

TASMA-OSM: Toponymic Atlas and Military Areas of Sardinia using Open Street Map

Nome: Mirco Aresu

Matricola: 60/73/65283

Università di Cagliari, Italia

Dipartimento di Matematica e Informatica



Introduzione

TASMA-OSM è un'applicazione web pensata per aiutare sviluppatori che vogliono integrare **OSM** nel loro progetto.

Il progetto TASMA-OSM è basato su OpenStreetMap e **GeoJSON** che permettono agli utenti di visualizzare e interagire con dati geografici e topologici.

Gli utenti possono **caricare i loro file** GeoJSON, visualizzarli come layer/strati di mappa interattivi e controllabili, **cercare** specifiche aree o città e ottenere **informazioni** dettagliate attraverso pop-up dinamici.

TASMA-OSM utilizza le librerie **Leaflet** e **MarkerCluster** per offrire funzionalità avanzate e offre una ricca base per gli sviluppatori.



Background

I dati geospaziali sono diventati sempre più importanti in molteplici settori, dalla pianificazione urbana alla ricerca ambientale. La comprensione e la visualizzazione di questi dati geografici e topologici sono cruciali per prendere decisioni informate.

Lo sfondo di questo progetto si basa sull'importanza dei dati geospaziali e sulla crescente disponibilità di tali dati per la visualizzazione e l'interpretazione delle informazioni geografiche.



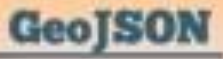
GeoJSON è uno standard aperto per la rappresentazione di semplici elementi geografici e dei loro attributi non spaziali, basato su JSON.

OpenStreetMap (OSM) è un progetto collaborativo che mira a creare una mappa del mondo libera e modificabile. OSM fornisce dati geospaziali utilizzabili in diverse applicazioni, come la visualizzazione, la geocodifica e la pianificazione dei percorsi.

La libreria **Leaflet.js** è ampiamente utilizzata per la creazione di applicazioni di mappatura web. Offre funzionalità per disegnare marker, poligoni e livelli sulla mappa, applicare stili e associare eventi.

Leaflet.js supporta anche plugin che forniscono funzionalità aggiuntive, come il **clustering dei marker**.

Tecnologie usate

Technologies of Development	
	Html/CSS are languages that allow the formatting and styling of hypertext documents.
	JavaScript is a versatile scripting language used for web development, allowing dynamic interactivity and manipulation of webpage content.
	GeoJSON is a format for encoding geospatial data, commonly used in web mapping applications, that allows for storing and exchanging geographical information in a compact and easy-to-parse manner.

Dataset

Dataset

The Sardinian Toponymic Atlas project collaborates with the Universities of Cagliari and Sassari, along with the Department for Regional Affairs of the Presidency of the Council of Ministers, to catalog and scientifically study place names in the Sardinian region. This interdisciplinary effort aims to revise regional toponymy in an organized manner, aligning with the guidelines provided by the Department for Regional Affairs. The atlas serves as a foundation for practical applications, also in bilingual approaches.

The selection of place names for each municipality is determined by the respective Municipal Administration, following specific resolutions in accordance with Law 15-12-1999, No. 482. The data collection process involves gathering information from diverse sources, including historical records, archival documents, local knowledge, surveys, and interviews with residents. Rigorous documentation, analysis, and verification are conducted to ensure the accuracy and consistency of the collected toponymic data, preserving forms in the appropriate dialect.

