
Projet Web Mapping avec OpenLayers et Node.js

Realisé par :

Malek Elmechi & Fatma Krichen

Encadré par :

Mr Riadh TEBOURBI



OpenLayers

Table of Contents

1- DESCRIPTION DU PROJET	3
2- FONCTIONNALITES PRINCIPALES	3
3- TECHNOLOGIES UTILISEES.....	3
4- TRAVAUX REALISES	4
4-1-DEVELOPPEMENT DU FRONTEND POUR LE WEB MAPPING	4
4-2-DEVELOPPEMENT DU BACKEND AVEC NODE.JS	5

List of Figures

FIGURE 1: DES FORMES GEOMETRIQUES DESSINEES SUR LA CARTE	4
FIGURE 2: MA POSITION ACTUELLE	4
FIGURE 3: DESSIN D'UN POINT SUR LA CARTE	5
FIGURE 4: TEST AVEC POSTMAN DE L'AJOUT D'UN POINT	6
FIGURE 5: DESSIN D'UNE LIGNE SUR LA CARTE	6
FIGURE 6:TEST AVEC POSTMAN DE L'AJOUT D'UNE LIGNE	7
FIGURE 7:DESSIN D'UN POLYGONE SUR LA CARTE	7
FIGURE 8:TEST AVEC POSTMAN DE L'AJOUT D'UN POLYGONE.....	8

1- Description du projet

Ce projet consiste en la création d'une application interactive de cartographie utilisant OpenLayers pour visualiser et manipuler des données géographiques de manière dynamique.

Les données des géométries dessinées par les utilisateurs (points, lignes, polygones) sont stockées dans une base de données PostgreSQL enrichie par l'extension PostGIS, qui offre des fonctionnalités avancées de gestion spatiale.

L'objectif principal de ce projet est de fournir aux utilisateurs les outils nécessaires pour :

- Dessiner des points, lignes et polygones directement sur une carte interactive.
- Localiser leur position actuelle grâce au GPS intégré.
- Sauvegarder les géométries créées dans une base de données géospatiale pour un usage ultérieur.

2- Fonctionnalités principales

- **Localisation utilisateur**
 - Permet de récupérer et d'afficher en temps réel la position GPS actuelle de l'utilisateur sur la carte interactive.
- **Dessin de géométries**
 - **Point** : Ajout d'un point ponctuel sur la carte.
 - **Ligne** : Tracé d'une ligne connectant plusieurs points.
 - **Polygone** : Création d'une surface fermée définie par plusieurs sommets.
- **Sauvegarde dans PostgreSQL/PostGIS**
 - Enregistrement des géométries dessinées avec leurs types et coordonnées dans une base de données géospatiale, assurant leur accessibilité et leur gestion efficace.
- **Affichage interactif**
 - Exploitation des capacités d'OpenLayers pour offrir une navigation fluide, des interactions intuitives et une expérience utilisateur optimisée.

3- Technologies utilisées

- **Frontend** : Développement d'une interface utilisateur dynamique et interactive en utilisant **HTML5**, **CSS3**, et **JavaScript**, avec l'intégration d'**OpenLayers** pour la gestion de la cartographie.
- **Backend** : Mise en place d'un serveur backend avec **Node.js**, accompagné du framework **Express.js**, pour gérer les API REST et les interactions entre le frontend et la base de données.
- **Base de données géospatiale** : Utilisation de **PostgreSQL** enrichie par l'extension **PostGIS** pour assurer le stockage, la manipulation et l'analyse des données spatiales de manière performante et fiable.

4- Travaux réalisés

4-1-Développement du frontend pour le web mapping

Après avoir finalisé les configurations nécessaires sur GeoServer, nous avons entamé la création de notre projet de cartographie web côté client. Cette étape a abouti au développement du **frontend**, une interface interactive permettant plusieurs fonctionnalités essentielles.

Grâce à cette interface :

- Les utilisateurs peuvent dessiner diverses formes géométriques, telles que des points, des lignes, et des polygones, directement sur la carte.
- Il est également possible de récupérer la position géographique actuelle de l'utilisateur grâce à une intégration avec le système GPS.

