



**SMART**FACTOR

## BootCamp: zajęcia 12

„Nie zawsze potrafię przewidywać,  
ale potrafię **kłaść podwaliny**.  
Bo **przyszłość** jest czymś,  
co się **buduje**”

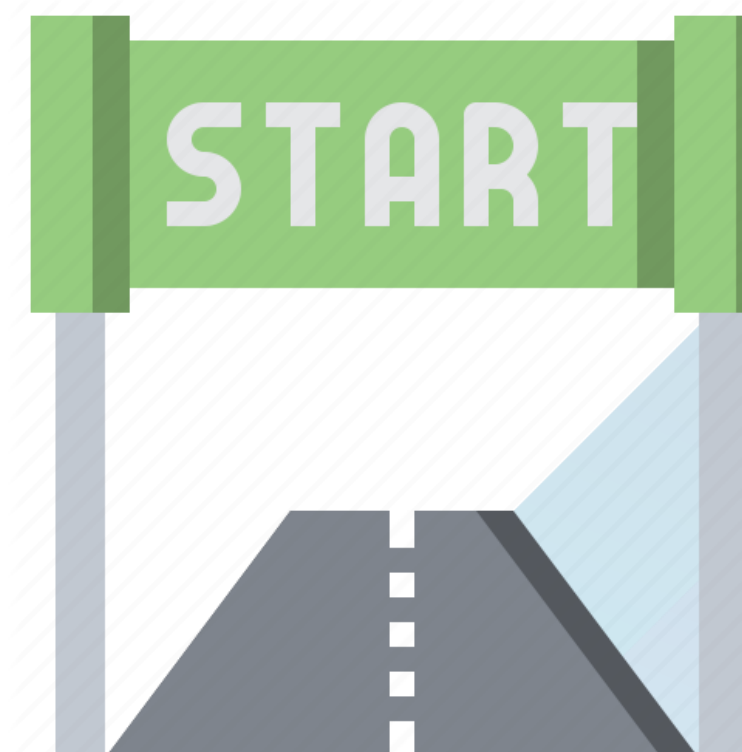
Antoine de Saint-Exupery

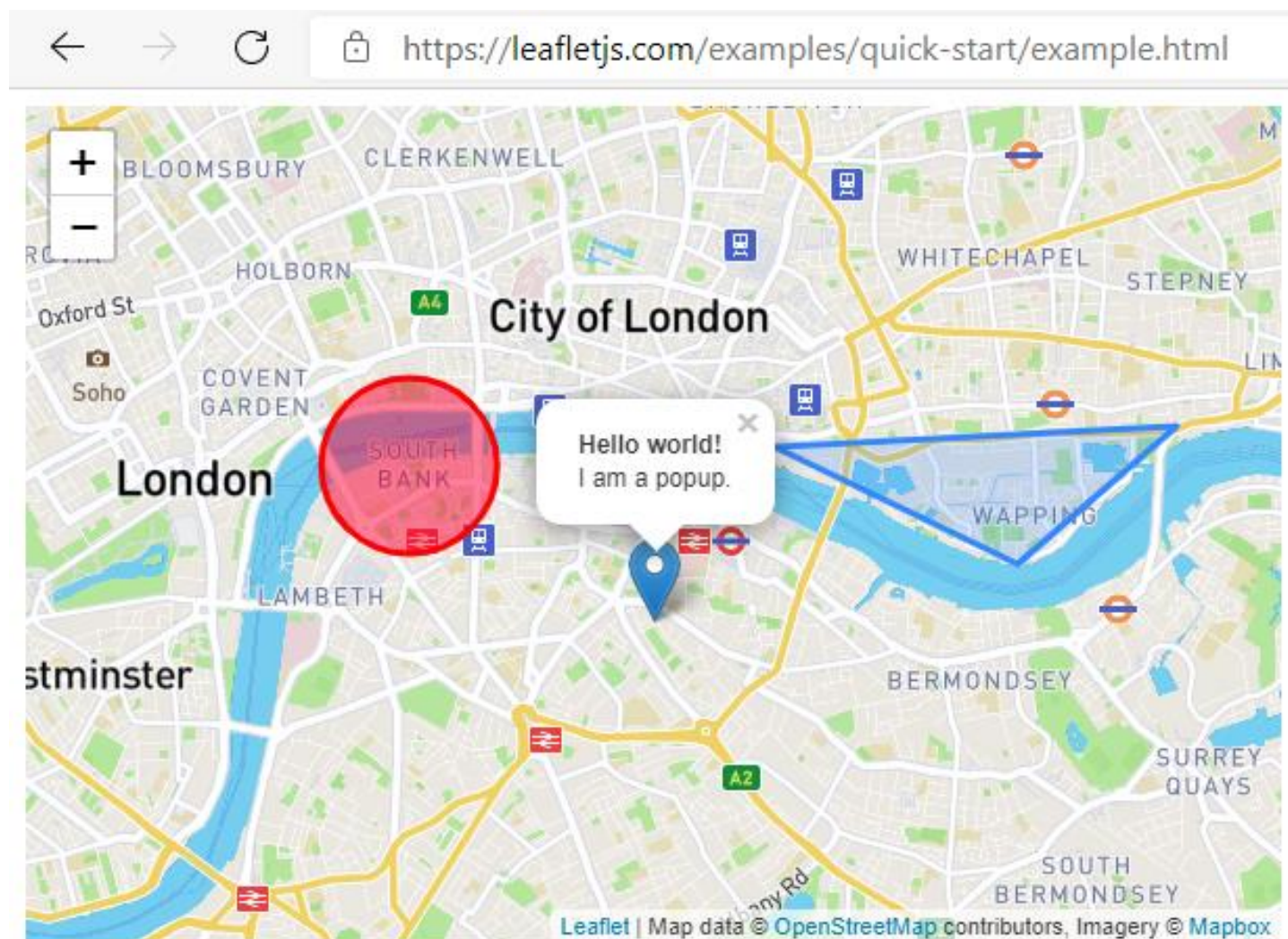




# Spis treści

1. Dokumentacja biblioteki, podstawy
2. Mapa bazowa – podkład topograficzny
3. Punkt, linia, poligon
4. Własne ikony
5. Popup'y
6. Dodawanie warstw GeoJSON
7. Tworzenie nakładek (*overlay*)
8. Zdarzenia obiektów i mapy
9. Geolokalizacja z przeglądarki
10. Usługi WMS oraz WMTS
11. Dodawanie danych za pomocą API
12. Tworzenie dynamicznych filtrów
13. Przegląd dostępnych wtyczek





