



my service

PARDINI Raphaël — MIMOUN Avi

github.com/av1m/my-service

01

A PROPOS DU SITE

Explication du
contexte, des
objectifs, attentes...

02

PRESENTATION DU BACKEND

Comment le backend
a été pensé et
architecturé

03

PRESENTATION DU FRONTEND

Création de
l'application VueJS

04

CI / CD

Intégration et
déploiement continu

05

DEMO

Présentation de
l'application VueJS

06

ÉVOLUTION

L'évolution souhaitée
pour le site (frontend et
backend)

INTRODUCTION

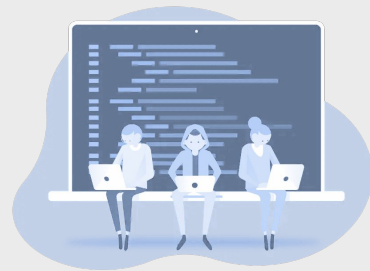
1. Motivation
2. Description du service



1. Architecture
2. Base de données
3. Référence API

BACKEND

TECHNOLOGIES



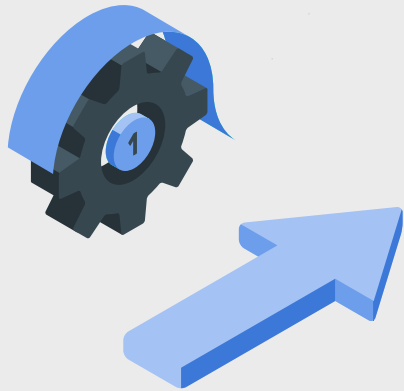
JWT express



OPENAPI
INITIATIVE

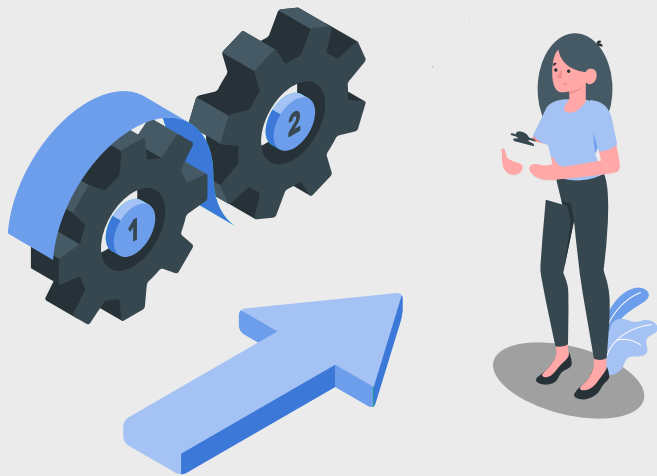


Architecture



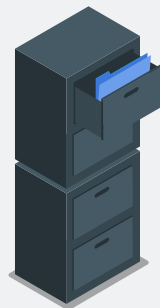
- Indépendant du Frontend
- Proposé un web service REST
- Utilisation de TypeScript
- Découpage du code
- Logique simplifié

Architecture



Découpage du code

Les composants de l'application sont réutilisables et fonctionnent à travers un principe d'encapsulation



router.ts

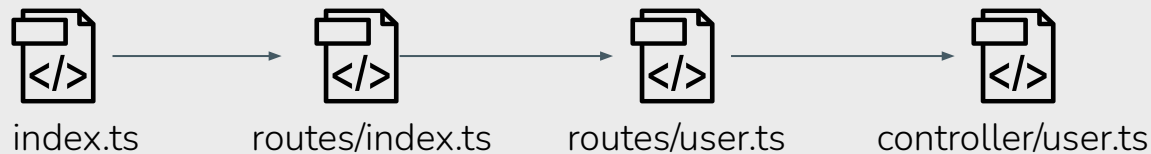


service.ts



user.ts

Architecture



Logique

1. L'utilisateur requête le serveur
2. Le serveur cherche la route associée dans le dossier des routes
3. Le serveur vérifie qu'une telle route avec telle méthode existe
4. Le serveur récupère le contrôleur associé à cette route



- Pas de schéma en dur
- Les schémas sont dans le backend (dans des fichiers .ts)
- Des “migrations” sont faites à chaque démarrage du back
- Stockés sous forme de JSON (fichiers)
- Facilite le retour de l'API (étant déjà en JSON)



