Compte Rendu

05 2017

Compte rendu pour des BTS SIO 2ème année pour le PPE : Galaxy Swiss Bourdin (GSB) Application métier de gestion de frais.

Hugo Chouraki

Guillyan Chapput

David Molinari

Table des matières

[Mission 1 : Installation et test de l’application 1](#_Toc482421453)

[Mission 2 : Développement de la partie comptable 5](#_Toc482421454)

[Mission 3 : Modifications de l’application 7](#_Toc482421455)

[Mission 4 : Ajout de la génération d'un PDF 8](#_Toc482421456)

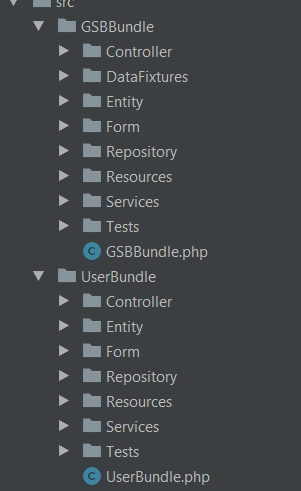
[Mission 5 : Application mobile 10](#_Toc482421457)

# Mission 1 : Installation et test de l’application

Disposer d’un serveur web avec MySQL et PHP  Avoir traité le cours SI3 "Exploitation des données"  Avoir traité le cours SI5 "Support des services et des serveurs" (facultatif)  Avoir traité le cours SI6 "Développement d’applications »

## Tâche 1 : Installation de l’application existante

L’application a été entièrement refaite migrée vers le framework php Symfony.

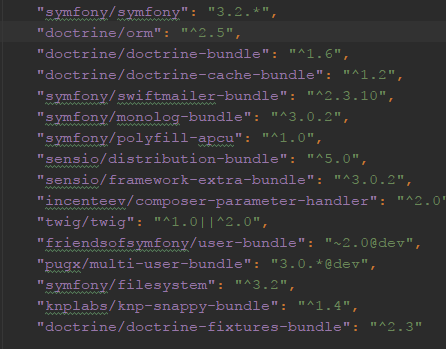


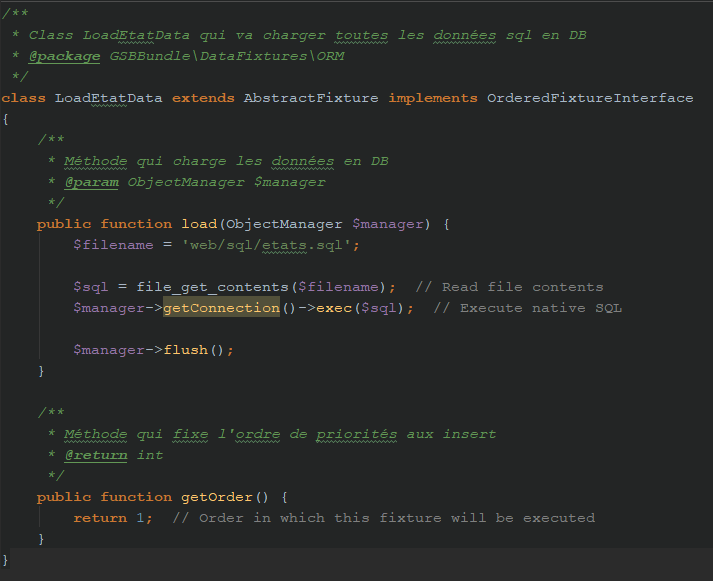
Nous avons créé deux bundles, GSBBundle qui sert pour côté back-end et métier de l’application et UserBundle pour la gestion des utilisateurs.

Nous utilisons également FosUserBundle pour la gestion des utilisateurs, DoctrineFixtureBundle pour la mise à jours automatique en base de données.

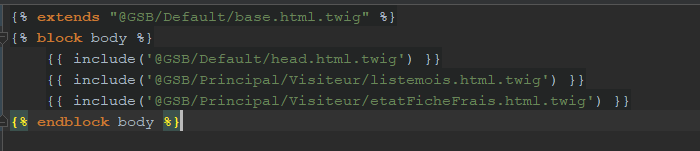
Twig comme moteur de template, KnpSnappyBundle pour gérer le Pdf.