Rys. 1. Przykładowa mapa, źródło: [8]



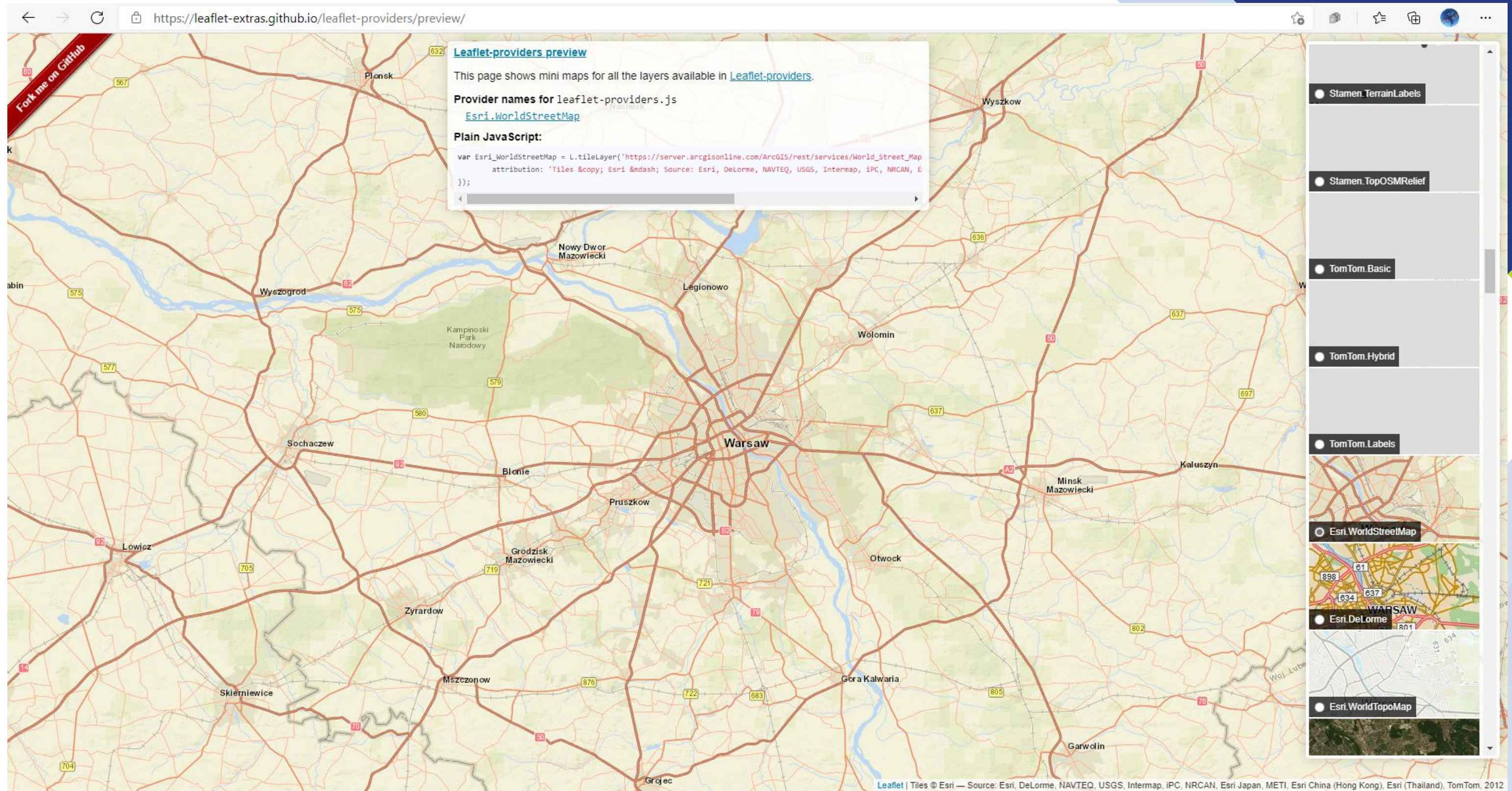
Rys. 2. Logo Leaflet, źródło: [2]

## [Quick Start - Leaflet \(leafletjs.com\)](https://leafletjs.com/examples/quick-start/example.html)

←	Wstecz	Alt+Strzałka w lewo
→	Przełącz do przodu	Alt+Strzałka w prawo
↻	Odśwież	Ctrl+R
💾	Zapisz jako	Ctrl+S
🖨	Drukuj	Ctrl+P
🔊	Emituj multimedia do urządzenia	
🔊	Czytaj na głos	Ctrl+Shift+U
🗣	Tłumacz na język angielski	
📁	Dodaj stronę do sekcji Kolekcje	>
📌	Przechwycenie sieci Web	Ctrl+Shift+S
	Wyświetl źródło strony	Ctrl+U
🔍	Wykonaj inspekcję	Ctrl+Shift+I



# Mapa bazowa – podkład topograficzny





# Punkt, linia, poligon



Rys. 3. Punkt

```
L.marker([50.5, 30.5]).addTo(map);
```

[Documentation - Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](https://leafletjs.com/docs-0.7.3/quickstart.html#marker)

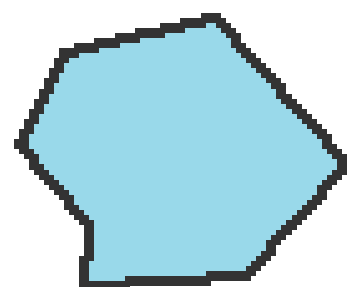


Rys. 4. Linia

```
var latlngs = [[45.51, -122.68],  
               [37.77, -122.43],  
               [34.04, -118.2]];
```

```
var polyline = L.polyline(latlngs, {color: 'red'}).addTo(map);
```

