

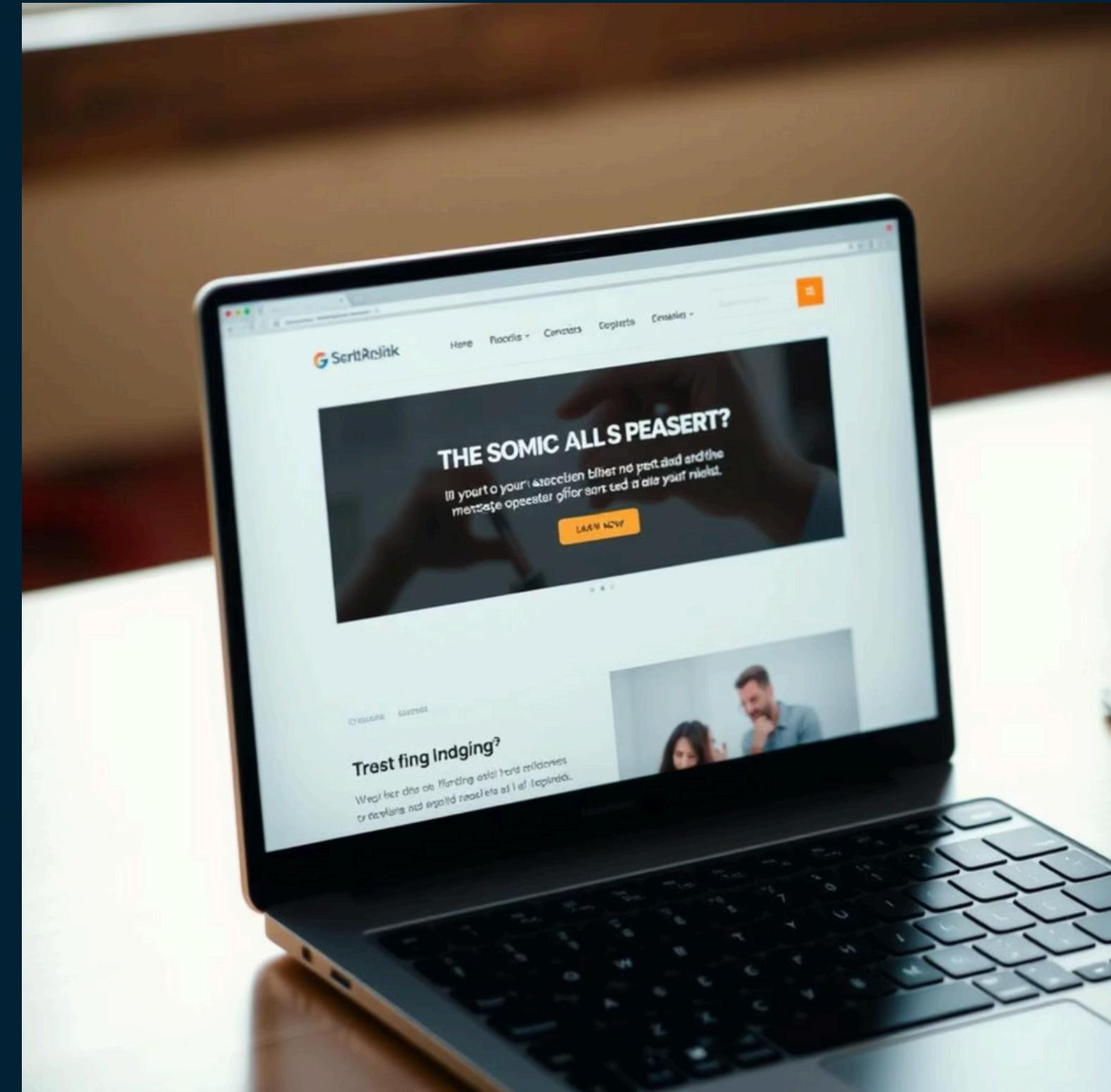


🚀 Code Challenge : Mini Application Express.js

Vous travaillez pour une petite entreprise qui veut une application simple pour gérer ses clients.

🎯 Objectif général

Construire en 72h maximum une mini application web backend avec Node.js ou Express (choix de l'étudiant), permettant de : - gérer une liste de clients via une API REST JSON (GET, POST, DELETE), - servir des pages HTML statiques (accueil, liste des clients, à propos), - tester le projet directement depuis un navigateur et via curl en ligne de commande.



API REST pour la Gestion des Clients

Vous auriez développé une API REST pour la gestion complète des clients, basée sur les opérations essentielles ci-dessous :



GET /clients

Récupère la liste complète des clients enregistrés.

