DOCUMENTATION TECHNIQUE

PROJET : NODE JS

PAR : Robensciale FAHA DEFFO

PERIODE : DU 08-07-2024 AU 14-07-2024

INTRODUCTION

Le projet consiste à développer une application backend en utilisant Node.js, Express, et Mongoose, qui sera capable de gérer des SMS marketing conformément aux exigences du Règlement Général sur la Protection des Données (RGPD) en France. Ce règlement impose des règles strictes sur la manière dont les entreprises doivent gérer les données personnelles des consommateurs. En particulier, il est crucial de permettre aux utilisateurs de se désabonner facilement des communications marketing. dans le cas d'espèces, l'utilisateur doit envoyer un message STOP qui sera par la suite traité par un serveur API qui se chargera de blacklister son numéro pour qu'il ne reçoive plus de message.

FONCTIONALITE

Réception des messages :

- Le serveur reçoit des requêtes POST envoyées par le fournisseur téléphonique, et chaque requête POST représente un SMS reçu.

Traitement des messages :

- Lorsqu'un SMS est reçu, le serveur analyse le contenu du message.
- Si le message contient le mot "STOP", le numéro de téléphone de l'utilisateur est directement ajouté à la base de données des numéros blacklistés.
- Lorsque le message est traité, le serveur retourne un statut code 200 pour spécifier que le message a été traite avec succès.

Gestion de la liste noire :

- **Le serveur permet aux administrateurs d'accéder à la liste complète des numéros blacklistés**. Lorsque la demande est faite par l'administrateur. Lorsque qu'une requête GET est envoyée, le server interroge la base de données à travers Mongoose pour récupérer tous les numéros de téléphone blacklistés. Par la suite, les numéros blacklistés sont renvoyés au format **JSON**.
- **Le serveur permet également de vérifier si un numéro de téléphone spécifique est blacklisté ou non**. Une requête GET est utilisée avec un paramètre phoneNumber pour spécifier le numéro de téléphone à vérifier. le serveur interroge la base de données à l'aide de Mongoose pour vérifier si ce numéro existe dans la liste noire. En fonction du résultat de la recherche, l'application renvoie une réponse **JSON** avec un statut 200 indiquant si le numéro est blacklisté ou non.

TECHNOLOGIES UTILISEES

MongoDB : Base de données NoSQL utilisée pour stocker les numéros de téléphone blacklistés.

Express.js : Cadre d'application web pour Node.js.

Node.js : Environnement d'exécution JavaScript côté serveur.

Mongoose : Outil de modélisation d'objets MongoDB pour Node.js, facilitant l'interaction avec la base de données.

Postman : Logiciel permettant de tester les Endpoint de l'API à travers des requêtes POST et GET.

STRUCTURE DES FICHIERS

1. server.js

Le fichier server.js est le point d'entrée principal de notre application. Il configure un serveur Express qui écoute les requêtes HTTP sur un port précis.

2. src/

Ce répertoire gère toutes les configuration et traitement des requêtes de toute l'application. Il contient 4 sous-répertoires à savoir :

- config/
  Ce répertoire contient le fichier db.js qui permet d'établir la connexion avec la base de données MongoDB à l'aide de Mongoose.

- controllers/
  Ce répertoire contient le fichier smsControler.js qui a pour rôle principal de traiter tous les messages entrants. Il vérifie le contenu du message s'il contient le mot STOP, il met le numéro correspondant dans la liste noire. Il récupère également tous les numéros dans la liste noire et recherche aussi si un numéro existe dans la liste noire ou pas

- minddelwares/
  Ce répertoire contient un fichier validateRequest.js qui définit deux middlewares pour valider des champs spécifiques des requêtes http.

- Models/
  Ce répertoire contient le fichier blacklistNumber.js qui définit la structure du document pour le stockage des numéros de téléphone blacklisté dans notre base de données MongoDB à l'aide de Mongoose.

3. Dockerfile

Ce ficher pernet de créer une image de notre application pour la rendre plus légère et portable.

4. docker-compose

Ce ficher permet de configurer les différents services qu'utilise notre application pour son bon fonctionnement.

5. .env

Ce fichier contient toutes les variables d'environnements liées à la connexion à la base de données MongoBD.

6. Readme.md

Ce ficher montre les étapes d'installation et d'utilisation de notre serveur.

GitHub: Robenscial/3NPM_EXAMS (github.com)