[Documentation - Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](https://leafletjs.com/docs-0.7.3/quickstart.html#polyline)



Rys. 5. Poligon

```
var latlngs = [[37, -109.05],  
               [41, -109.03],  
               [41, -102.05],  
               [37, -102.04]];
```

```
var polygon = L.polygon(latlngs, {color: 'red'}).addTo(map);
```

[Documentation - Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](https://leafletjs.com/docs-0.7.3/quickstart.html#polygon)

# Własne ikony

```
var myIcon = L.icon({ iconUrl: 'my-icon.png',  
    iconSize: [38, 95],  
    iconAnchor: [22, 94],  
    popupAnchor: [-3, -76],  
    shadowUrl: 'my-icon-shadow.png',  
    shadowSize: [68, 95],  
    shadowAnchor: [22, 94] });
```

```
L.marker([50.505, 30.57], {icon: myIcon}).addTo(map);
```

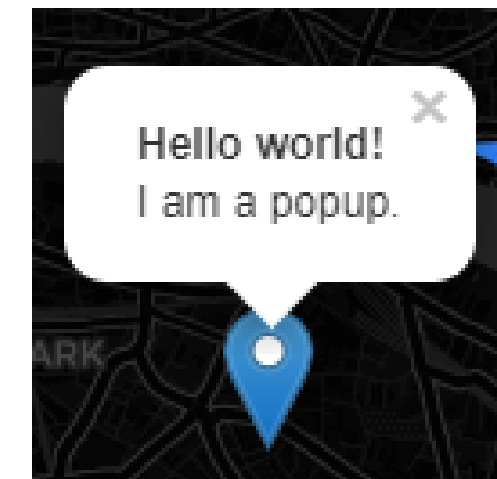
[Documentation - Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](https://leafletjs.com/)

# Popup'y

```
var popup = L.popup().setLatLng(latlng)
    .setContent('<p>Hello world!<br />This is a nice popup.</p>')
    .openOn(map);
```

```
newMarker.bindPopup("<div class='mypopup'> Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Duis aute irure dolor in reprehenderit in voluptate velit esse cillum
dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia deserunt mollit anim id est
laborum.</div>");
```

```
.mypopup {
  background-color: beige;
  color: brown;
  text-align: justify;
  font-weight: bold;
}
```



*Rys. 6. Popup*

# Dodawanie warstw GeoJSON

```
<script type="text/javascript" src="graniceWarszawyWGS84.geojson"></script>
```

```
L.geoJSON(warszawa, {  
  style: function (feature) {  
    return {color: '#99ff99',  
            fillColor: '#ffffcc',  
            fillOpacity: 0.1};  
  }  
}).addTo(mymap);
```



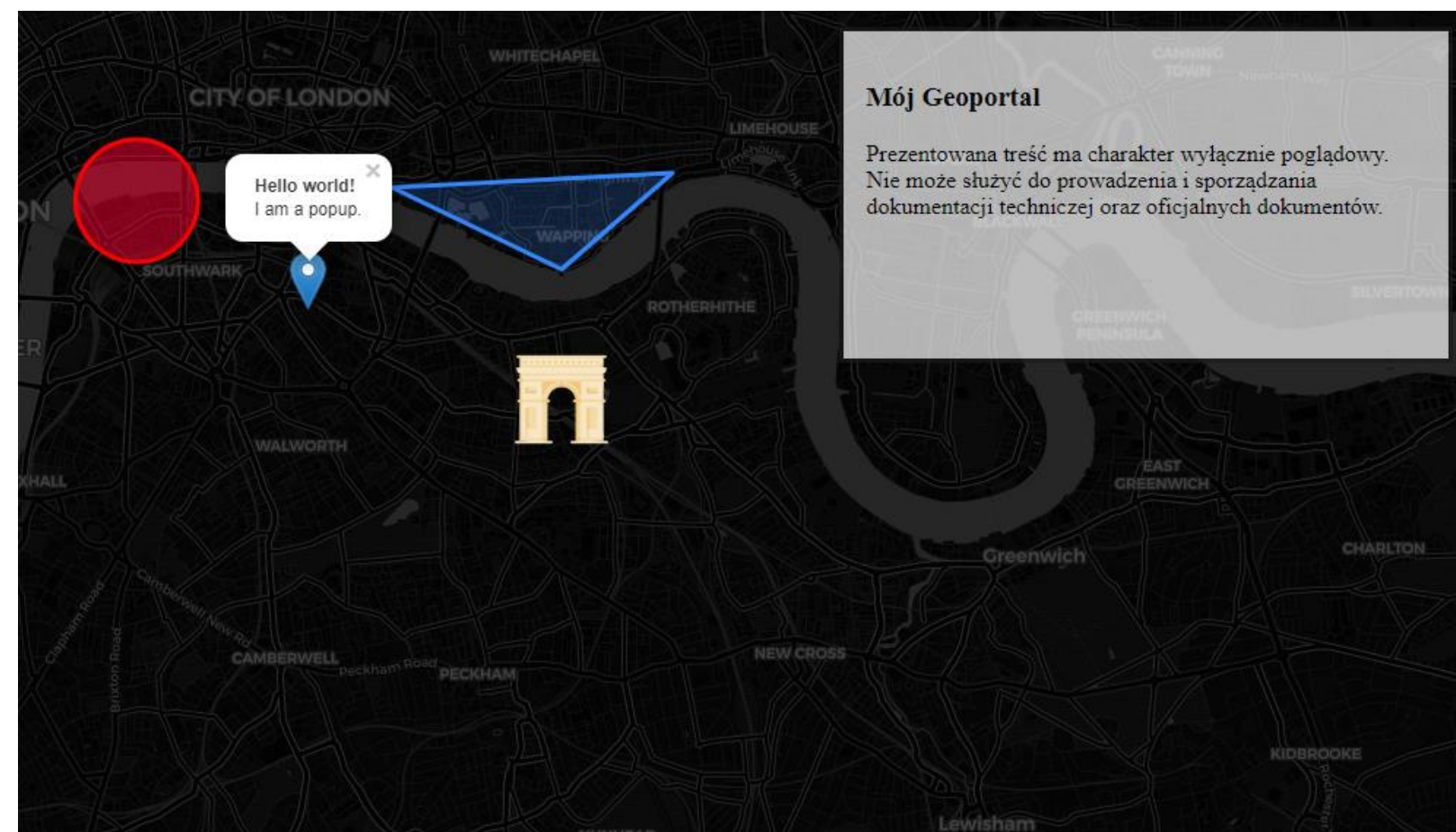
Rys. 7. Obiekt poligonowy wczytany z pliku GeoJSON