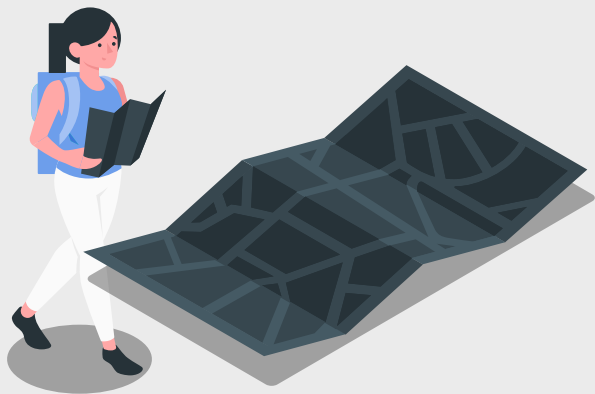
Les identifiants de la base de données sont stockés dans l'environnement et non dans le code !

Référence API

Respect de la spécification
OpenAPI



Collections Postman



Base URL

127.0.0.1:3000 ou my-service-ts.herokuapp.com

Authentification et Autorisation

Bearer token et JWT



Middleware

- Vérification du token
- Vérification des ID (conforme MongoDB)
- Vérification des autorisations

Endpoints

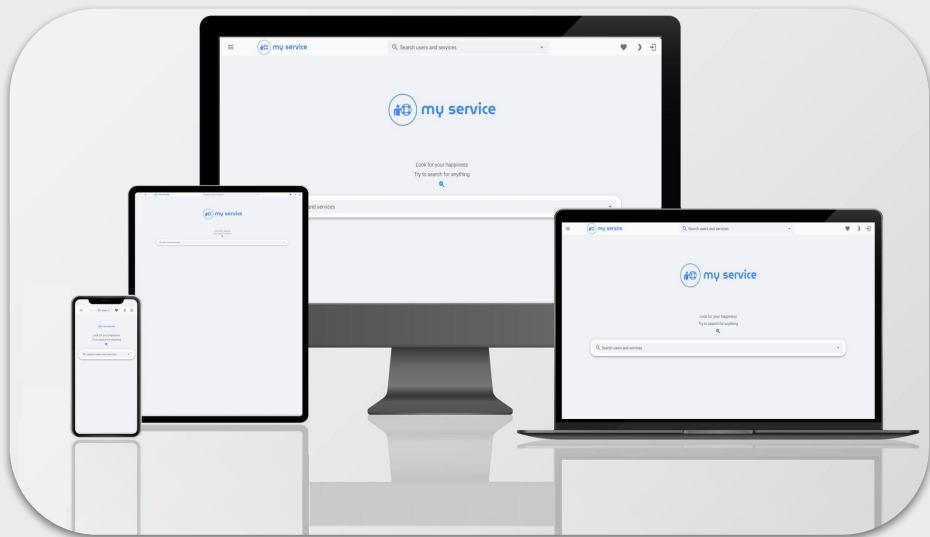
- /user/ /:id /upload/:id /payment
- /service/ /:id /upload/:id /random/:count
- /tags/