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [ 8.82102235, 40.12390307 ]
      },
      "properties": {
        "ITA": "Abbasanta",
        "SARDEGNA": "ABBASANTA",
        "ETNICO": "Abbasantesu/Abbasantesos/",
        "SUBREGIONE": "Barigadu",
        "ETIMOLOGIA": "SPAND1872: quivi era la stazione romana Medias Aquas, da cui ha preso il primo nome;",
        "PRIME_ATTESTAZIONI": "Rationes Decimarum Italiae, 1341 (408), Abbe Sancte; 1342 (953), Acqua Sancta"
      }
    },
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [ 9.06392097, 40.92800745 ]
      },
      "properties": {
        "ITA": "Aggiu",
        "SARDEGNA": "ÀGGIU",
        "ETNICO": "Aggesu/Aggesi/Aghiesi/Agliesi/Ajes//Azesos",
        "SUBREGIONE": "Gaddura",
        "ETIMOLOGIA": "PAULIS1987, dal latino 'Allium', 'Aglio' e poi 'Allius'",
        "PRIME_ATTESTAZIONI": "Villadaios Villa Daiois Agios"
      }
    }
  ]
}
```


Codice

Il codice dell'applicazione TASMA-OSM è strutturato in 5 passaggi chiave:

1. **Inizializzazione della mappa:** La mappa viene creata utilizzando la libreria Leaflet.js. Viene definita la posizione iniziale e il livello di zoom della mappa. Vengono anche aggiunti diversi layer di base come OpenStreetMap, immagini satellitari e stili di mappa di Mapbox.
2. **Caricamento dei dati GeoJSON:** Gli utenti possono caricare file GeoJSON che rappresentano le entità geografiche di loro interesse. Il file viene letto e il contenuto viene convertito in un oggetto JavaScript utilizzando FileReader e JSON.parse(). Vengono quindi creati marker per ogni feature nel file e viene mostrato un pop-up con le relative proprietà.
3. **Selezione delle proprietà e pop-up:** Gli utenti possono selezionare le proprietà delle feature da visualizzare nei pop-up. Per ogni proprietà di una feature, viene creato un checkbox dinamicamente. Quando l'utente seleziona o deseleziona una proprietà, le proprietà dei pop-up vengono aggiornate in base allo stato dei checkbox.
4. **Controllo dei layer:** La mappa fornisce un controllo per selezionare i layer da visualizzare. Viene utilizzata la funzione L.control.layers di Leaflet.js per creare un pannello di controllo che consente agli utenti di selezionare tra diversi layer di base. Inoltre, ogni file GeoJSON caricato aggiunge una checkbox separata per controllare la visibilità del layer corrispondente.
5. **Funzionalità di ricerca:** Gli utenti possono filtrare le feature GeoJSON digitando una query nella casella di ricerca. Ogni volta che viene digitato un carattere, l'applicazione confronta la query con le proprietà delle feature e aggiorna i marker visualizzati sulla mappa in base alla corrispondenza.

```

1 let map;
2 let markers = {};
3 let geoJsonData = {};
4 let propertyStatus = {};
5 let allLayers = [];
6 let selectedProperties = {};
7
8 let layerNames = ['atlante_toponomastico.geojson', 'AreeSpecialiMilitari.geojson'];
9
10 function initMap() {
11     map = L.map('map').setView([40.1209, 9.0129], 8);
12
13     let openStreetMapLayer = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
14         maxZoom: 19,
15     });
16
17     let satelliteLayer = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/satellite-v9/tiles/{z}/{x}/{y}?access token=pk.eyJJI...');
18
19     let mapboxStreetsLayer = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/streets-v11/tiles/{z}/{x}/{y}?access token=pk.eyJJI...');
20
21     let mapboxSatelliteStreetsLayer = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/satellite-streets-v11/tiles/{z}/{x}/{y}?access token=pk.eyJJI...');
22
23     let mapboxOutdoorsLayer = L.tileLayer('https://api.mapbox.com/styles/v1/mapbox/outdoors-v11/tiles/{z}/{x}/{y}?access token=pk.eyJJI...');
24
25     let baseMaps = {
26         openStreetMapLayer.addTo(map);
27
28         let searchBox = document.getElementById('search-box');
29         searchBox.disabled = true;
30         searchBox.style.backgroundColor = '#ddd';
31
32     };
33
34     function updatePopupContent(feature, layer) {
35
36         document.getElementById('geojson-file').addEventListener('change', function() {
37
38             function loadGeoJSONFile(file) {
39
40                 searchBox.addEventListener('input', function() {
41
42                     document.getElementById('remove-geojson').addEventListener('click', function() {
43
44                         let layerControl = L.control.layers(baseMaps, {}).addTo(map);
45
46                         map.addLayer(openStreetMapLayer);
47
48                     });
49
50                 window.onload = initMap;
51
52                 let hintButton = document.getElementById('hint-button');
53                 let hintSection = document.getElementById('hint-section');
54
55                 hintButton.addEventListener('click', function() {
56
57                     // ... (rest of the code)
58
59                 });
60
61             }
62
63         });
64
65     }
66
67 }
68
69 
```

Index

