

# Premier Stage LocaService

Introduction

Présentation de l'Entreprise

Objectifs du Stage

Technologies et Outils Utilisés

Frontend : Vue.js (TypeScript)

Backend : Kuzzle

Chart.js

NodeMailer

Réalisations

1. Prise en Main du Projet et Premiers Tests
2. Optimisation de la Gestion des Données
3. Amélioration de l'Interface Utilisateur
4. Intégration et Amélioration des Graphiques
5. Sécurisation et Gestion des Utilisateurs
6. Gestion des Capteurs et des Groupes
7. Optimisation du Code et Responsivité

Compétences Acquises

Conclusion

## Introduction

Dans le cadre de mon stage chez **Loca Service**, j'ai développé une solution web permettant d'afficher et de traiter les données issues de capteurs IoT en temps réel. Ce projet m'a permis d'acquérir des compétences en **développement full stack**, en gestion de **base de données IoT**, ainsi qu'en **optimisation et sécurisation d'une application web**.

## Présentation de l'Entreprise

**Loca Service** est une société spécialisée dans la **location de meubles réfrigérés**. L'entreprise privilégie une gestion optimisée de ses ressources via le **smart building**, en utilisant des capteurs pour surveiller la **température, l'humidité, la qualité du réseau et l'ouverture des portes**. Fondée par Pascal Bouve en 1981 et ensuite reprise par son fils Aurélein Bouve en 2016. Ce dernier porte énormément d'attention à ce que l'entreprise reste en constante innovation et ne cesse de se réinventer, d'explorer des projets divers et variés. Le projet qui m' a été assigné poursuit cette idée.

# Objectifs du Stage

À mon arrivée dans l'entreprise, mes missions principales étaient :

- **Me familiariser avec de nouvelles technologies** (Vue.js, TypeScript, Kuzzle, Chart.js, NodeMailer).
- **Concevoir une solution permettant de visualiser et d'analyser en temps réel les données des capteurs.**
- **Optimiser l'affichage et la navigation** du site web pour améliorer l'expérience utilisateur.
- **Sécuriser l'accès aux données** avec un système de connexion et de gestion des utilisateurs.

## Technologies et Outils Utilisés

### Frontend : Vue.js (TypeScript)

J'ai développé l'interface utilisateur avec Vue.js. L'utilisation de TypeScript a renforcé la rigueur et la sécurité du code.

### Backend : Kuzzle

Kuzzle a servi à la gestion des capteurs et à la communication avec la base de données. Cette technologie m'a permis de **créer, modifier et stocker** les données en **temps réel** via WebSockets.

```

▼ ☐ 7niPVJAB42UrKSq9LrZU
  ▾ Object
    type: "capteurTemp"
    deviceId: "Proto2.1_TH_4"
    devEUI: "FAEB45C1F999A033"
    ▾ humidity: Object
      humidity: 66.9
    ▾ temperature: Object
      degree: 22.3
    ▾ dateFormat: Object
      dateFormat: "26-06-2024 14:39:48"
    ▾ reseau: Object
      reseau: -69
    ▾ moyenne: Object
      last: true
    ▾ alert: Object
      tempMax: 26
      tempMin: 16
      compteur: 0
    ▾ minMax: Object
      minTemp: 19.9
      maxTemp: 23
      minHumi: 66.9
      maxHumi: 76.3
    ▾ _kuzzle_info: Object
      author: null
      createdAt: "26/06/2024, 14:39:48"
      updatedAt: "27/06/2024, 09:29:29"
      updater: "Maxime"

```

Exemple de données enregistrées sur Kuzzle

## Chart.js

L'intégration de Chart.js a permis d'**afficher les données sous forme de graphiques interactifs**, offrant aux utilisateurs la possibilité de **zoomer, filtrer par période et exporter les données** en PNG ou CSV.

