

Projet 3NPM

Session rattrapage

Table des matières

1. Introduction :	3
2. Contexte :	3
3. Énoncé :	3
4. Contraintes techniques :	4
5. Barème :	4
6. Astuces et techniques :	5
7. Rappels des consignes :	7
8. Rendu :	7

1. Introduction :

Le but de ce projet est de créer une application backend en utilisant Node.js, Express et Mongoose. Ce projet mettra en pratique les différentes parties et notions abordées durant le cours. Certains choix technologiques vous seront imposés pour encadrer au mieux le projet et rester dans le contexte du cours. Il est essentiel d'avoir bien suivi le cours et assimilé les différentes notions clé de Node.js, notamment son approche asynchrone et les différentes façons de l'aborder.

2. Contexte :

En France, le Règlement Général sur la Protection des Données (RGPD) impose des règles strictes sur la manière dont les entreprises peuvent collecter et utiliser les données personnelles des consommateurs. Une des exigences du RGPD est de permettre aux utilisateurs de se désabonner facilement des communications marketing, par exemple en envoyant le mot "STOP" par SMS.

Une grande chaîne de magasins de parfums aux extraits naturels, "OBD" (Ô Bois Dormant), envoie régulièrement des SMS marketing à ses clients pour les informer des promotions et des nouveautés. Pour se conformer au RGPD, ÔBD doit mettre en place un système permettant de traiter automatiquement les demandes de désabonnement envoyées par SMS. Pour cela, ils font appel à vous, développeur(se) backend pour implémenter cette fonctionnalité en une semaine.

Vous devez alors créer une API qui servira de webhook pour le fournisseur téléphonique de ÔBD. Cette API devra intercepter les SMS reçus par le fournisseur, vérifier la présence du mot "STOP" dans le message, et ajouter le numéro de téléphone de l'utilisateur à une liste noire (blacklist) si le mot "STOP" est présent. Cela empêchera l'envoi futur de SMS marketing à ce numéro.

3. Énoncé :

Votre solution consiste à mettre en place un serveur Node.js capable de recevoir des requêtes POST envoyées par le fournisseur téléphonique. Ce dit fournisseur sert de relais entre l'expéditeur et votre server ; il enverra une requête POST à chaque fois qu'il reçoit un sms. Lorsqu'un SMS est reçu, votre serveur doit vérifier si le message contient le mot "STOP". Si c'est le cas, le numéro de téléphone de l'utilisateur doit être ajouté à une base de données de numéros de téléphone blacklistés.

Votre serveur doit être capable de fournir aux administrateurs la liste complète des numéros blacklistés et permettre également de déterminer si un numéro donné est blacklisté ou non.

4. Contraintes techniques :

Comme annoncé précédemment, vous êtes dans l'obligation de respecter certaines consignes, dont :

- Langage de programmation : Node.js
- Framework http : Express
- Validateur de données : express-validator
- Système de base de données : MongoDB
- OBD (Object Data Modeling) : Mongoose

5. Barème :

a) Ressources (7 points) :

- Implémentation de la ressource pour recevoir les SMS (interception de la requête + blacklistage du N° de tél + réponse 200) : **3 points**
- Implémentation de la ressource pour récupérer tous les numéros de téléphone blacklistés : **2 points**
- Implémentation de la ressource pour déterminer si un numéro donné fait partie de la blacklist : **2 points**

b) Fonctionnalités (2 points) :

- Votre serveur peut se lancer : **1 point**
- Votre serveur peut se connecter à une base de données : **1 point**

c) Sécurité (2 points) :

- Vous valider les informations reçues dans la requête POST de blacklistage : **2 points**

d) Bonne pratiques et propreté du code (2 points) :

- Vous avez utilisé les middlewares avec Express : **1 point**
- Vous avez bien divisé votre projet et organisé le code, écrit et produit un code lisible et facilement maintenable : **1 point**

e) Documentation (2 points) :

- Fournir une documentation technique de votre application qui doit expliquer les fonctionnalités de votre application, les technologies utilisées etc. : **1 point**

- Fournir un fichier readme.md qui explique comment faire les installations, lancer le server et obtenir les différentes ressources, etc : **1 point**

f) Bonus **(5points)** :

Dans cette partie vous êtes libre de proposer des fonctionnalités supplémentaires qui vont dans le sens de l'amélioration du projet. Ne seront pas pris en comptes les ajouts à des fonctionnalités déjà existantes ou redondantes. Toutefois, si vous êtes en manque d'inspiration, voici quelques pistes :

- Utilisation du https au lieu du http
- Capacité de relance automatique en cas de crash
- Ajout d'un index sur le champ « phoneNumber » de la collection blacklist
- Protection des routes avec un middleware d'authentification
- Ajout d'une ressource user signup/signin/logout
- Gestion d'une session avec JWT

6. Astuces et techniques :

Les requêtes POST en provenance du fournisseur téléphonique et en destination de votre Api, auront le body suivant (au format JSON) :

```
{
  "SmsSid": "SMXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "SmsStatus": "received",
  "MessageSid": "SMXXXXXXXXXXXXXXXXXXXXXXXXXXXX",
  "To": "+1234567890",
  "From": "+33612345678",
  "Message": "STOP"
}
```

Dans votre cas de figure, vous aurez besoin des champs :

- « From » qui représente le N° de téléphone de l'expéditeur du sms.
- « Message » qui représente le contenu du sms. Vous devez alors vous assurer qu'il contient le mot « STOP »

