

# PROJET

---

# Development WEB

---

*SANOU*

*Salimata Ana*

# Sommaire

Introduction : Sujet du  
projet et objectifs.

01

04

Ce qui a bien/mal  
marché

Choix techniques/Travaux  
réalisés

02

05

Ce qui était difficile/Ce  
que vous avez appris

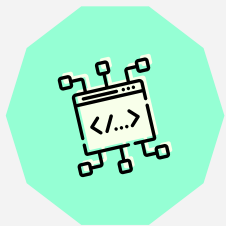
Organisation du  
travail

03

06

Conclusion





01



# Introduction

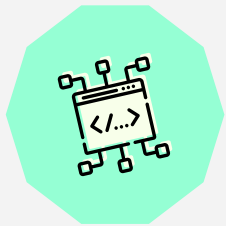
---

# Introduction

Mon projet est une application de gestion de livraison de colis, utilisant Spring Boot et Thymeleaf pour le backend et Vue.js pour le frontend, offrant une solution complète pour suivre et gérer les colis.



**L'objectif** de ce projet est de créer une application qui facilite la gestion et le suivi des livraisons de colis, permettant aux utilisateurs de suivre leurs colis.



# 02



## Choix techniques

# Choix techniques (Outils)



## Backend

Langage : Java

Framework : Spring Boot.

Moteur de template : Thymeleaf. (Dans la version monolithique)

Base de données: MYSQL

Architecture: API RESTful pour faciliter la communication entre le frontend et le backend.



## Frontend:

Framework : Vue.js pour la création d'une interface utilisateur dynamique.

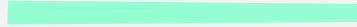
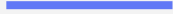
Bibliothèques associées : Axios pour les requêtes HTTP

## Outils de développement

IDE : IntelliJ IDEA pour le backend, Webstorm pour le frontend, HTTPie pour faire des requêtes de tests

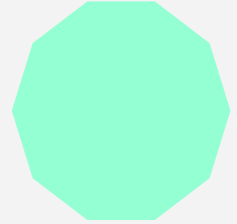
Contrôle de version : Git, GitHub

# Choix d'architecture



Notons qu'on a eu deux versions du projet :

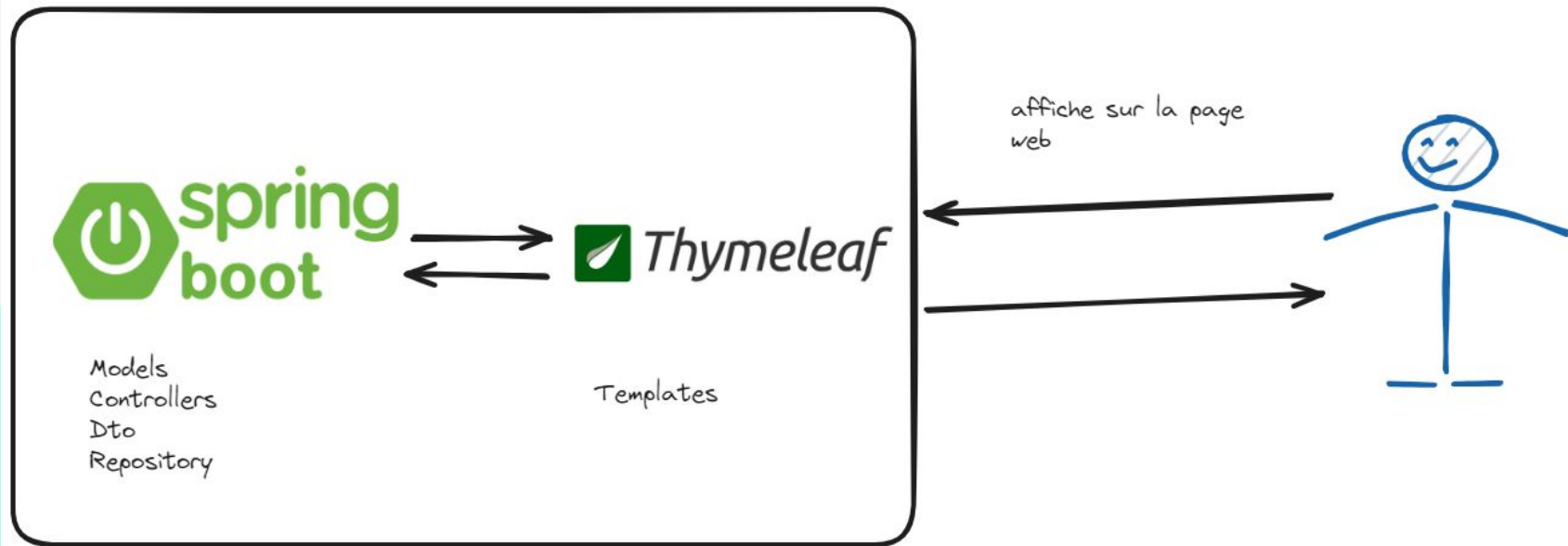
- Version 1 : Monolithique(Spring boot + Thymeleaf)
- Version 2 : Back-end(Spring Boot) et front-end(Vue Js)