Liste des composants utilisés :

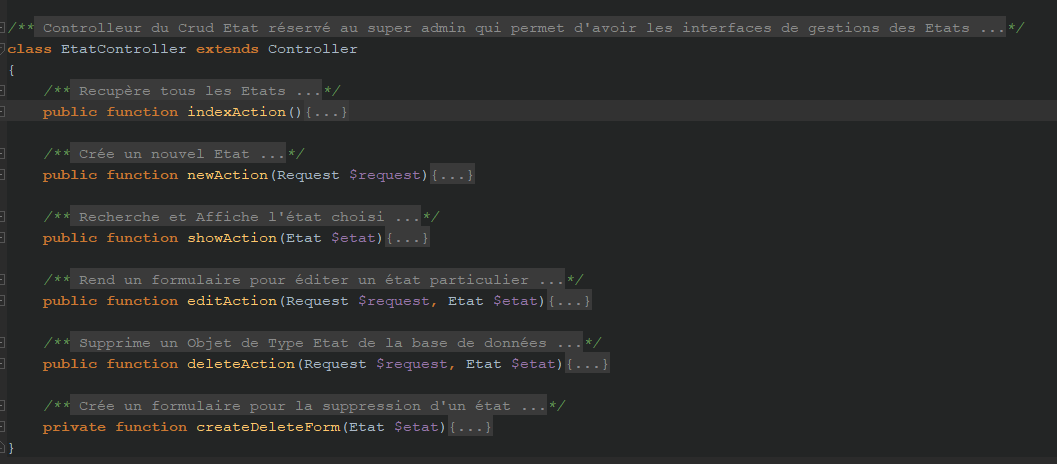
d

Plus besoin d’importer les script SQL à la main et devoir configurer le php.ini pour augmenter la taille maximal des upload en DB sous mysql, on utilise DoctrineFixtureBundle :

Nos vues sont séparées, épurées et plus simples à maintenir

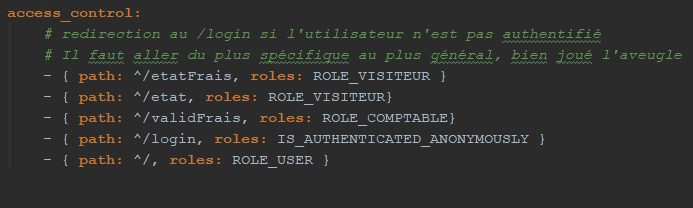


Nous utilisons également des CRUD pour gérer toutes les Entity de notre projet



Nos utilisateurs ont tous des Rôles : Visiteur / Comptable / Admin

Les accès sont tous gérés par le fichier security.yml



## Tâche 2 : Tests de l’application

Toutes les vues ont été refaites, sous le moteur de templates Twig offrant une meilleure gestion et une meilleure maintenabilité du code.

## Tâche 3 : Production de la documentation

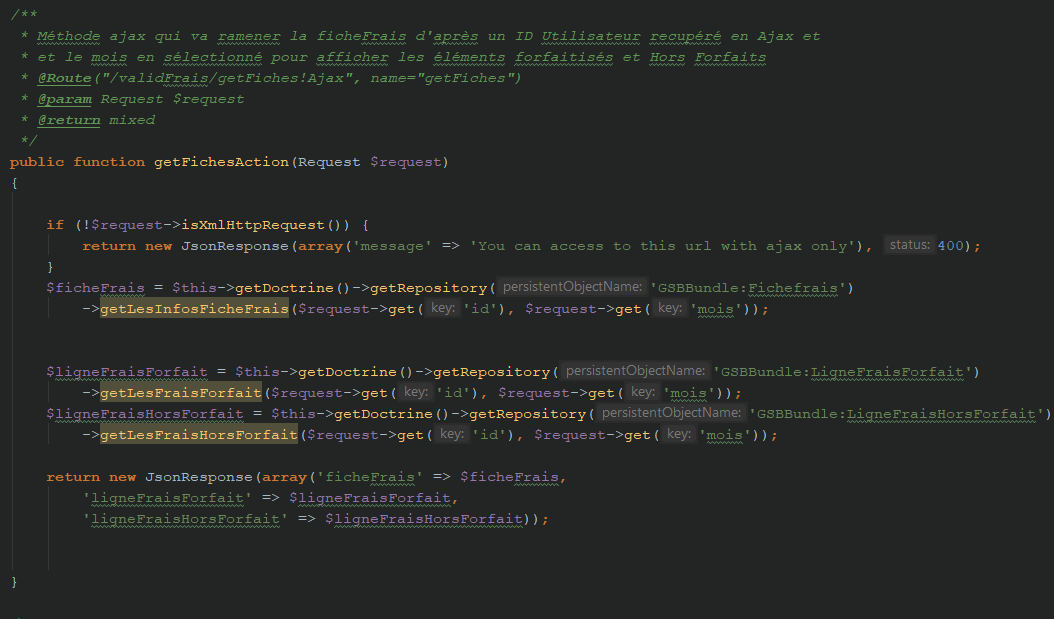
La documentation se fait avec PhpDocumentor