Ce frontend constitue la base fonctionnelle pour offrir une expérience utilisateur fluide et répondre aux besoins du projet.

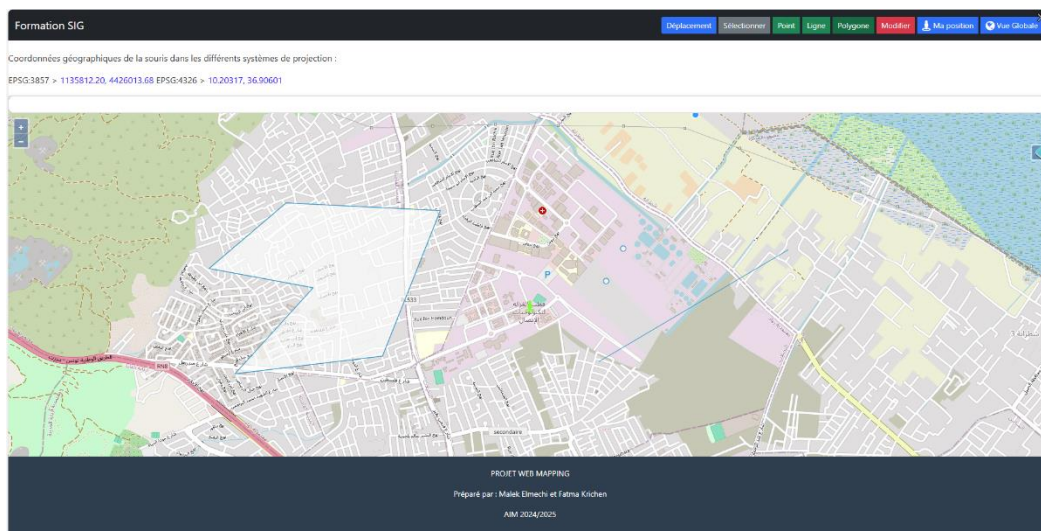


Figure 1: Des formes géométriques dessinées sur la carte

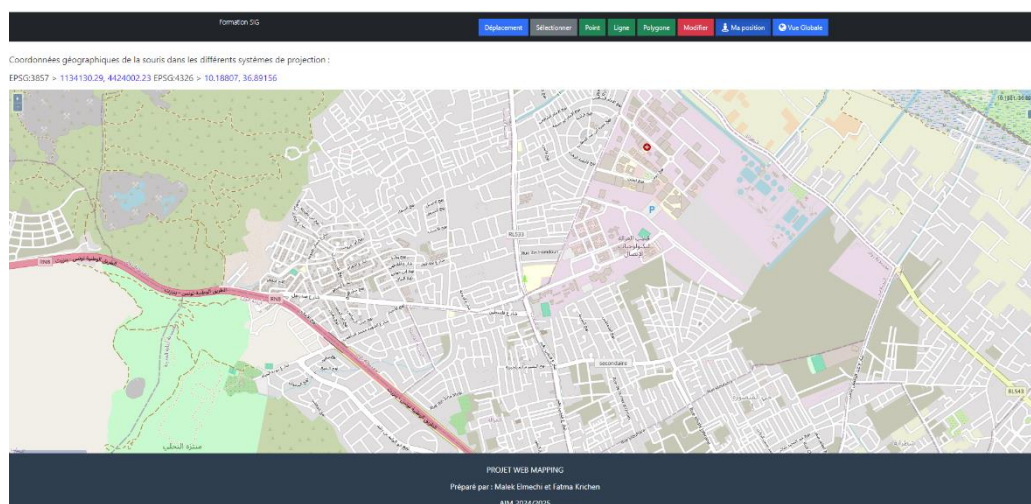


Figure 2: Ma position actuelle

4-2-Développement du Backend avec Node.js

Après la mise en place du frontend, nous avons entamé le développement du côté serveur en utilisant **Node.js**. L'objectif principal de cette étape était d'assurer la gestion des géométries dessinées sur la carte. Chaque forme géométrique (point, ligne, ou polygone) est récupérée depuis le frontend, traitée, puis sauvegardée dans une base de données **PostgreSQL** enrichie par l'extension **PostGIS** pour une gestion efficace des données spatiales.

