

Projet d'intégration



Franck Baldassini

Licence professionnelle Métiers du Numérique

SOMMAIRE

- I) Objectif**
- II) Base de données**
- III) Développement**
 - a) Android**
 - b) API Laravel**
 - c) VueJS**
- IV) Manuel de déploiement**
 - a) Android**
 - b) API Laravel**
 - c) VueJS**

I) Objectif

Nous travaillons pour une entreprise qui s'occupe de la maintenance de chaudières chez des particuliers. L'entreprise souhaite posséder une application qui permette d'enregistrer les rapports d'intervention des techniciens sur leurs portables et qu'ils puissent en fin de journée envoyer tous ces rapports sur le serveur de l'entreprise. Le patron veut pouvoir ainsi accéder à tous les rapports sur son ordinateur afin de suivre le travail de ses techniciens.

Le patron doit pouvoir consulter le nom et le prénom du technicien et du client, l'adresse du client, la marque, le modèle, le numéro de série et la date de mise en service de la chaudière. Enfin la date d'intervention du technicien et le temps qu'il a passé.

Pour cela, nous allons créer une application mobile sur android qui permettra d'envoyer à une API Laravel les rapports d'intervention des techniciens. Les rapports seront consultables via l'API grâce à une application VueJS.

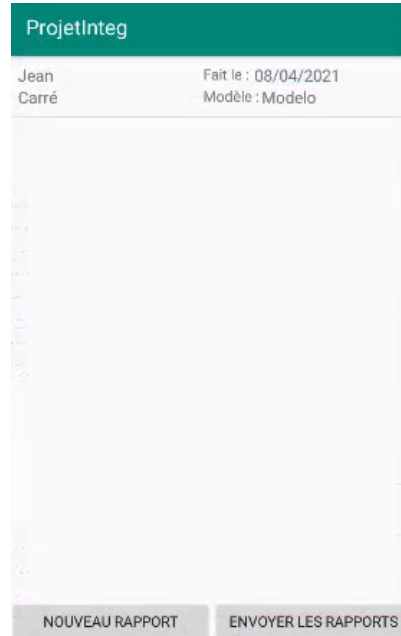
II) Base de données

Rapport
id
nomTechnicien : string
prenomTechnicien : string
nomClient : string
prenomClient : string
adresse : string
marqueChaudiere : string
modeleChaudiere : string
numSerie : integer
dateMiseService : date
dateIntervention : date
description : string
temps : time

La base de données est constituée d'une seule table que ce soit pour l'API ou l'application Android. Cependant sur android le type des attributs utilise string pour les type date et time dû à des difficultés pour stocker les valeurs dans le bon format.

III) Développement

a) Android



L'application se présente en 2 pages (Activity). La première présente la liste des rapports qui ont été créés avec 2 boutons en bas de page. Le bouton de gauche renvoie au formulaire de création d'un rapport d'intervention. Celui de droite envoie tous les rapports stockés en base de données à l'API puis vide la base de données.

ProjetInteg

Rapport d'intervention

Infos technicien

Nom technicien
Michel

Prénom technicien
Legrand

Infos client

Nom client
Jean

Prénom client
Carré

Adresse
23 rue des Pins

Infos chaudière

Marque chaudière
Lamarque

Modèle chaudière
Modelo

La deuxième page présente le formulaire avec tous les champs à remplir et à stocker en base de données. Chacun des champs prévoit de recevoir un type de données précis. Par exemple les champs pour les dates amène l'utilisateur à ne pouvoir écrire qu'une date et pas un texte cependant cela ne permet pas de d'écrire uniquement les dates au format dd/mm/yyyy.

Au moment de l'ajout, le formulaire vérifie que tous les champs ont été bien remplis.