# Mission 2 : Développement de la partie comptable

## Tâche 1 : Validation d’une fiche de Frais

La partie Validation d’une fiche de Frais se fait en Ajax pour récupérer les données.

Exemple d’une action pour récupérer les données d’un visiteur choisi :

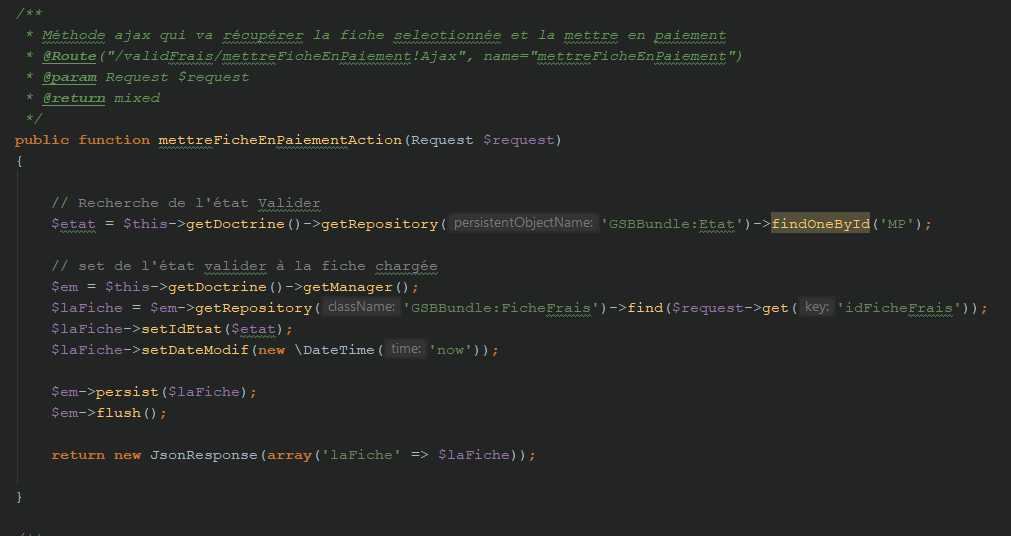


.

## Tâche 2 : Suivre une Fiche de Frais

Le suivi d’une fiche de Frais se fait également en Ajax

Exemple pour mettre une fiche en Mise en Paiement :



**Tâche 3 : Production de la documentation**

La documentation se fait avec PhpDocumentor

# Mission 3 : Modifications de l’application

## Tâche 1 : Gestion du refus de certains frais hors forfait

Appel en Ajax qui se fait au click sur le lien supprimer un frais.

## Tâche 2 : Sécurisation des mots de passe stockés

Une simple ligne en DB aurait suffit à hash tous les mot de passes, nous avons préféré la gestion propre avec FOSUSERBUNDLE .