Pour valider et tester les fonctionnalités backend, nous avons utilisé **Postman**. Ces tests nous ont permis de :

- Vérifier la réception correcte des données envoyées par le frontend.
- Confirmer l'enregistrement fiable des géométries dans la base de données.
- Assurer une communication fluide et cohérente entre le **frontend** et le **backend** via des API REST.

Cette étape a été cruciale pour garantir la synchronisation entre les différentes parties de l'application et assurer la robustesse du système global.

- La géométrie point :

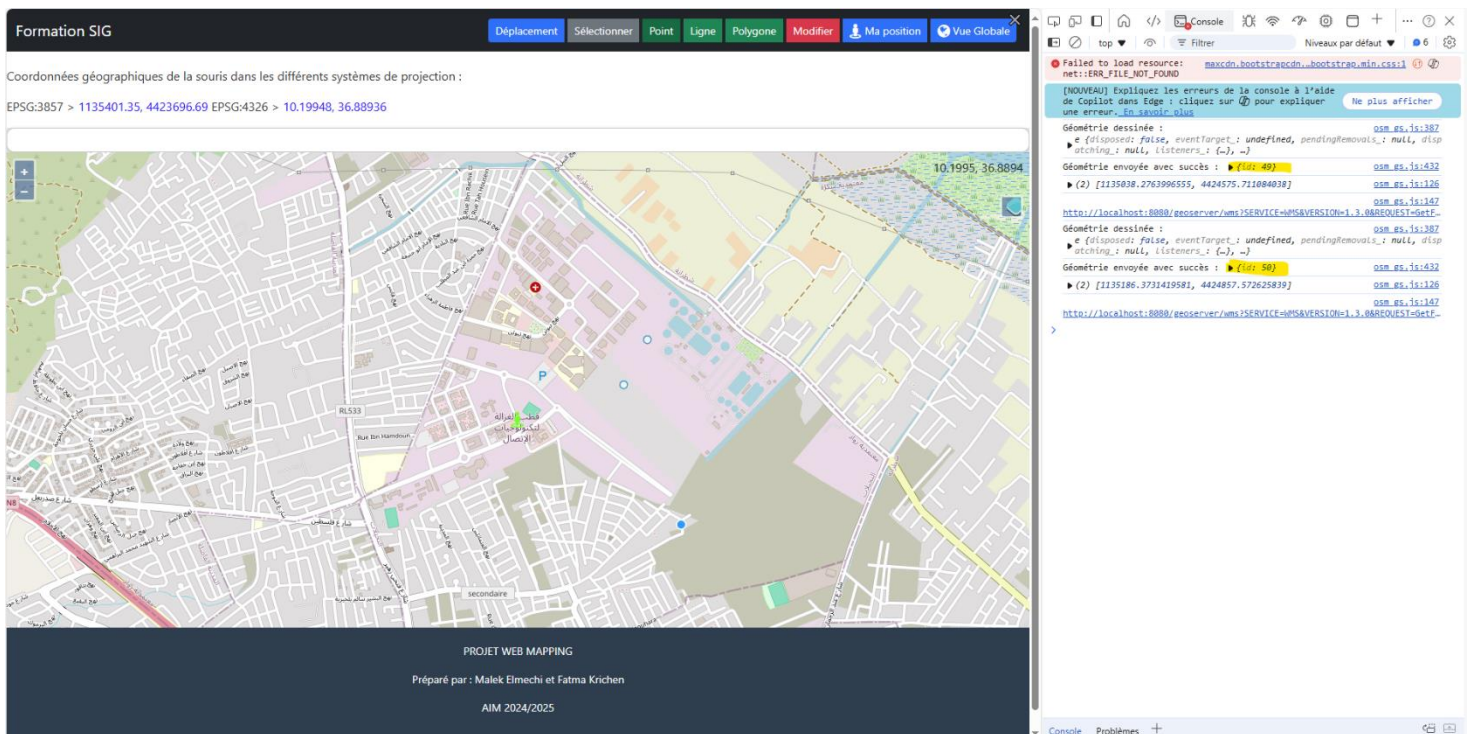


Figure 3: Dessin d'un point sur la carte

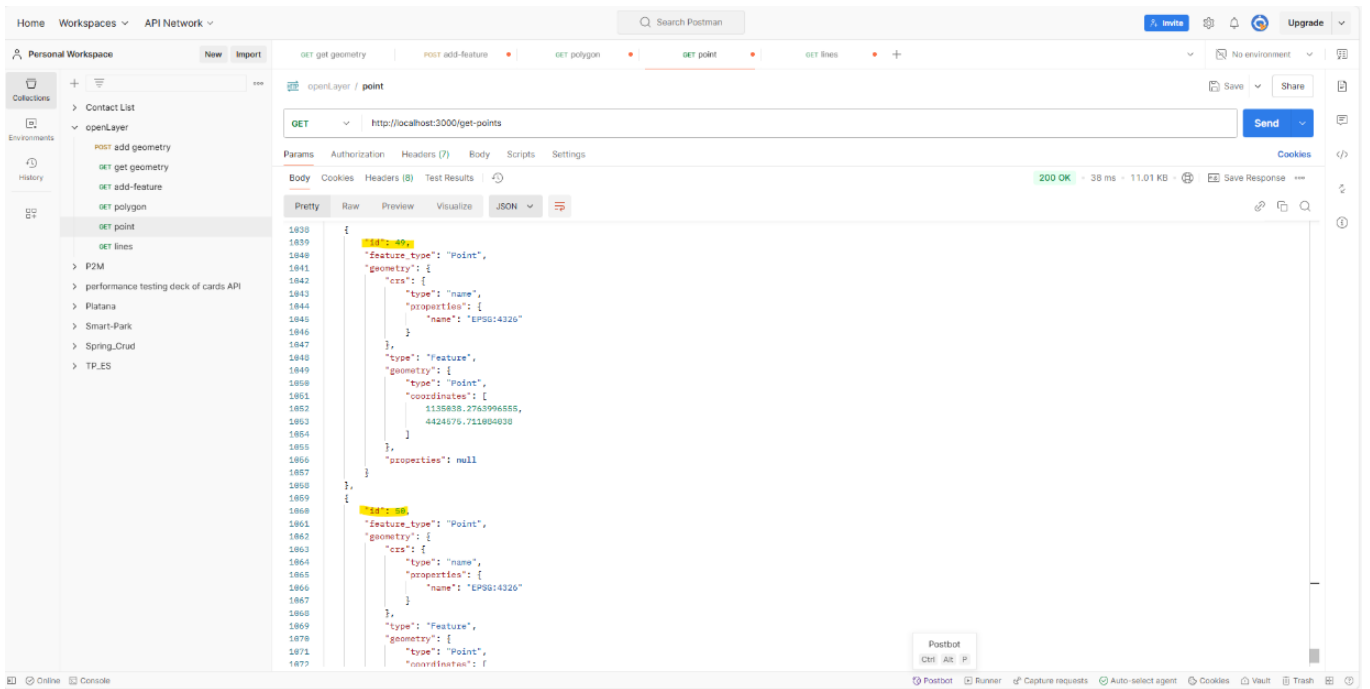


Figure 4: Test avec Postman de l'ajout d'un point

- La géométrie ligne :

