[Choisir la date]

Portail Wb M2L

Document technique

Dozu

Table des matières

L'application Portail Web : Vue globale

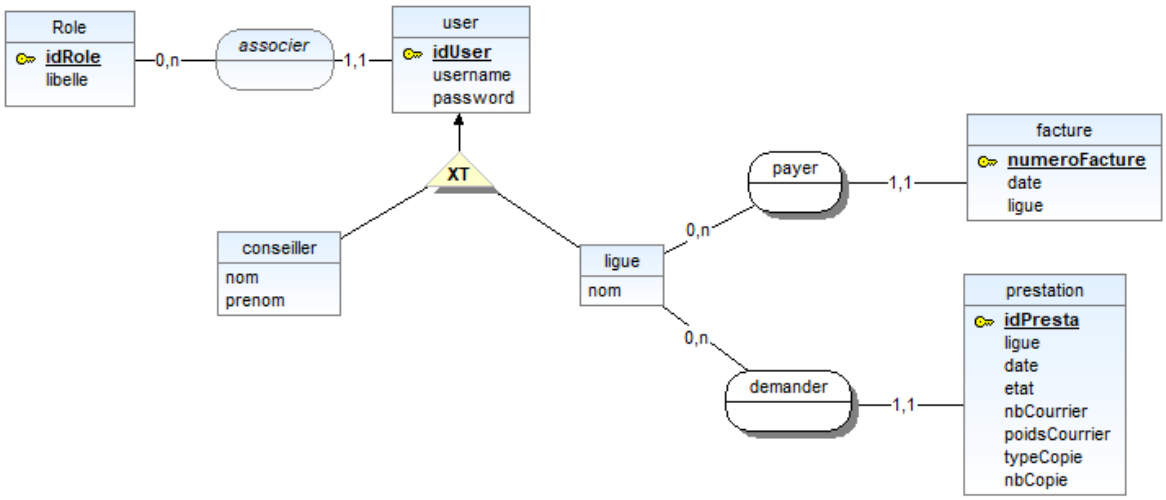
Présentation générale

L'application web de la M2L est une application qui permet de gérer les prestations (reproduction et affranchissement) demandés par les ligues. Elle est développée en NodeJs selon le pattern MVC (Modèle-Vue-Contrôleur) et est connectée à une base de données stockée sous PostgreSQL.

La base de données

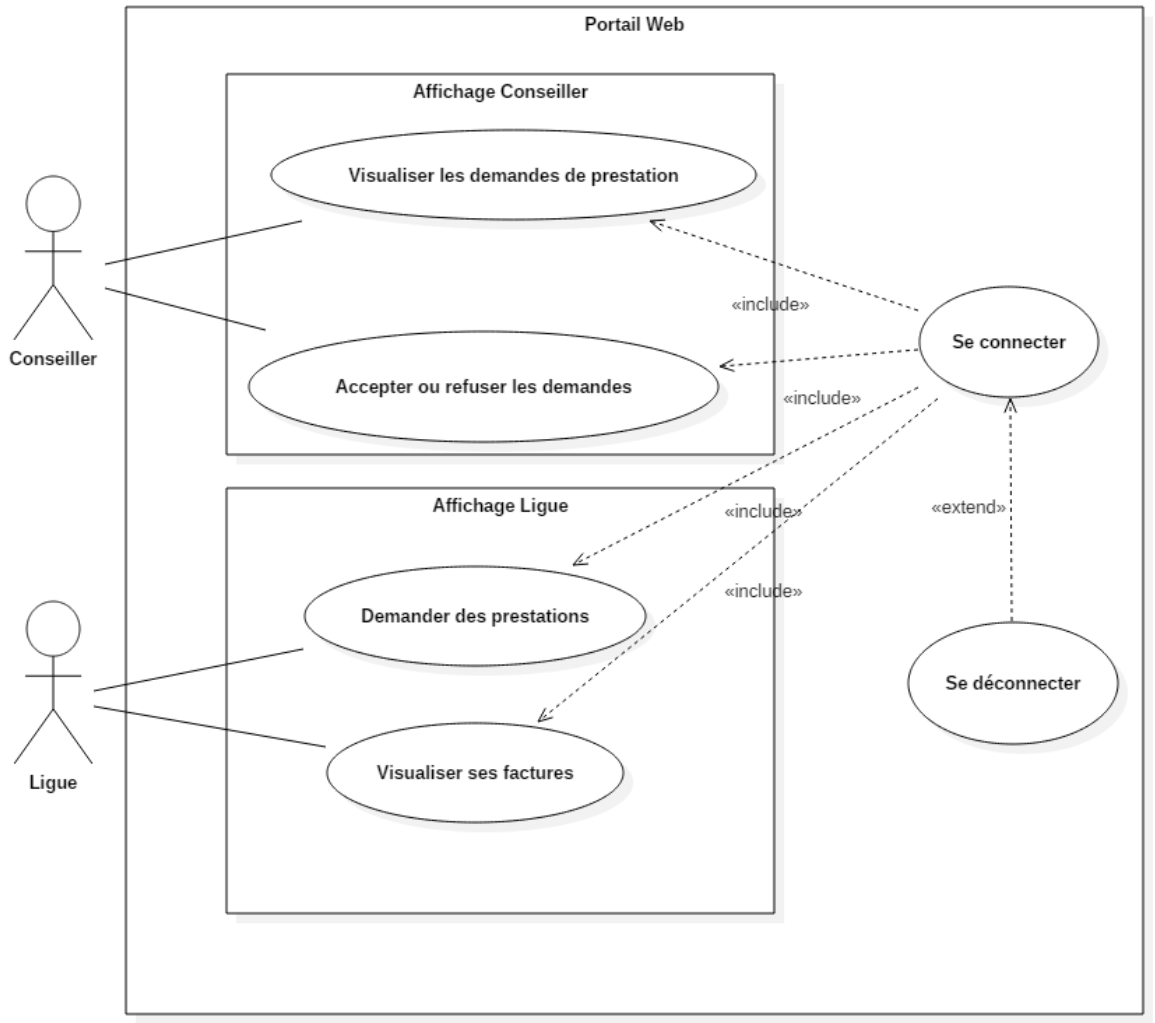
La base de données nommées M2L est stockée sur une instance PostgreSQL du serveur 192.168.222.70 sur le port 5432.