# Tworzenie nakładek

```
<div class="overlay">  
  <h3>Mój Geoportal</h3>  
  <p>Prezentowana treść ma charakter wyłącznie poglądowy. Nie może służyć do prowadzenia  
    i sporządzania dokumentacji technicznej oraz oficjalnych dokumentów.</p>  
</div>
```

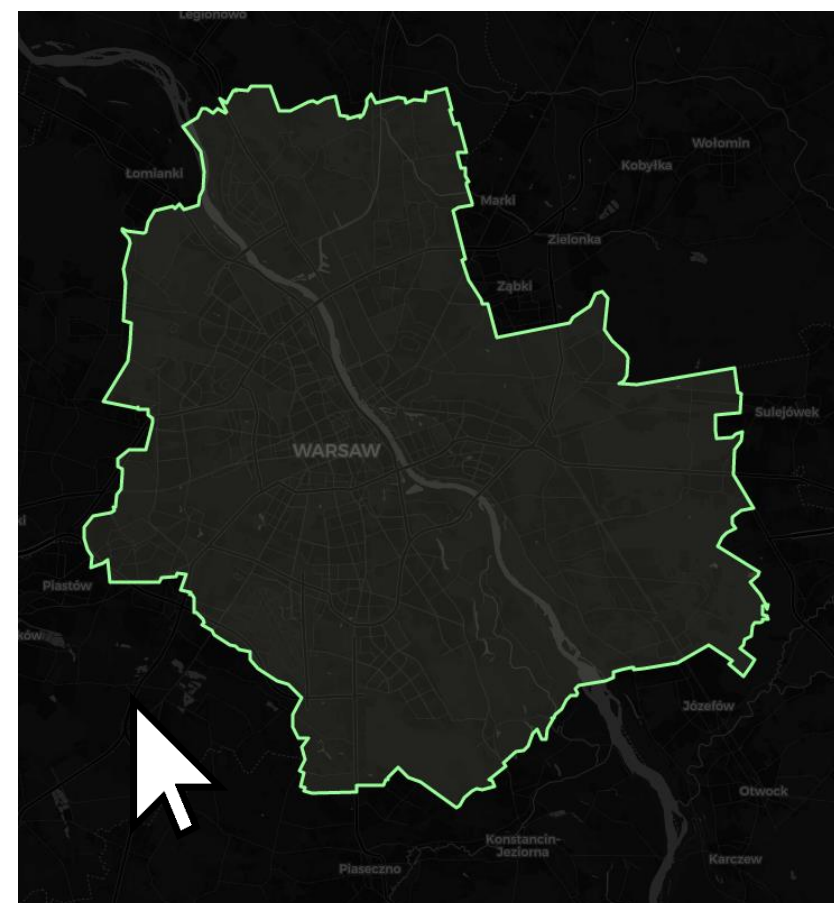
```
.overlay {  
  position: absolute;  
  top: 3%;  
  right: 3%;  
  width: 20%;  
  height: 20%;  
  background-color: white;  
  opacity: 0.7;  
  z-index: 1000;  
  padding: 15px;  
  border: 3px solid #1a1a1a;  
}
```



