Leaflet Interactive Mapping Integration

Table des matières

1. Overview 2
2. Technical Architecture 2
3. Key Integrations 2
A. Geocoding (Nominatim API) 2
B. Routing & Itinerary (OSRM) 2
C. Points of Interest (Overpass API) 2
4. Data Flow: From Map to Database 3

1. Overview

The mapping module is the core component of **Trips & Roads**. It provides an interactive interface for users to visualize itineraries, search for locations, and discover Points of Interest (POI). The system integrates **Leaflet.js** with open-source APIs to avoid proprietary licensing costs.

2. Technical Architecture

The mapping logic is split into two primary scripts:

- index.js: Manages the discovery phase (geolocation, POI search via Overpass API, and Nominatim search).
- map.js: Handles the road trip creation (routing via OSRM, waypoint management, and transport strategies).

3. Key Integrations

A. Geocoding (Nominatim API)

We use the Nominatim service to transform text-based queries into geographical coordinates (Latitude/Longitude).

- **Implementation:** An event listener on the search input triggers a fetch request to the OpenStreetMap Nominatim endpoint.
- **Result:** The map view is automatically updated and a marker is placed at the found location.

B. Routing & Itinerary (OSRM)

The application calculates routes based on different transport modes (Driving, Cycling, Walking).

- **Transport Strategies:** Defined in TRANSPORT_STRATEGIES to map user choices to OSRM profiles.
- **Segment Management:** Routes are divided into segments. Each time a waypoint is added or moved, a request is sent to the OSRM server to fetch the new polyline and update the map display.

C. Points of Interest (Overpass API)

To enhance the user experience, we fetch real-time data from OpenStreetMap using the Overpass API.

- **Filtering:** Users can filter POIs by category (e.g., tourism, historic, sustenance) within a specific radius.
- **Visualization:** POIs are rendered as markers with specific icons on a dedicated poiLayer to keep the map clean.

4. Data Flow: From Map to Database

1. **Selection:** User clicks on the map or searches for a city.
2. **Geocoding:** Nominatim provides the GPS coordinates.
3. **Backend Storage:** The coordinates are sent to the RoadtripsController and stored in the geocoded_places table to ensure the itinerary can be reloaded later without redundant API calls.
4. **GPX Export:** Users can export their planned route as a GPX file. The controller reconstructs the XML structure using the stored waypoints.