Une partie de la structure de M2L est illustrée à travers le schéma ci-dessous :



Les fonctionnalités

Les fonctionnalités de l'application sont illustrées à travers le schéma ci-dessous :

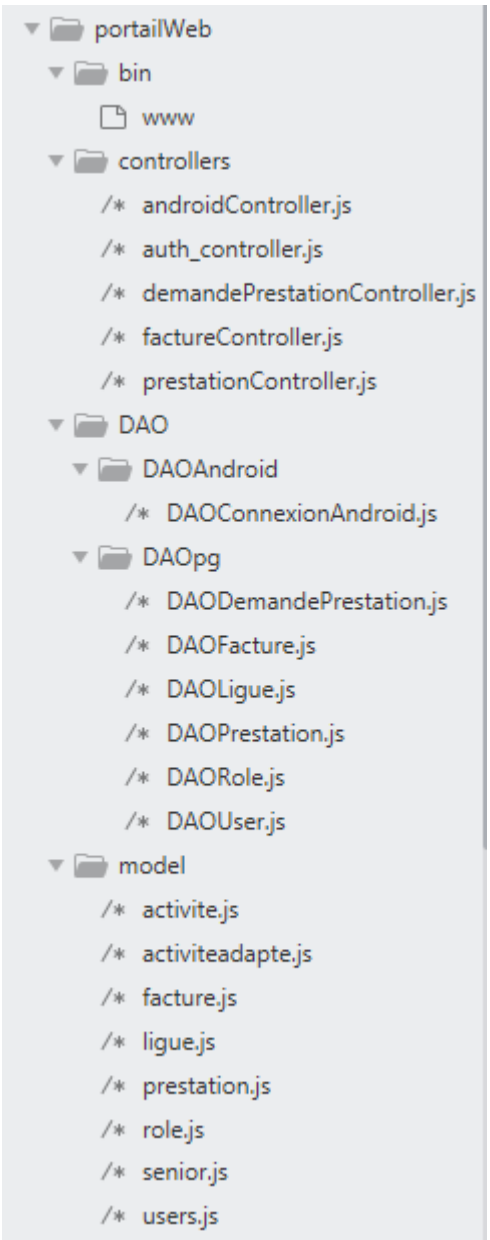


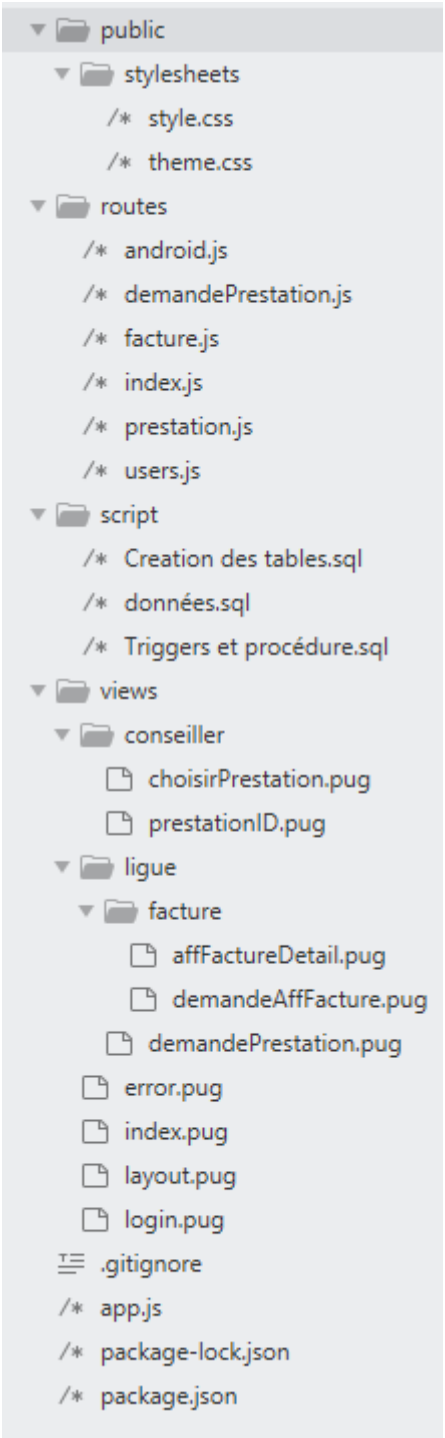
Pour l'instant, toute les fonctionnalités sont totalement fonctionnelles. Seul une modification de la visualisation des factures semble nécessaire.

L'application portail Web : Vue Technique

L'application NodeJs