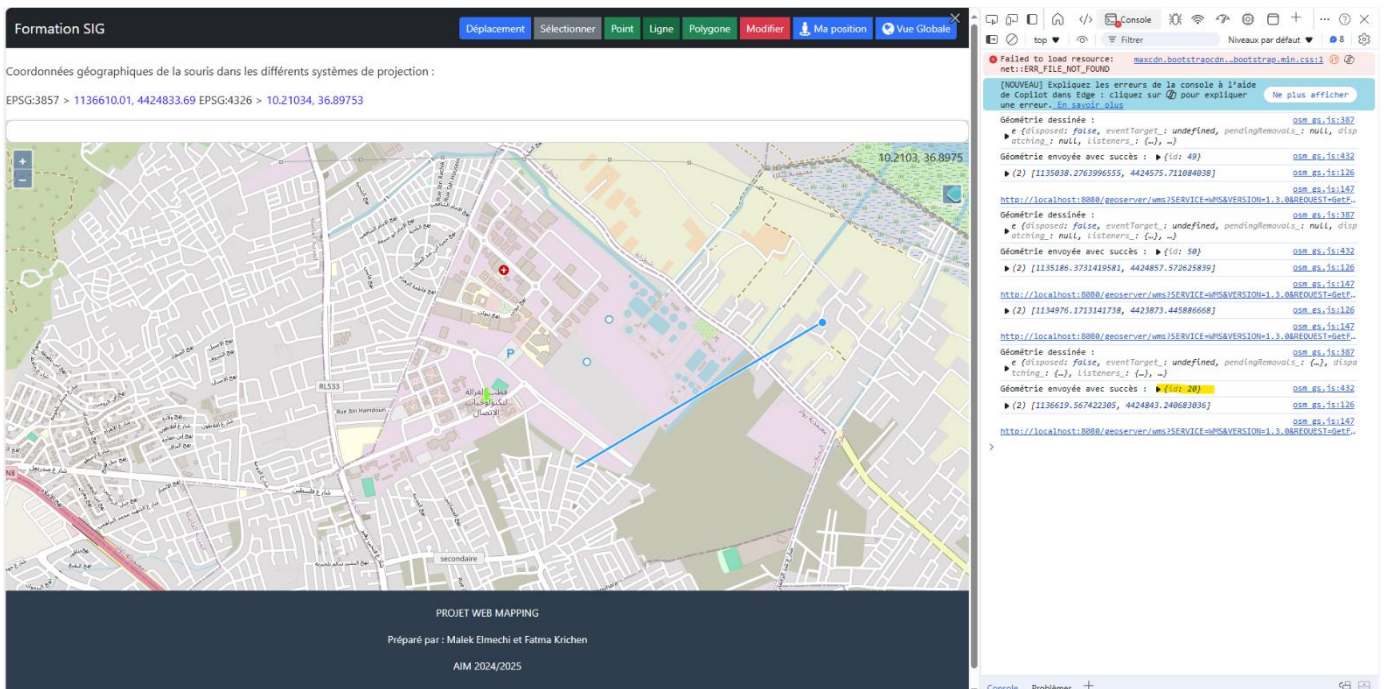


Figure 5: Dessin d'une ligne sur la carte

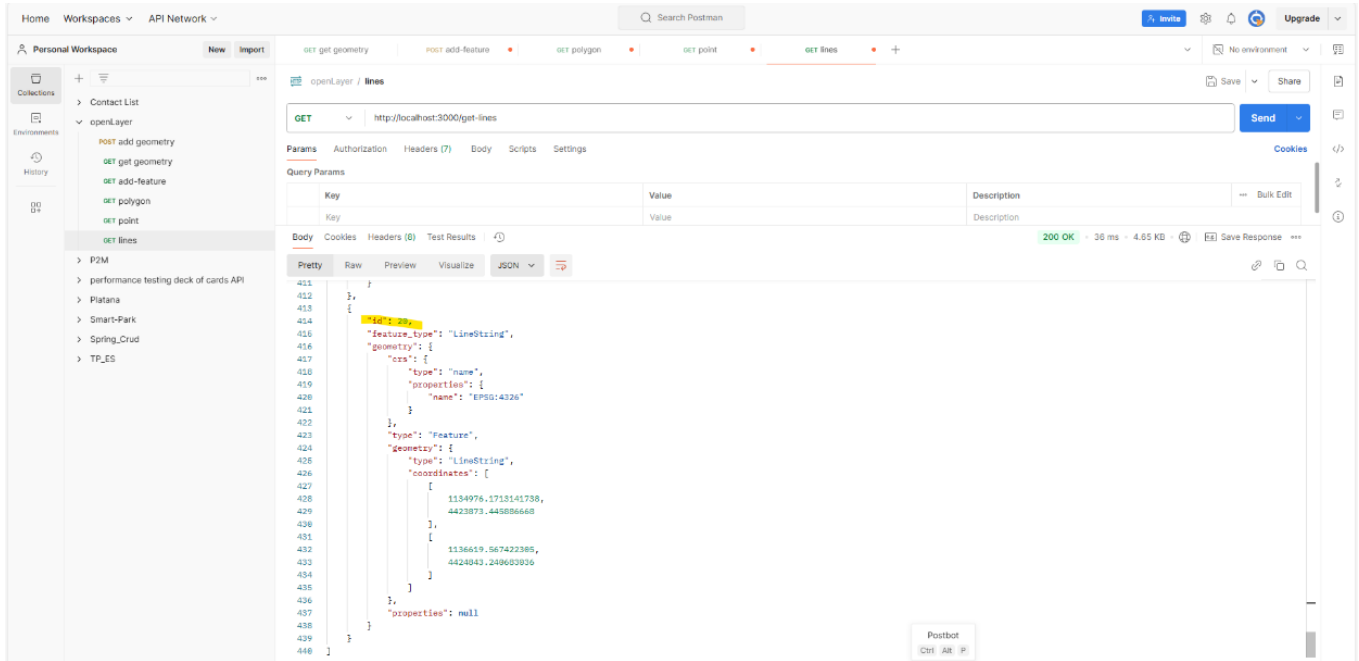


Figure 6: Test avec Postman de l'ajout d'une ligne

- La géométrie polygone :