Méthodes HTTP

GET, POST, PATCH, DELETE

Frontend

VueJS

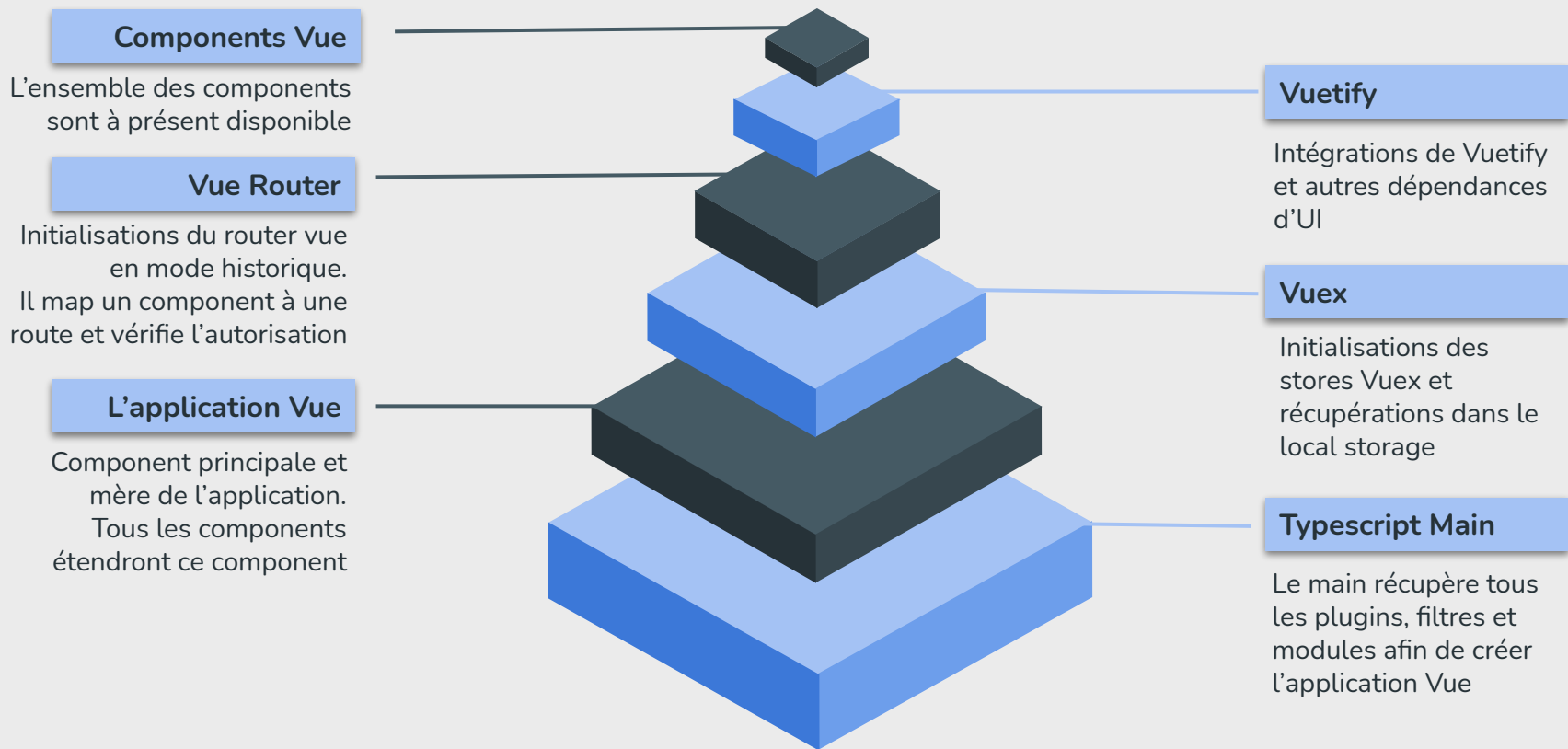


1. Architecture
2. Ergonomie
3. Paiement



Architecture

Résumé



Ergonomie



MATERIAL DESIGN

Utilisation des règles de Google pour un visuel ergonomique, épuré et harmonieux



PROGRESSIVE WEB APPS

Permet à notre site de fonctionner sur toutes les plateformes (tablette, PC, smartphone...) et de s'adapter à la taille de l'écran



GESTION DES PAGES

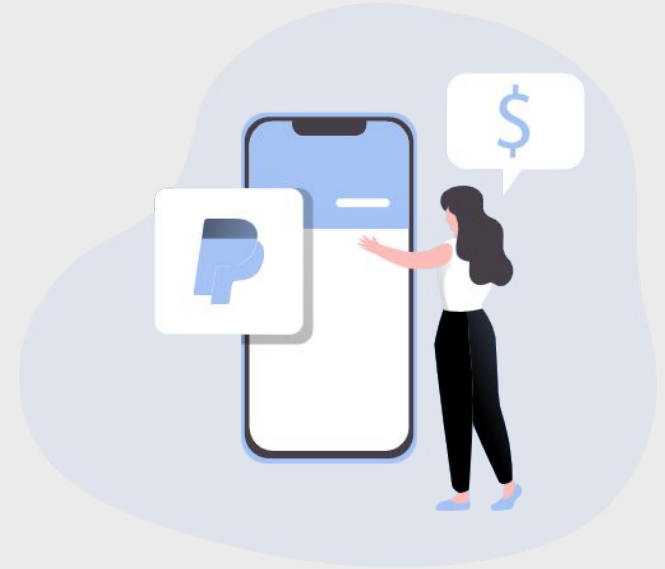
Nombre de pages différentes faible et plan du site facilement compréhensible
Accès rapide à toutes les pages depuis l'appbar