Les codes sources de l'application sont trouvable à l'adresse : https://github.com/StephanSo/M2LNodeJsTEAM/commits/master ou sur l'archive M2LNodeJsPortailWeb. Ils sont organisés sous l'arborescence suivante :





- App.js retient les informations relatives au routes et permets le lancement de l'application.

- Package.json recense toute les librairies qui est nécessaire au bon fonctionnement du projet

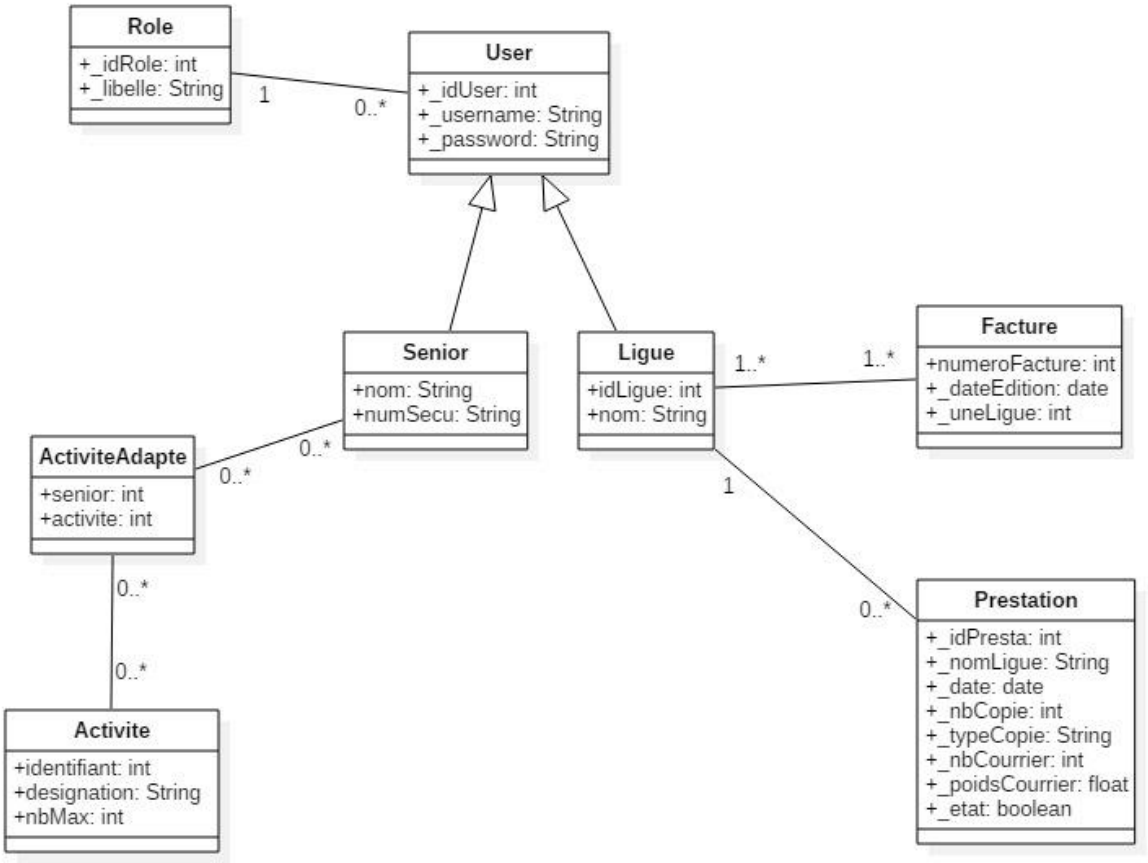
- bin/www est le fichier qui sert à exécuter le serveur web

