:24

Données

Général

Searchname: Hubert corporate

Indice de gasoil: Oui

Niveau de service: Maximum

Code activité: Distribution

Date

Date de mise en place: 04-02-2021

📄 télécharger la grille tarifaire

Adresse

Point de départ: Sarrebourg

Point de livraison: Strasbourg

Type de groupage: Zone (FR)

Type de tarif: Forfaitaire

Historique

👤 louis kern

04/02/2021 11:24

Création de la demande

Les informations sont rangées en fonction du type de données, avec les données propres à l'utilisateur ayant fait la demande étant affichées à droite et les données relatives à la demande en elle-même sont dans leur catégorie à droite.

On inclut aussi un historique permettant de tracer les différentes actions et/ou commentaires menés sur la demande. Il sert aussi en quelque sorte d'espace de chat.

Ceci conclut le processus de création de demande.

Conclusion :

Ce PP, m'a permis d'approfondir mes connaissances en PHP, et surtout de s'initier à l'utilisation d'un Framework, spécialement du Framework Laravel. Mais aussi d'avoir un projet concret avec un suivi du « client » de manière régulière en planifiant des réunions, poser des questions, répondre de manière pertinente aux demandes.