## NodeMailer

J'ai implémenté NodeMailer pour l'envoi d'**alertes par e-mail** lors du dépassement de seuils prédéfinis, permettant une réaction rapide aux anomalies.

```

const nodemailer = require("nodemailer");
let mailConfig: any;

mailConfig = {
  host: 'smtp.ethereal.email',
  port: '587',
  auth: {
    user: 'marietta.mccullough34@ethereal.email',
    pass: 'aq6dcfAaU6s7x6wHHf'
  }
}

let transporter = nodemailer.createTransport(mailConfig);

// Fonction pour envoyer des messages
async function messageLimitTemp(device, compteur, date, limiteMax, limiteMin, temp){
  try {
    const info = await transporter.sendMail({
      from: '"Test Kuzzle" <marietta.mccullough34@ethereal.email>',
      to: "marietta.mccullough34@ethereal.email",
      subject: "Dépassement des limites du capteur "+device+" le "+date,
      text: "Le capteur de température "+device+" a atteint "+temp+"°C et a dépassé les limites données ( Max: "+limiteMax+"°C et Min: "+limiteMin+"°C)",
      html: "<b>Le capteur de température "+device+" a atteint "+temp+"°C et a dépassé les limites données ( Max: "+limiteMax+"°C et Min: "+limiteMin+"°C)"
    });
    console.log('Email envoyé :', info.response);
  } catch (error) {
    console.error("Erreur lors de l'envoi de l'email :", error);
  }
}

```

Code du NodeMailer

## Réalisations

### 1. Prise en Main du Projet et Premiers Tests

Sans connaissance préalable des outils utilisés, j'ai consacré les premiers jours à me former sur Vue.js, Kuzzle et la gestion des données IoT. J'ai débuté par l'affichage des données de température et d'humidité, ce qui m'a permis de maîtriser les interactions entre le frontend et le backend.

### 2. Optimisation de la Gestion des Données

- J'ai restructuré les documents JSON pour **réduire le nombre de requêtes** et optimiser le stockage des données.
- J'ai mis en place un **système d'abonnement en temps réel** pour actualiser automatiquement l'affichage à la réception de nouvelles données.

### 3. Amélioration de l'Interface Utilisateur

- Création d'un **menu de navigation latéral dynamique**.
- Intégration de couleurs conditionnelles et d'icônes pour une meilleure lisibilité.
- Développement d'un **système de recherche** et de **tri des capteurs**.

## 4. Intégration et Amélioration des Graphiques

- **Affichage des évolutions de température et d'humidité** en courbes.
- Implémentation des fonctions de **zoom, filtrage et export des graphiques**.
- Ajout de **zones colorées** pour identifier rapidement les valeurs hors seuil.

## 5. Sécurisation et Gestion des Utilisateurs

- Développement d'un **système de connexion avec stockage des identifiants en cookie sécurisé**.
- Mise en place d'une **protection contre les attaques par force brute** (blocage après 5 tentatives échouées).
- Création de **rôles et de permissions** distinguant administrateurs et utilisateurs simples.

## 6. Gestion des Capteurs et des Groupes

- Développement d'une interface d'**administration des capteurs** (ajout, modification, suppression).
- Création d'un système de **groupes de capteurs** pour faciliter la gestion des accès et l'organisation.
- Intégration de l'**import de capteurs via fichier CSV**.

## 7. Optimisation du Code et Responsivité

- **Adaptation de l'interface** pour une expérience optimale sur mobiles et tablettes.
- **Optimisation du code**, élimination des éléments obsolètes et documentation approfondie.
- Perfectionnement des **animations** pour une navigation plus fluide.

## Compétences Acquises

Ce stage m'a permis de développer plusieurs compétences essentielles :

- **Développement full stack** avec Vue.js, TypeScript et Kuzzle.
- **Optimisation et structuration des bases de données IoT**.
- **Création d'interfaces utilisateur ergonomiques et réactives**.

- **Mise en place de systèmes d'authentification sécurisés.**
- **Utilisation avancée de Chart.js** pour la visualisation des données.
- **Gestion et manipulation des fichiers CSV** pour l'import/export de données.

## Conclusion

Ce stage a constitué une expérience enrichissante qui m'a permis d'acquérir des compétences solides en développement web et en gestion de données IoT. J'ai approfondi ma compréhension des enjeux du smart building et de la **visualisation de données en temps réel**. Cette expérience a renforcé mon autonomie dans l'apprentissage de nouvelles technologies et la résolution de problèmes complexes.