POST /clients

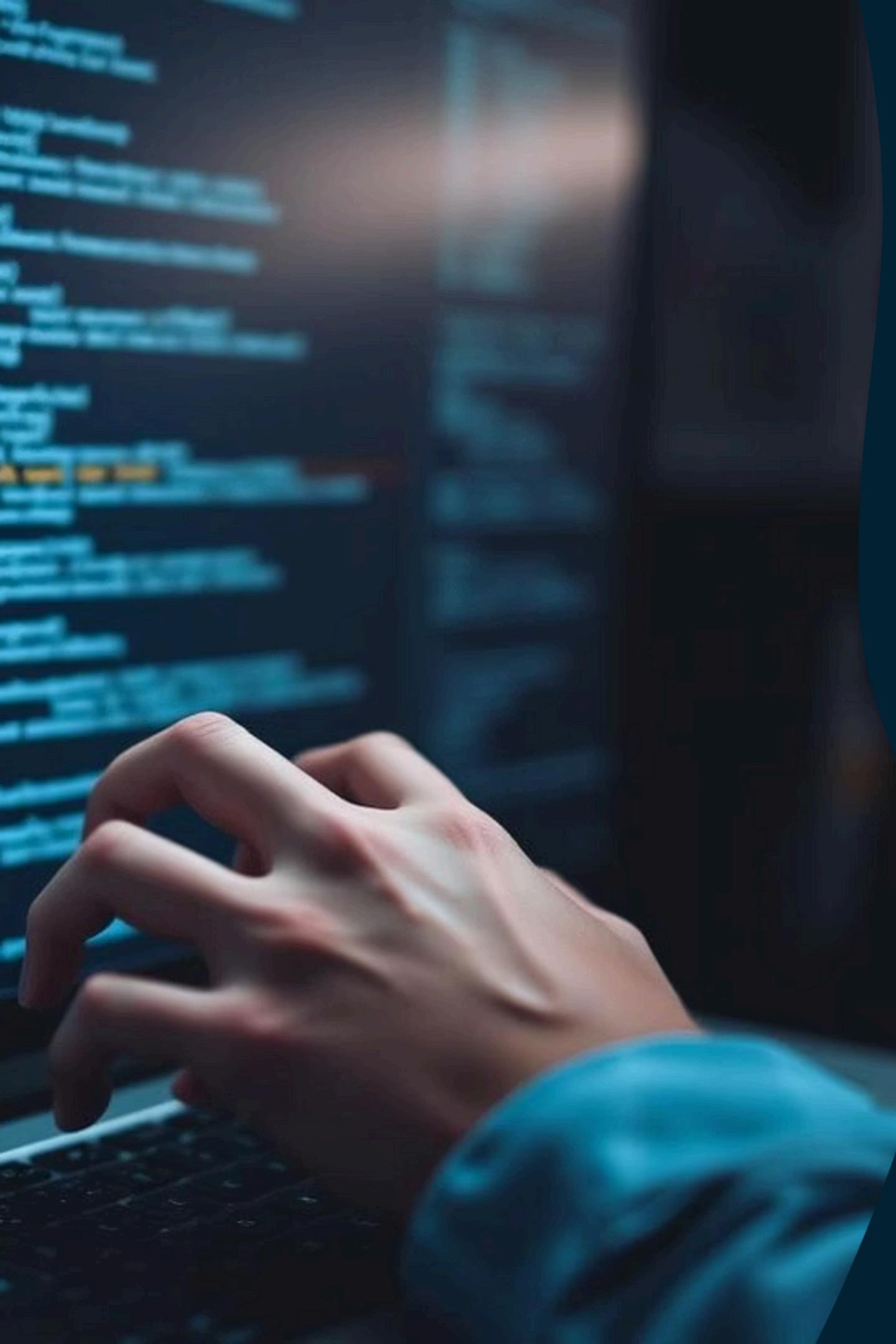
Ajoute un nouveau client, avec des données structurées en JSON (id, nom, email, téléphone).

DELETE /clients/{id}

Supprime un client spécifique en utilisant son identifiant unique.

✓ Structure attendue du projet

```
.  
|   └── server.js  
|   └── package.json  
|   └── public/  
|       |   └── index.html  
|       |   └── clients.html  
|       |   └── about.html  
|       └── style.css
```



✓ Pages HTML

- / → sert index.html (page d'accueil)
- /clients → sert clients.html (avec un formulaire HTML pour ajouter un client et afficher la liste)
- /about → sert about.html (page à propos)

Ces pages doivent être servies via express.static ou via des routes avec res.sendFile.

✓ API REST

✗ Endpoints requis

- GET /api/clients : Retourne la liste des clients au format JSON.
- POST /api/clients : Accepte un JSON {"name": "example", "email": "example@mail.com"} et ajoute un nouveau client.
- DELETE /api/clients/:id : Supprime un client par son identifiant. Les données doivent être stockées en mémoire dans un tableau JavaScript (pas besoin de base de données).

✓ Gestion des erreurs

- Si un POST ne contient pas name ou email, renvoyer une réponse HTTP 400 Bad Request.
- Si un DELETE vise un id inexistant, renvoyer une réponse HTTP 404 Not Found.

 Tests du projet Tests via navigateur

- clients.html doit contenir :
 - Un formulaire HTML pour ajouter un client.
 - Une liste des clients mise à jour depuis /api/clients.

 Tests via curl

Fournir dans le README.md des exemples tels que :

```
curl http://localhost:3000/api/clients curl -X POST -H "Content-Type: application/json" -d ;  
'{"name":"example","email":"example@mail.com"}'  
http://localhost:3000/api/clients ;  
curl -X DELETE http://localhost:3000/api/clients/1
```



Bonne chance !

Montre ton professionnalisme :

- écris un code lisible, bien structuré,
- gère proprement les erreurs,
- et livre un projet bien documenté si possible.