## Tâche 3 : Gestion plus fine de l'indemnisation kilométrique

/

# Mission 4 : Ajout de la génération d'un PDF

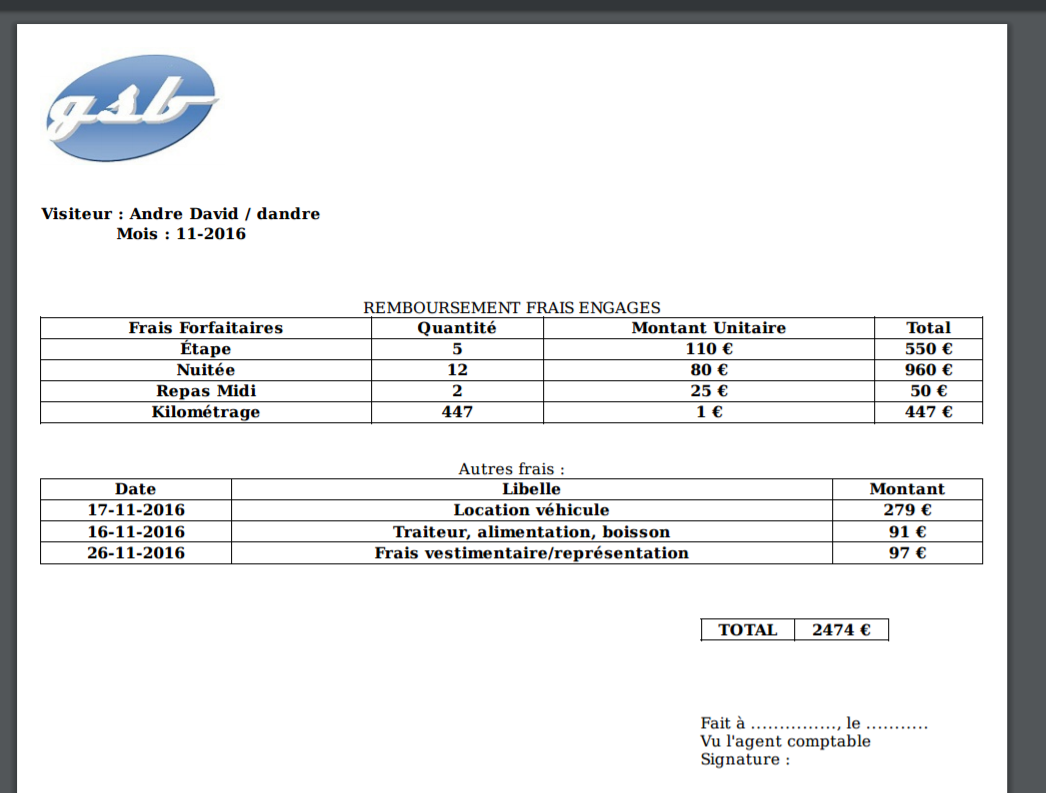
## Tâche 1 : Génération d'un état de frais au format PDF

La génération se fait au click sur lien pour générer le pdf en question.

Nous utilons wkhtmltopdf comme binaire pour générer le pdf :

Il s’enrengistre dans un dossier spécifique avec téléchargement qui se lancera :

Aperçu du PDF :



## Tâche 2 : Davantage d'écologie dans l'application

Un simple test suffit à ne pas générer le PDF une deuxième fois, les pdf sont formatés avec l’ID utilisateur et le mois de la fiche

Ajout d’une condition « fs » qui vérifie si le fichier existe déjà dans le PATH d’enregistrement donnée :

if (!$fs->exists('PDFs/' . $iduser . '-' . $mois . '.pdf')

# Mission 5 : Application mobile

## Tâche 1 : Configuration

Création de l’AVD

## Tâche 2 : Configuration Interdiction de saisie directe des quantités

Interdiction de rendre cliquable la zone de texte, ainsi seul les « + » et « -» sont utilisables

## Tâche 3 : Enregistrement des autres catégories de frais forfaitisés

Copie de l’existant pour créer les autres activités et modification des noms et images (via leur nom)

## Tâche 4 : Suppression de frais hors forfait

Fonction trouvée sur internet utilisant le holder permettant de supprimer le frais sur la ligne cliquée :

holder.**imgSuppr**.setTag(index);  
  
  
  
holder.**imgSuppr**.setOnClickListener(**new** View.OnClickListener() {  
  
 @Override  
 **public void** onClick(View v) {  
 String pos = v.getTag().toString();  
 **int** position = Integer.*parseInt*(pos);  
 **lesFrais**.remove(position);  
 Serializer.*serialize*(Global.***filename***, Global.*listFraisMois*,**context**) ;  
 notifyDataSetChanged();