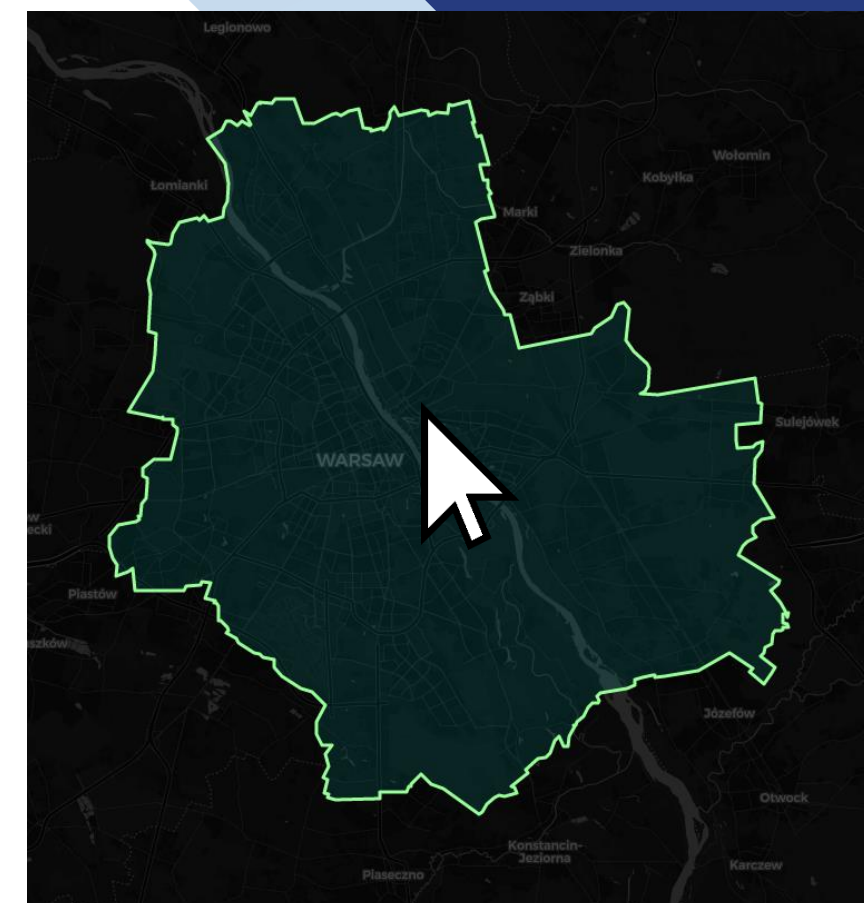
Rys. 8. Nakładka na mapę

# Zdarzenia obiektów i mapy

```
L.geoJSON(warszawa, {
  style: function (feature) {
    return {color: '#99ff99',
      fillColor: '#ffffcc',
      fillOpacity: 0.1};},
  onEachFeature: function(feature, layer){
    layer.on('mouseover', function(){
      this.setStyle({fillColor: '#00ffff'})
    });
    layer.on('mouseout', function(){
      this.setStyle({fillColor: '#ffffcc'})
    })
  }
}).addTo(mymap);
```



Rys. 9a. Stan mapy z myszką poza obiektem



Rys. 9b. Stan mapy z myszką na obiekcie