DEUX THÈMES DISPONIBLE

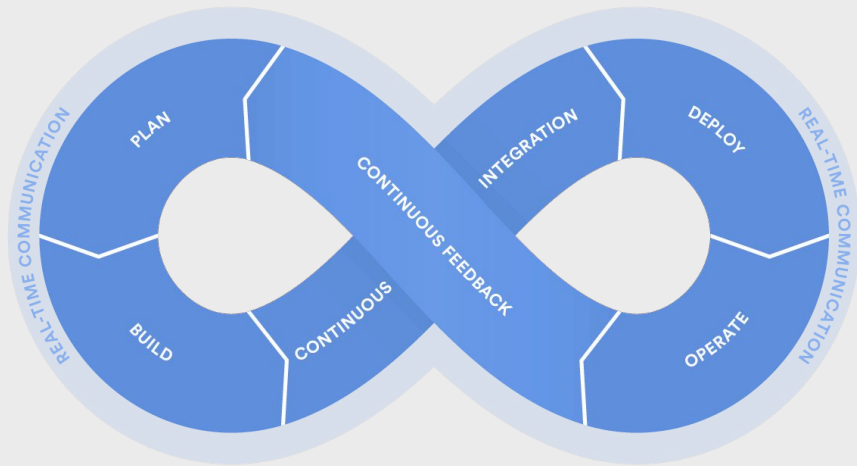
Possibilité de naviguer sur tout le site en mode clair ou en mode sombre

- Créé par Elon Musk
- Utilise l'API V2 de PayPal
- Envoi à l'API `/user/payment/`
- Disponible sur toutes les plateformes



Livraison Déploiement Continu

CI / CD



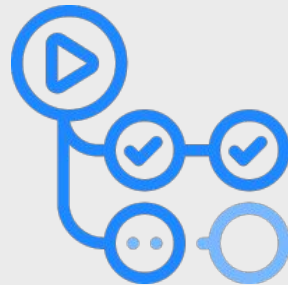
1. Github Actions

2. Github Pages

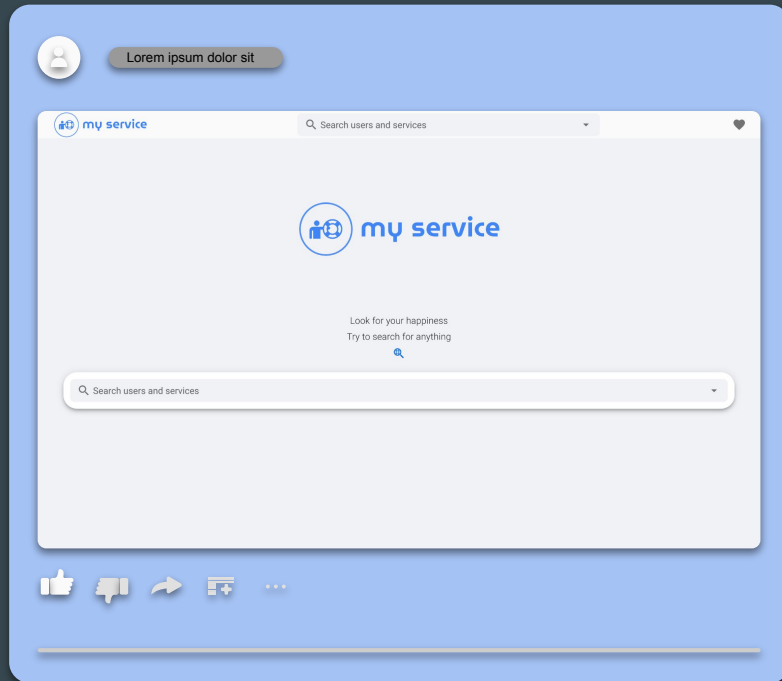
<https://av1m.github.io/my-service>

3. Heroku

<https://my-service-ts.herokuapp.com>



Présentation de notre site



ÉVOLUTIONS À VENIR



Etape 1

Ajout de notation des
profils et services



Etape 2

Création d'une
messagerie instantanée



Etape 3

Possibilité d'échanger
un service contre un
autre





Merci !



Avez-vous des questions ?

Code source

<https://github.com/av1m/my-service>

Démo

<https://av1m.github.io/my-service>

API

<https://my-service-ts.herokuapp.com>