Lors de l'envoi des dates à l'API le format passe de dd/mm/yyyy au format yyyy-mm-dd sans quoi l'API ne pourra pas stocker le rapport.

b) API Laravel

L'API Laravel utilise la base de données présentée ultérieurement. Elle reçoit et stocke les rapports récupérés par l'application android. Tous les rapports sont ensuite consultables au format JSON en écrivant l'URL dans le navigateur.

```
▼ data:
  ▼ 0:
    id: 25
    nomTechnicien: "Michel"
    prenomTechnicien: "Legrand"
    nomClient: "Jean"
    prenomClient: "Galoup"
    adresse: "23 rue des pins"
    marqueChaudiere: "La marque"
    modeleChaudiere: "Le modele"
    dateMiseService: "2022-06-12"
    numSerie: 56648
    dateIntervention: "2033-05-23"
    description: "Ceci est une description"
    temps: "01:50:00"
    created_at: "2021-04-09T07:48:25.000000Z"
    updated_at: "2021-04-09T07:48:25.000000Z"
  ▼ 1:
    id: 26
    nomTechnicien: "Michel"
    prenomTechnicien: "Legrand"
    nomClient: "Karim"
    prenomClient: "Machin"
    adresse: "45 rue des pins"
    marqueChaudiere: "Une autre marque"
    modeleChaudiere: "Un autre modèle"
    dateMiseService: "2021-06-30"
    numSerie: 55666
    dateIntervention: "2019-12-21"
    description: "Un autre description"
    temps: "00:50:00"
    created_at: "2021-04-09T07:48:25.000000Z"
    updated_at: "2021-04-09T07:48:25.000000Z"
```

Pour tester si l'API est capable de recevoir des données, j'ai utilisé l'extension Rester sur le navigateur Firefox avant de chercher à poster des données depuis android

pour vérifier que si l'envoi des données ne marche pas, alors cela venait de l'application android et non de l'API.

c) VueJS

La consultation des rapports se fait avec VueJS. En arrivant sur l'accueil, on a la liste de toutes les interventions que l'API stocke.

Accueil

Liste des interventions

Rechercher un nom de technicien

jj/mm/aaaa

Rechercher

Intervention de Michel Legrand

Client : Jean Galoup

Intervenu le : 23/05/2033

Intervention de Michel Legrand

Client : Karim Machin

Intervenu le : 21/12/2019

Intervention de Michel Legrand

Client : Jean Carré

Intervenu le : 08/04/2021

Les rapports sont présentés dans des cards Bootstrap avec le nom et prénom du technicien et du client ainsi que la date de l'intervention.

L'utilisateur a la possibilité de chercher des rapports d'intervention grâce au nom du technicien et/ou avec la date de l'intervention. Ces recherches sont faites grâce à des fonctions de filtrages proposées par VueJS au moment de l'appel à l'API.

Lors du survol d'une card, la card s'anime et propose instinctivement de cliquer dessus pour voir les détails du rapport choisi. Cette animation a été faite avec du CSS.

Lors du clic sur un rapport d'intervention, on est redirigé sur une page qui présente les détails du rapport.

Intervention de Legrand Michel

Informations client
Nom : Jean Prénom : Galoup Adresse : 23 rue des pins
Informations chaudière
Marque : La marque Modèle : Le modele Numéro de série : 56648 Date de mise en service : 12/06/2022
Informations intervention
Date de l'intervention : 23/05/2033 Temps passé : 01:50 Description : Ceci est une description

IV) Manuel de déploiement

a) Android

Dans le dossier android, ouvrez le dossier ProjetInteg dans android studio puis lancez l'émulateur.

b) API

Dans le dossier API Laravel, placez le dossier ProjetIntegBaldassini sur votre serveur.

Configurez la connexion à votre base de données dans le fichier .env.

```
DB_CONNECTION=mysql
DB_HOST=lakartxela
DB_PORT=3306
DB_DATABASE=fbaldassini_sym
DB_USERNAME=fbaldassini_sym
DB_PASSWORD=fbaldassini_sym
DB_TABLE_PREFIX=projetInte_
```

Enfin, ouvrez un terminal de commande dans le dossier ProjetIntegBaldassini et lancez la commande `./artisan migrate`

Cette commande va permettre de créer dans votre base de données la table Rapport.

c) VueJS

Ouvrez un terminal de commande dans le dossier VueJS/ProjetIntegBaldassini.

Lancez la commande yarn dev et dans l'URL d'un navigateur web, tapez <http://localhost:3000/> afin de pouvoir consulter la liste des rapports enregistrés.