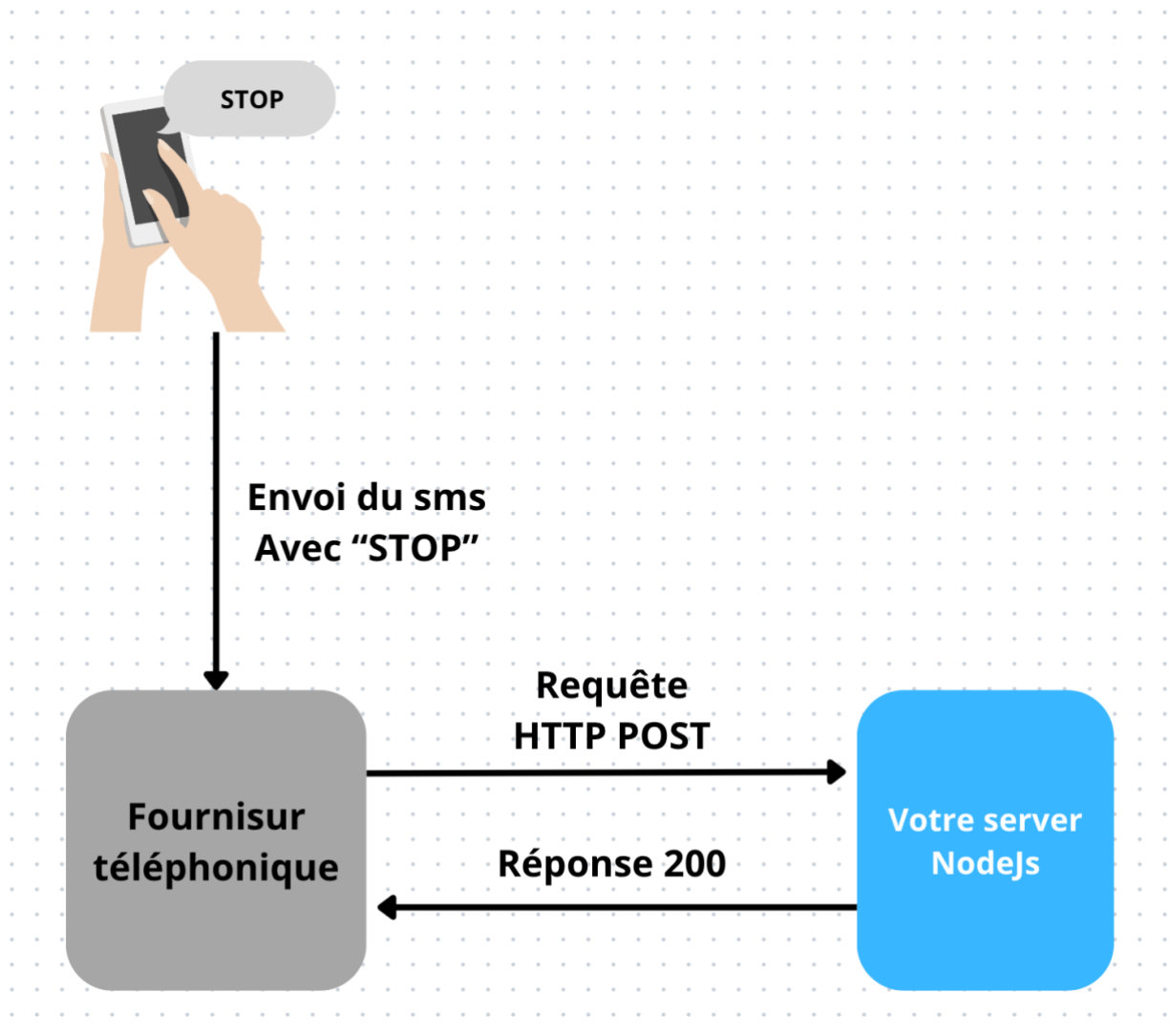
Vous pouvez structurer votre base de données comme suit :

- Collection « blacklist »
- Des documents au format suivant :

```
{  
  "_id": "ObjectId",  
  "phoneNumber": "+33612345678",  
  "addedAt": "2024-06-28T12:00:00Z"  
}
```

Chaque fois qu'il est nécessaire d'ajouter un numéro de téléphone à la blacklist, une insertion doit être effectuée dans la collection. Il est impératif de s'assurer qu'aucun doublon n'est présent.

Voici un schéma du déroulement du processus de blacklisting, à partir du sms envoyé jusqu'au traitement de la demande :



7. Rappels des consignes :

Ce projet a pour but de faire pratiquer les différentes connaissances acquises durant le cours. Il doit être réalisé individuellement. Il est alors important d'essayer de le réaliser par ses propres moyens. Tout code qui ne provient pas de vous sera considéré comme une tentative de triche et vous serez sanctionné.

Considérez qu'il y aura des entretiens techniques dans lesquels il vous sera demandé d'expliquer votre code d'une manière technique. Si vous n'êtes pas capable de l'expliquer, vous serez considérés comme ayant recours à la triche.

Vous devez effectuer le rendu dans les temps impartis. Autrement vous recevrez une note nulle. Aucun rendu envoyé autrement qu'un dépôt sur la plateforme ne sera pris en compte.

8. Rendu :

Vous devez rendre votre projet dans une archive ZIP (pas rar) qui doit comprendre le dossier du projet dans lequel doivent être retrouvés tous les fichiers et dossiers qui constituent votre projet à l'exception du dossier `node_modules` qu'il n'est pas nécessaire d'inclure (facultatif).

Attention, aucun fichier hébergé ou stocké ailleurs que dans le dossier du projet ne sera pris en compte.