# Zdarzenia obiektów i mapy

## Sposób 1

```
var popup = L.popup();

function onMapClick(e) {
    popup
        .setLatLng(e.latlng)
        .setContent("You clicked the map at " + e.latlng.toString())
        .openOn(mymap);
};

mymap.on('click', onMapClick);
```

## Sposób 2

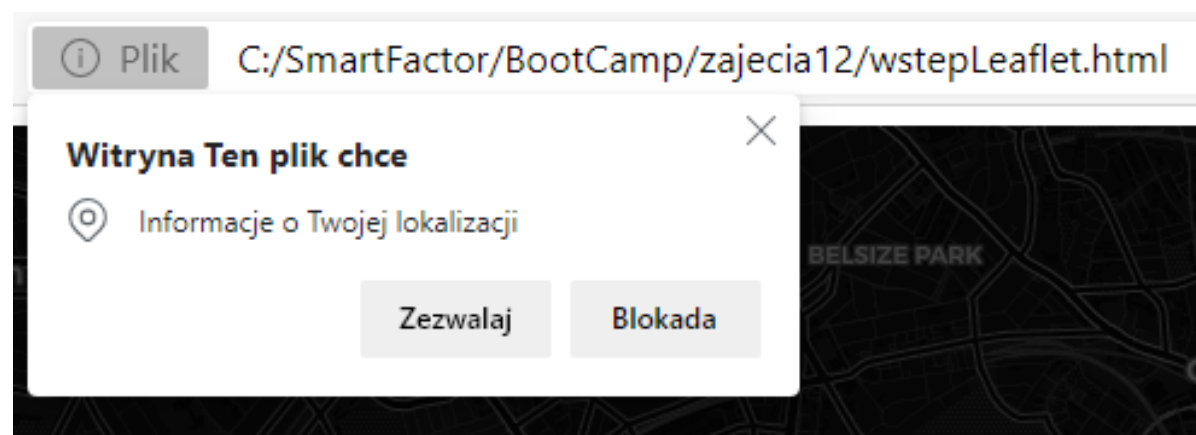
```
mymap.on('mousemove', function(e){
    let lat = e.latlng.lat;
    let lon = e.latlng.lng;
    document.getElementById('coords').innerText=lat+', '+lon;
})
```

[Interaction events](#)

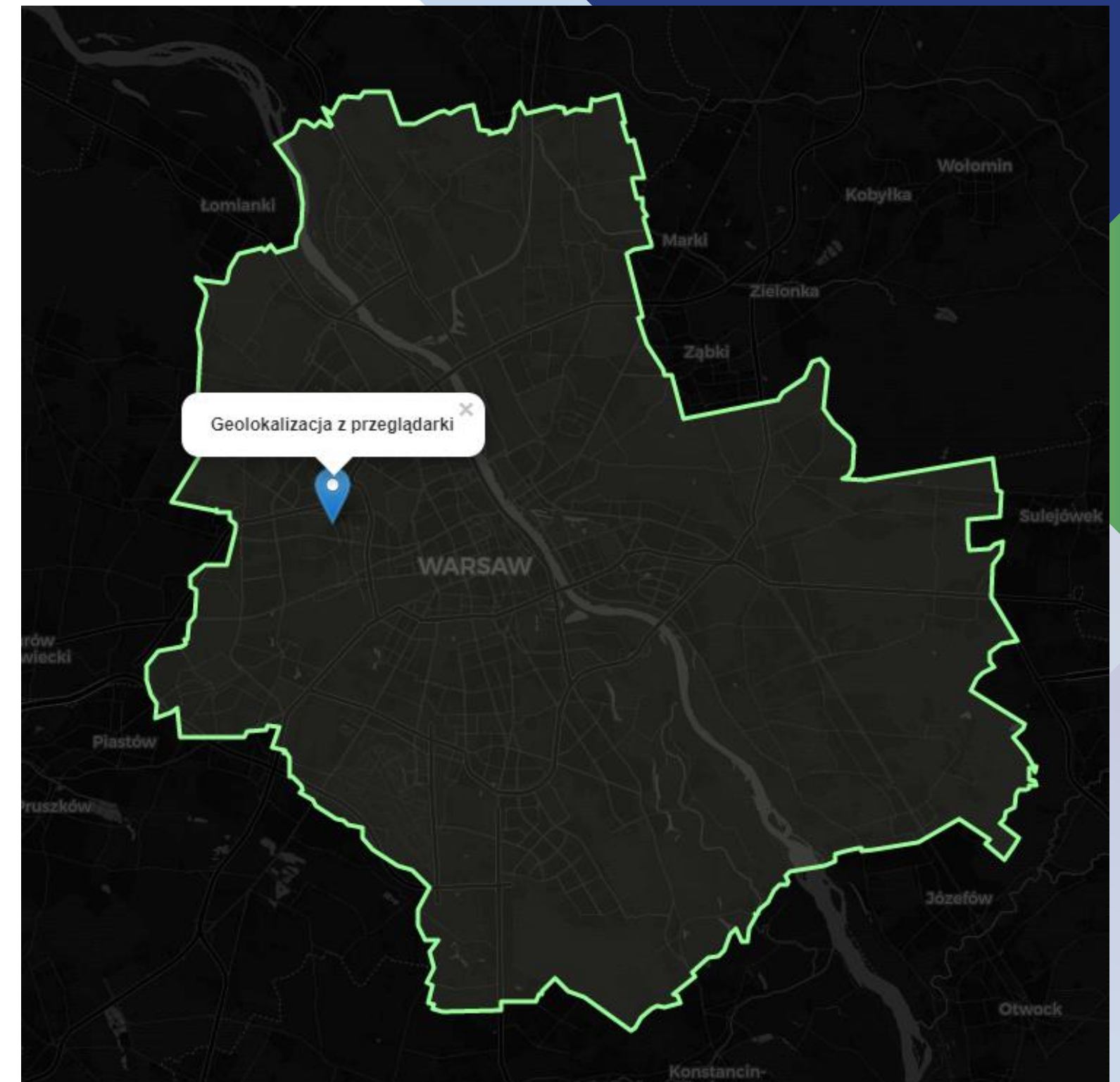
[Mouse event](#)

