

260302 Mise en place du Frontend

Nous allons créer une page web contenant une carte interactive (basée sur OpenStreetMap). Quand tu cliqueras n'importe où sur cette carte, la page web enverra les coordonnées à ton API (que nous avons codée la semaine dernière). Ton serveur vérifiera si le clic est dans la zone "Paris Test" et affichera le résultat directement sur la carte sous forme de petite bulle (popup).

Pour cela, nous allons utiliser **Leaflet.js**, qui est la bibliothèque standard, légère et gratuite pour faire de la cartographie sur le web.

Voici les 3 étapes pour y parvenir.

Étape 1 : Créer le dossier "public"

Sur ton VPS, nous allons créer un dossier qui contiendra les fichiers visibles par les utilisateurs (comme les pages HTML).

1. Connecte-toi à ton VPS et place-toi dans le dossier du projet :

Bash

```
cd /var/www/alteris-geomatic
```

2. Crée le dossier `public` :

Bash

```
mkdir public
```

Étape 2 : Créer la page web (`index.html`)

C'est ici que nous allons intégrer la carte.

1. Crée et ouvre le fichier HTML dans le dossier `public` :

Bash

```
nano public/index.html
```

2. Copie et colle ce code complet. J'ai ajouté des commentaires pour t'expliquer exactement ce que fait chaque partie :

HTML

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Alteris - Carte Interactive</title>

  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.9.4/dist/leaflet.css" />

  <style>
    /* On s'assure que la carte prenne tout l'écran */
    body { margin: 0; padding: 0; }
    #maCarte { width: 100vw; height: 100vh; }
  </style>
</head>
<body>

  <div id="maCarte"></div>

  <script src="https://unpkg.com/leaflet@1.9.4/dist/leaflet.js"></script>

  <script>
    // --- INITIALISATION DE LA CARTE ---
    // On centre la vue sur Paris avec un niveau de zoom de 10
    const map = L.map('maCarte').setView([48.85, 2.35], 10);

    // On ajoute le fond de plan gratuit OpenStreetMap
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
      maxZoom: 19,
      attribution: '© OpenStreetMap contributors'
    }).addTo(map);

    // --- LOGIQUE DE CLIC ---
    // On écoute l'événement "click" sur la carte
    map.on('click', async function(event) {
```

```

// On récupère la latitude et la longitude cliquées
const lat = evenement.latlng.lat;
const lng = evenement.latlng.lng;

// On prépare notre format GeoJSON pour l'API
// Attention : en GeoJSON, on met toujours [Longitude, Latitude]
const pointGeoJSON = {
  type: "Point",
  coordinates: [lng, lat]
};

try {
  // On envoie la donnée à notre serveur Node.js
  const reponse = await fetch('/api/check-zone', {
    method: 'POST',
    headers: { 'Content-Type': 'application/json' },
    body: JSON.stringify(pointGeoJSON)
  });

  const resultat = await reponse.json();

  // On affiche le message du serveur dans une bulle sur la carte
  L.popup()
    .setLatLng(evenement.latlng)
    .setContent("<b>Résultat spatial :</b><br>" + resultat.message)
    .openOn(map);

} catch (erreur) {
  console.error("Erreur :", erreur);
  alert("Impossible de joindre le serveur Alteris.");
}

});
</script>
</body>
</html>

```

3. Sauvegarde et quitte (`Ctrl+O`, `Entrée`, `Ctrl+X`).

Étape 3 : Indiquer au serveur Node.js d'afficher cette page

Actuellement, notre serveur `server.js` ne sait faire que de l'API. Nous allons lui dire de "servir" notre nouveau dossier public lorsqu'un visiteur arrive sur le site.

1. Ouvre ton fichier serveur :

Bash

```
nano server.js
```

2. Trouve ces lignes au début de ton fichier :

JavaScript

```
app.use(bodyParser.json());

// Route de vérification simple
app.get('/', (req, res) => {
  res.send('Serveur Alteris Géomatique opérationnel 🇫🇷');
});
```

3. **Remplace la route** `app.get(...)` par cette simple ligne de code qui indique le dossier public :

JavaScript

```
app.use(bodyParser.json());

// On demande à Express de servir les fichiers du dossier "public"
app.use(express.static('public'));
```

4. Sauvegarde et quitte (`Ctrl+O`, `Entrée`, `Ctrl+X`).

Étape 4 : Redémarrer et Tester

Pour que Node.js lise ton fichier `server.js` mis à jour, il faut le relancer via PM2 :

Bash

```
pm2 restart alteris-api
```

C'est l'heure du test visuel ! ☐☐

Ouvre ton navigateur web (Chrome, Firefox, etc.) et rends-toi sur l'adresse de ton projet : ☐☐

<https://alteris.juxjux.ovh>

Tu devrais voir une magnifique carte s'afficher.

Revision #1

Created 2026-03-02 10:04:05 UTC by Julien

Updated 2026-03-02 10:06:22 UTC by Julien