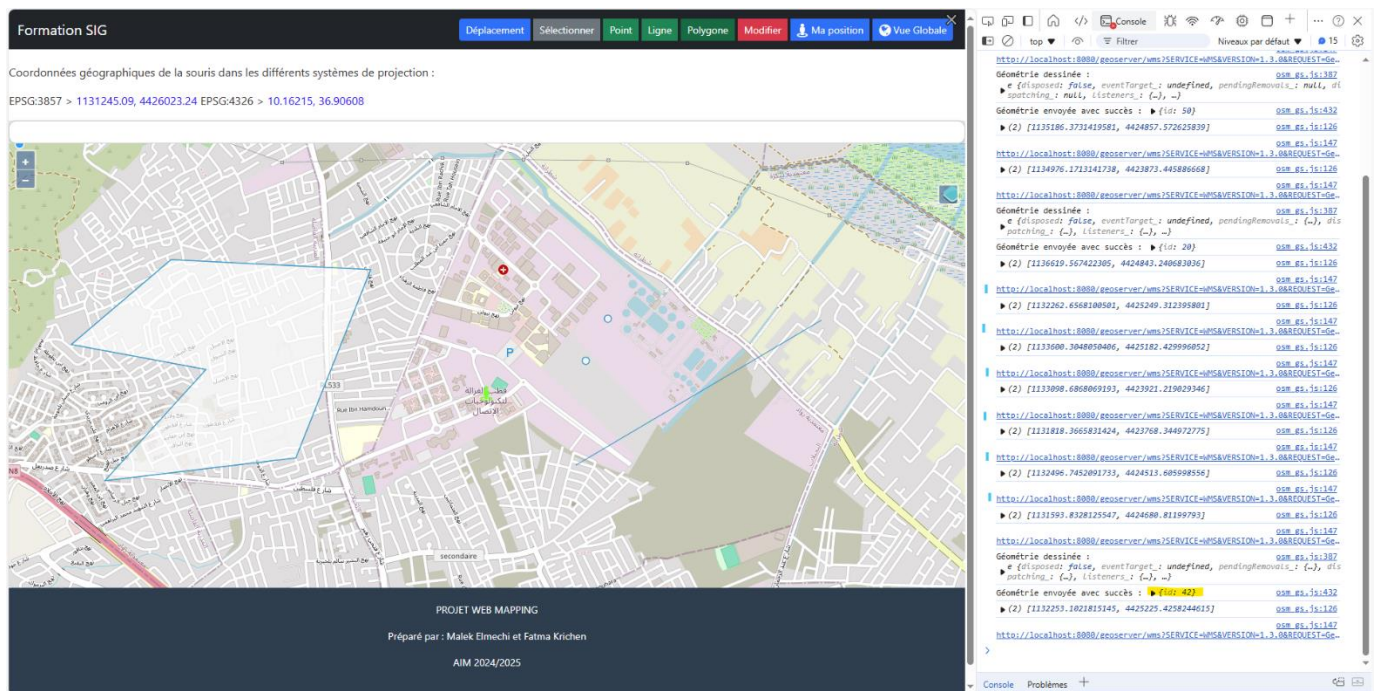


Figure 7: Dessin d'un polygone sur la carte