# Architecture V1



Version 1 : Monolithique







# Architecture V2

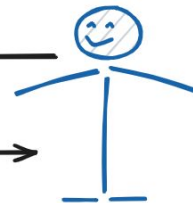
Version 2 : back-end front-ent séparé

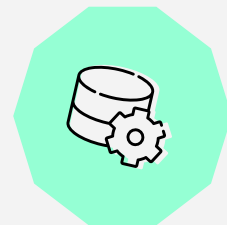


POST  
GET



affiche sur la page web



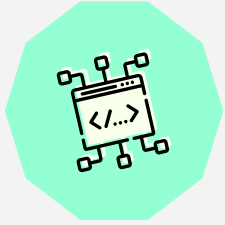


# — Les réalisations

- Les pages réalisations :
  - Authentification, (V1 et V2)
  - Création de compte, (V1 et V2)
  - Création de colis, (V1 et V2)
  - Suivi de colis, (V1 et V2)
  - Ajoute de status de colis, (V1 et V2)
  - la liste des colis, (V1 et V2)

(Réalisées avec Thymeleaf dans la V1 et avec Vue Js dans la V2)

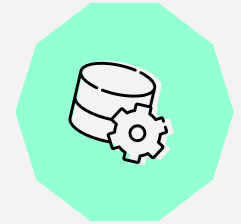
- Mise en place d'un REST API qui expose les données du Back-End



# 03



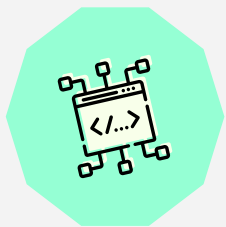
## Organisation du travail



# — Organisation du travail

Toutes les tâches de ce projet ont été réalisé par moi- même. Cela comprends :

- Controllers
- Models
- Templates
- Repositories
- Interface Frontend
- Appel des différents API



# 04

Ce qui a

bien

marché/mal

marché

# Ce qui a marché

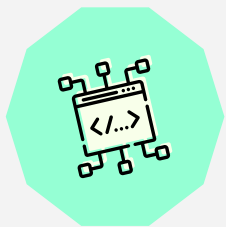


- Suivre les standards d'écriture de code java [Code Conventions for the Java Programming Language: 9. Naming Conventions](#)
- Organisation du code en suivant les bonnes pratiques ([Best Practices for Spring Boot Project Structure](#))
- Intégration des apis pour la communication entre le back-end et front-end
- Mise en place de l'authentification et inscription (V1 & V2)
- Creation et liste de colis (V1&V2)
- Création de l'état d'un colis (V1&V2)
- Suivre les détails d'un colis (V1&V2)

# Ce qui n'a pas marché



- Intégration de la geolocation dans l'application afin de trouver facilement l'emplacement d'un colis grâce à la position du livreur
- Affection d'un colis à un livreur
- Gérer l'authentification avec un Jwt entre Spring boot et Vue Js



# 05

— Ce que j'ai  
appris/Ce qui  
était difficile



# Ce que j'ai appris



- Développement d'applications web avec Spring Boot.
- Gestion des dépendances[maven et npm]
- Intégration des bases de données avec JPA et MySQL.
- Sécurisation des applications.
- Utiliser des composants avec Vue.js
- Intégration API avec axios pour la communication entre backend et frontend.



## Ce qui était difficile

- La partie Sécurité (V1 & V2)
- Mise en place d'un map pour localiser un paquet :  
malheureusement ça ne marche pas pour le problème du temps.
- Gestion des erreurs API lors de la communication entre backend et frontend (Utilisation des bons codes HTTP pour prise en compte de l'erreur)



## 6. Conclusion

---

En perspective

- ❖ Attribuer un colis à un utilisateur en fonction de son rôle.
- ❖ Page Google Maps pour afficher la localisation des colis