```
index.html x
1 <!DOCTYPE html>
2 <html lang="">
3 <head>
4   <title>Toponymic Atlas of Sardinia and Military Areas</title>
5   <link rel="stylesheet" href="https://unpkg.com/leaflet@1.7.1/dist/leaflet.css" />
6   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.4.1/MarkerCluster.css" />
7   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.4.1/MarkerCluster.Default.css" />
8   <link rel="stylesheet" href="styles.css">
9 </head>
10 <body>
11 <header>
12   <h1>Toponymic Atlas of Sardinia and Military Areas</h1>
13   <p>Upload your GeoJSON file to visualize the data on the map</p>
14   
15 </header>
16 <section id="controls">
17   <div class="file-upload">
18     <input id="geojson-file" type="file">
19   </div>
20   <button id="remove-geojson">Remove GeoJSON</button>
21   <div class="search-box-container">
22     <label for="search-box"></label>
23     <input id="search-box" type="text" placeholder="Search for a city or area">
24   </div>
25   <button id="hint-button" class="hint-button">Hint</button>
26 </section>
27 <div id="main-content">
28   <section id="map"></section>
29   <aside id="sidebar">
30     <h2>GeoJSON Layers</h2>
31     <div id="layer-control-panel">
32       <div id="hint-section">
33         <p class="hint">Choose the GeoJSON file from the "geojson" project folder that represents the layer. Search for<
34       </div>
35       <div id="property-checkboxes"></div>
36     </div>
37   </aside>
38   <div id="additional-popup" class="additional-popup">
39     <h3>Additional Popup</h3>
40     <p>This is the additional popup content.</p>
41     <button id="close-popup">Close Popup</button>
42   </div>
43 </div>
44 <script src="https://unpkg.com/leaflet@1.7.1/dist/leaflet.js"></script>
45 <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.4.1/leaflet.markercluster.js"></script>
46 <script src="main.js"></script>
47 </body>
48 </html>
```

Esempi d'uso

Ecco alcuni esempi di come gli utenti possono utilizzare l'applicazione TASMA-OSM:

- **Caricare un layer GeoJSON:** Gli utenti possono navigare fino alla posizione del file GeoJSON sul proprio dispositivo e selezionarlo attraverso l'opzione di caricamento. L'applicazione visualizzerà il layer selezionato sulla mappa, consentendo agli utenti di esplorare le relative feature.
- **Selezione delle proprietà da visualizzare:** Una volta caricato un file GeoJSON, gli utenti possono selezionare quali proprietà delle feature desiderano visualizzare nei pop-up. Possono farlo selezionando o deselezionando le checkbox corrispondenti alle proprietà disponibili nel pannello di controllo a destra.
- **Ricerca di aree o città specifiche:** L'applicazione offre una casella di ricerca che consente agli utenti di filtrare le feature GeoJSON in base a specifiche aree o città. Digitando il nome dell'area o della città desiderata nella casella di ricerca, l'applicazione aggiornerà dinamicamente i marker visualizzati sulla mappa per mostrare solo le feature che corrispondono alla query di ricerca.
- **Passaggio tra i diversi layer di base:** La mappa offre un controllo che consente agli utenti di passare tra i diversi layer di base disponibili. Possono scegliere tra opzioni come OpenStreetMap, immagini satellitari e stili di mappa di Mapbox. Questo consente agli utenti di personalizzare la visualizzazione della mappa in base alle proprie preferenze o alle necessità specifiche dell'analisi dei dati.

Questi sono solo alcuni esempi delle molte possibilità offerte dall'applicazione TASMA-OSM. Gli utenti possono esplorare i dati geografici, filtrare le informazioni, personalizzare la visualizzazione e ottenere una comprensione più approfondita delle caratteristiche geografiche e topologiche attraverso un'interfaccia intuitiva e interattiva.

Esempi d'uso

Toponymic Atlas of Sardinia and Military Areas

Upload your GeoJSON file to visualize the data on the map

Scegli file atlante_topon...astico.geojson Remove GeoJSON

Search for a city or area

Hint

GeoJSON Layers

- ☒ ITA
- ☒ SARDEGNA
- ☒ ETNICO
- ☐ SUBREGIONE
- ☒ ETIMOLOGIA
- ☒ PRIME_ATTESTAZIONI
- ☒ AreeSpecialiMilitari.geojson
- ☒ atlante_toponomastico.geojson

ETNICO: Santuasilisus/Santubasilisus

ETIMOLOGIA: PAULIS 1983, PITTAU 1997, BASILI continua la forma greco bizantina BASILI(S) 'reggio, regale'

PRIME_ATTESTAZIONI: Codex Diplomaticus Sardiniae, 1219, 'Sanctus Basili de Montis' Carte Volgari Campidanesi, 1226, 'Sanctu Basili de Montis'

Conclusioni

TASMA-OSM offre un'esperienza interattiva per l'esplorazione dei dati geospaziali. Gli utenti possono visualizzare mappe interattive, caricare layer GeoJSON, selezionare proprietà da visualizzare e cercare specifiche aree.

L'applicazione è flessibile, scalabile e fornisce una solida base di codice per ulteriori sviluppi e personalizzazioni.

Grazie per l'attenzione.