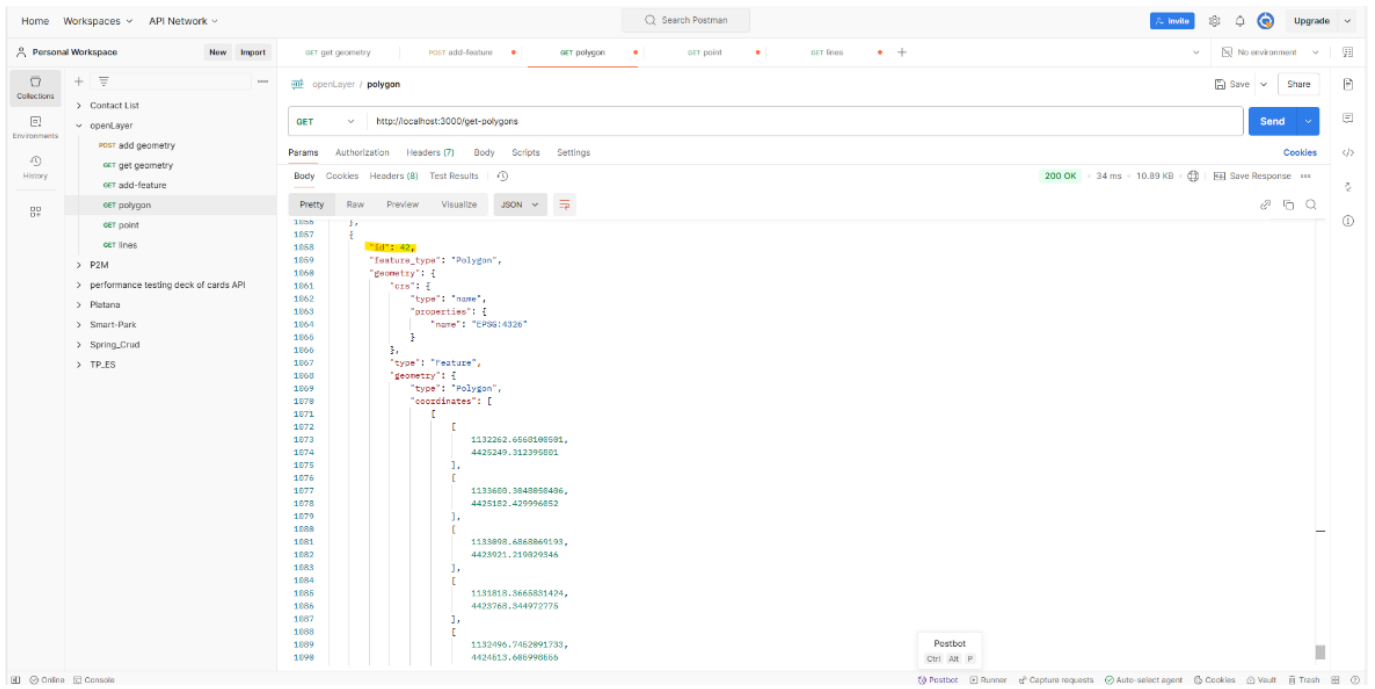


Figure 8: Test avec Postman de l'ajout d'un polygone