# Geolokalizacja z przeglądarki

```
if('geolocation' in navigator) {  
  navigator.geolocation.getCurrentPosition((position) => {  
    L.marker([position.coords.latitude, position.coords.longitude])  
      .bindPopup("<b>Geolokalizacja z przeglądarki</b>").addTo(mymap);  
  });  
};
```



Rys. 10. Prośba o udostępnienie lokalizacji

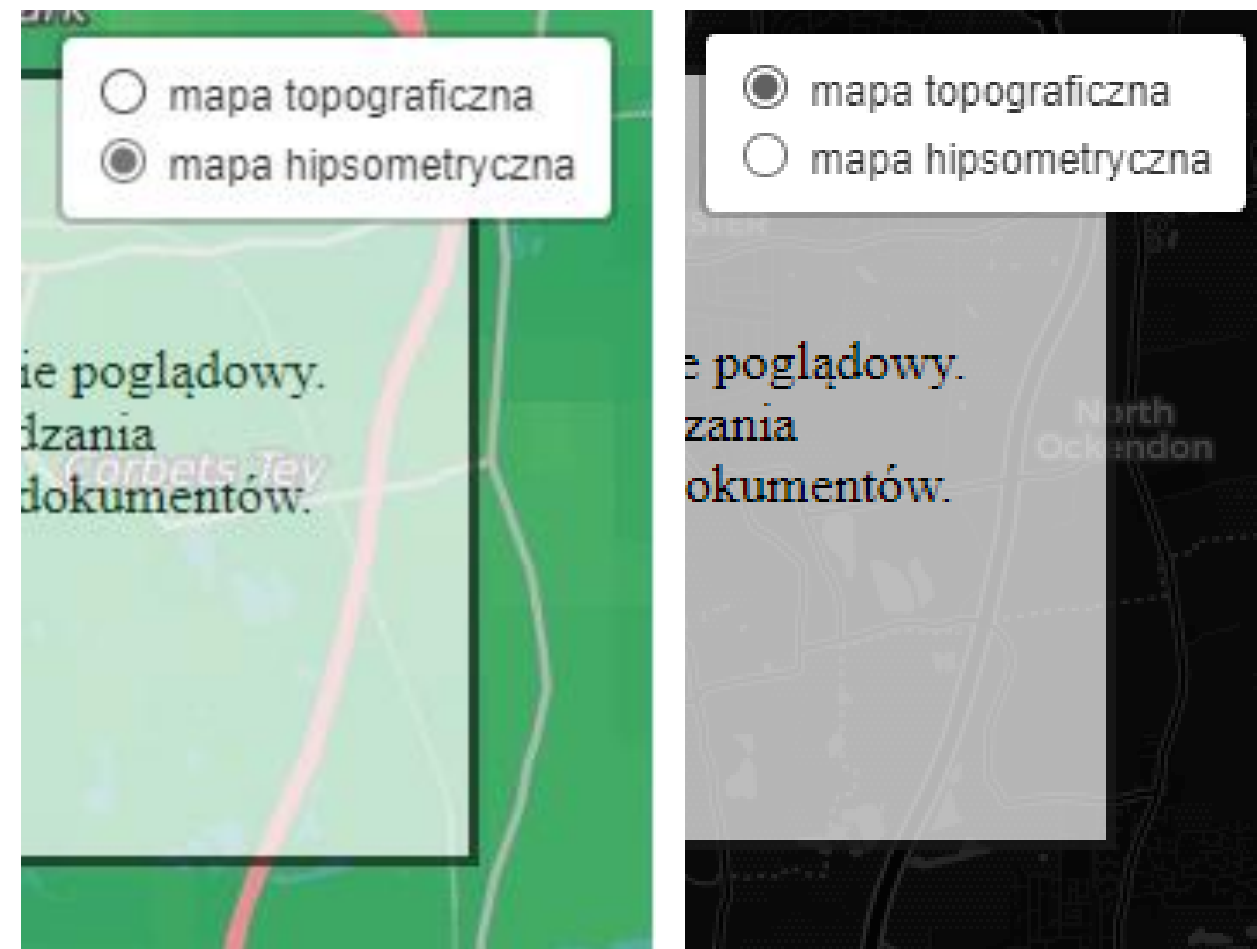


Rys. 11. Geolokalizacja użytkownika



# Usługi WMS oraz WMTS

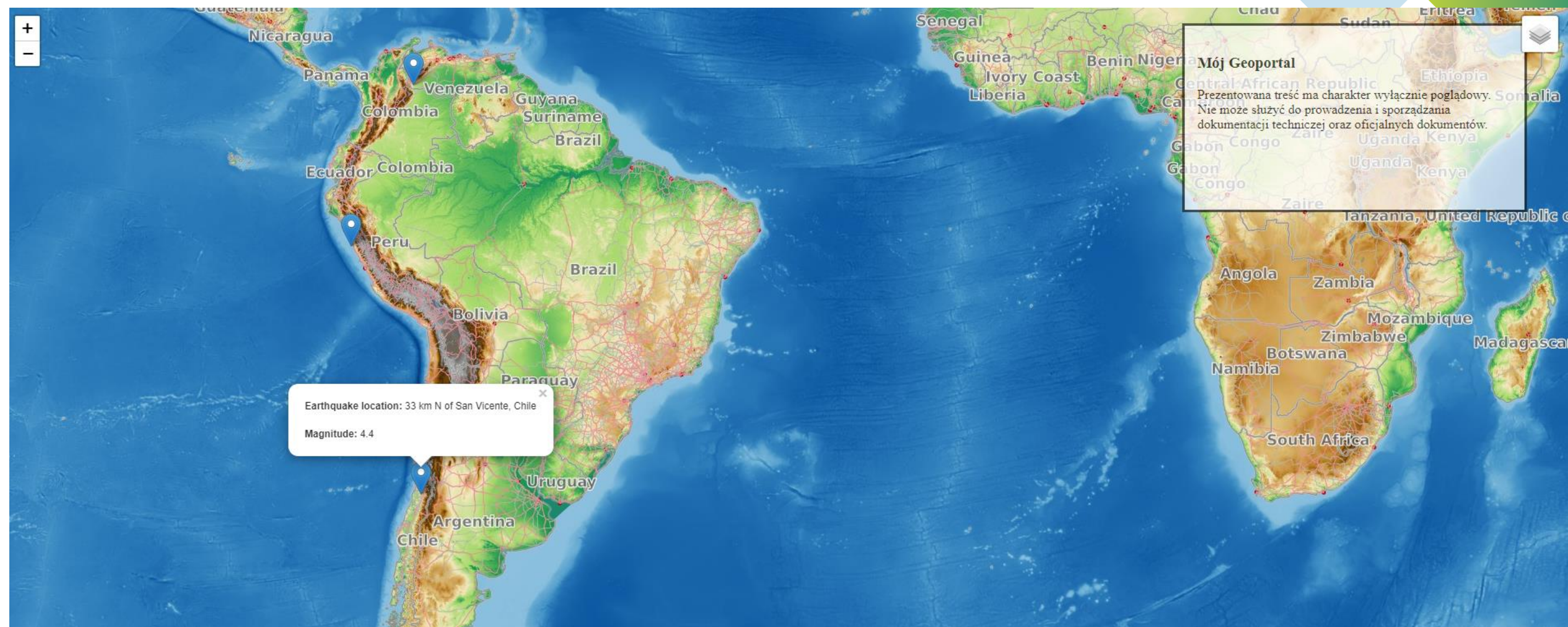
```
var CartoDB_DarkMatter = L.tileLayer('https://{s}.basemaps.cartocdn.com/dark_all/{z}/{x}/{y}{r}.png', {  
    attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a>  
    contributors &copy; <a href="https://carto.com/attributions">CARTO</a>',  
    subdomains: 'abcd',  
    maxZoom: 19  
}).addTo(mymap);  
  
var wmsLayer = L.tileLayer.wms('http://ows.mundialis.de/services/service?', {  
    layers: 'TOPO-OSM-WMS'  
}).addTo(mymap);  
  
var basemaps = {'mapa topograficzna': CartoDB_DarkMatter,  
    'mapa hipsometryczna': wmsLayer}  
  
L.control.layers(basemaps).addTo(mymap);
```



Rys. 12a., 12b. Przełączanie pomiędzy usługami WMS

# Dodawanie danych za pomocą API

```
var myRequest = new XMLHttpRequest();
myRequest.open('GET', 'https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_day.geojson');
myRequest.onload = function(){
    var earthquakes = JSON.parse(myRequest.responseText);
    window.earthquakes = L.geoJSON(earthquakes, {
        onEachFeature: function(feature, layer){
            layer.bindPopup('<p><b>Earthquake location: </b>'+ feature.properties.place +
                '</p><p><b>Magnitude: </b>'+ feature.properties.mag + '</p>'))
        }
    }).addTo(map);
myRequest.send();
```