- *Controllers, DAO, Model, public, routes, views* sont les dossiers du code sources de l'application

- *Script* recense les scripts de création et modification de la base de données

Le package Model

Le package *model* contient les classes métier de l'application



Le package DAO

Le package DAO permet la connexion avec la base de données afin de liés les modèles avec les entités de la base de données.
Nous avons 6 DAO utilisés sur cette application :

DAODemandePrestation {

ajouterDemandePrestation(idLigue, dateDemande, nbCopie, typeCopie, nbAffra, pdsAffr){}
Cette méthode permet l'ajout d'une prestation dans la base de données.

}

DAOFacture {

afficherFacture (ligue, annee, mois, cb){}
Cette méthode retourne les factures grâce à l'id de la ligue, l'année et le mois de la facture

}

DAOLigue {

voirToutesLesLigues (displaycb){}
Cette méthode retourne toutes les ligues présente dans la base de données

ligueByUser (username, cb){}
Cette méthode retourne le nom de la ligue en fonction de l'utilisateur connecté

ligueById (id, cb){}
Cette méthode retourne le nom de la ligue grâce à son id

}

DAOPrestation{

getAllPrestation (cb){}
Cette méthode retourne toutes les prestations présente dans la base de données

getPrestationById (id,cb){}
Cette méthode retourne une prestation en fonction de son id

UpdateTruePrestationById(id){}
Cette méthode permet de passer l'état d'une prestation à True en fonction de son Id

UpdateFalsePrestationById (id){}
Cette méthode permet de passer l'état d'une prestaton à False en fonction de son ID

}

DAORole {

getRoleByUsername (usernameP, cb){}
Cette méthode permet de récupérer le rôle en fonction de l'username de la personne connecté

}

DAOUser {

getUserAndroid(cb){}
Pour cette méthode voir la documentation technique de l'application Android

getUser(cb){}
Cette méthode retourne la liste des utilisateurs

loginUser (usernameP, passwordP, cb){}
Cette méthode permet de vérifier la connexion à l'application elle retourne « ok » si le username et le password sont correcte , elle retourne « nonOk » sinon

}

Le package View

Toutes les vues sont basées sur le **layout.pug** qui permet de définir le style principal de la page. Les fichier CSS sont dans le dossier **public/stylesheets**

Faire shéma des vus ici

Le package Controller

Le package controller contient les différentes classes qui permettent le traitement de certaine donnée du DAO pour ensuite les afficher grâce au View dans l'application.

La navigation dans l'application

Pour naviguer dans l'application nous avons divers URLS que voici :

GET / Affiche l'accueil de l'application

GET /users/login Permet d'afficher la page de connexion

POST /users/login Permet d'envoyer la demande de connexion

GET /users/logout Permet de se déconnecter

GET /demandePrestation Affiche le formulaire de demande prestation

POST /demandePrestation Envoie le formulaire

GET /prestation Affiche la liste des prestations

GET /prestation/:id Affiche les détails de la prestation :id

POST /prestation/:id Permet de valider la prestation :id

GET /facture/demandeAfficheFact Affiche la page pour demander une facture

POST /facture/demandeAfficheFact Affiche la facture demandée