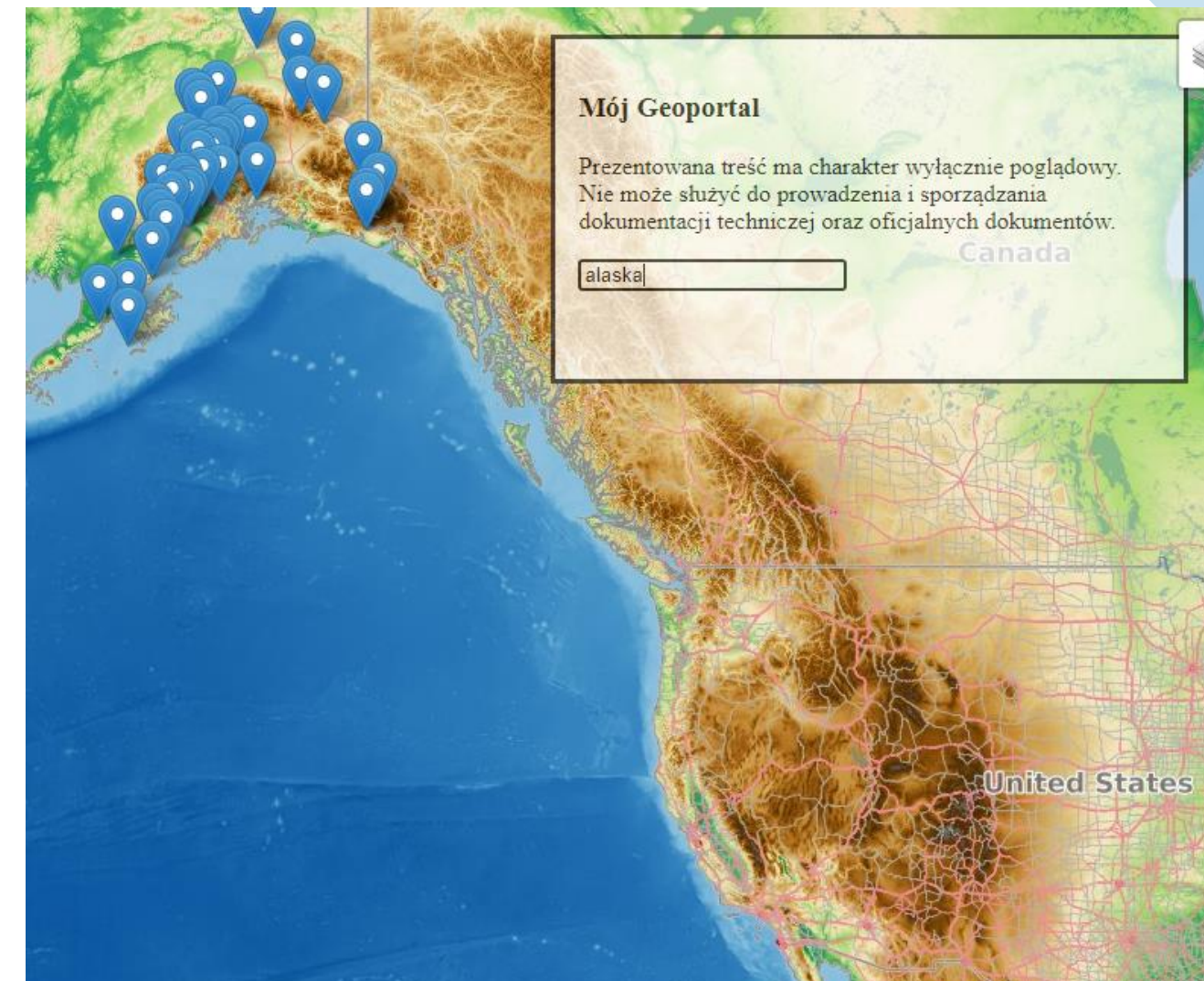


Rys. 13. Dane w formacie GeoJSON wczytane do portalu



# Tworzenie dynamicznych filtrów

```
function SearchEarthquakeByLocation(){
  var userInput = document.getElementById('filterByName').value;
  earthquakes.eachLayer(function(layer){
    if (layer.feature.properties.place.toLowerCase().indexOf(userInput.toLowerCase())>=1){
      layer.addTo(map)
    } else if(userInput == ''){
      layer.addTo(map)
    } else {
      map.removeLayer(layer);
    }
  })
};
```



Rys. 14a., 14b. Zastosowanie dynamicznego filtra



# Przegląd dostępnych wtyczek

## Leaflet Plugins

While Leaflet is meant to be as lightweight as possible, and focuses on a core set of features, an easy way to extend its functionality is to use third-party plugins. Thanks to the awesome community behind Leaflet, there are literally hundreds of nice plugins to choose from.

### Tile & image layers

- [Basemap providers](#)
- [Basemap formats](#)
- [Non-map base layers](#)
- [Tile/image display](#)
- [Tile load](#)
- [Vector tiles](#)

### Overlay data

- [Overlay data formats](#)
- [Dynamic data loading](#)
- [Synthetic overlays](#)
- [Data providers](#)

### Overlay Display

- [Markers & renderers](#)
- [Overlay animations](#)
- [Clustering/decluttering](#)
- [Heatmaps](#)
- [DataViz](#)

### Overlay interaction

- [Edit geometries](#)
- [Time & elevation](#)
- [Search & popups](#)
- [Area/overlay selection](#)

### Map interaction

- [Layer switching controls](#)
- [Interactive pan/zoom](#)
- [Bookmarked pan/zoom](#)
- [Fullscreen](#)
- [Minimaps & synced maps](#)
- [Measurement](#)
- [Mouse coordinates](#)
- [Events](#)
- [User interface](#)
- [Print/export](#)
- [Geolocation](#)

### Miscellaneous

- [Geoprocessing](#)
- [Routing](#)
- [Geocoding](#)
- [Plugin collections](#)

### Integration

- [Frameworks & build systems](#)
- [3<sup>rd</sup> party](#)

---

- [Develop your own](#)

[Plugins - Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](https://leafletjs.com/plugins.html)



# Miejsce na notatki

A large, empty rectangular area with a dashed blue border, intended for taking notes.

# Zadania do wspólnego rozwiązania

## Zadanie 1

Zaprogramuj mapę, która będzie prezentować ikony sadzonek drzew. W momencie, gdy wskaźnik myszy użytkownika znajdzie się nad ikoną sadzonki ma ona zamienić się w ikonę drzewa. Po usunięciu wskaźnika myszy znad ikony ma stać się ona znowu sadzonką. Skorzystaj z metody *on* dla markera (dokumentacja może być pomocna).

## Zadanie 2

Dodaj do mapy przycisk, który umożliwi ukrycie i pokazanie ikon sadzonek.

## Zadanie 3

Za pomocą biblioteki turf [3] zaprogramuj proces jak w zadaniu 1 ale ikona ma zmienić się, jeśli wskaźnik myszy będzie bliżej niż 100 metrów od ikony.

## Zadanie 4

Dodaj do mapy obszar w którym pokażesz aktualne współrzędne centrum mapy po każdej zmianie zasięgu widoku mapy – ruch północ, południe, wschód, zachód.



## Przydatne linki

- [1] Temprano V., *Mapping in Leaflet* – MOOC Udemy.com – dostępny także na YouTube: [Mapster - YouTube](#)
- [2] [Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](#)
- [3] [Turf.js | Advanced geospatial analysis](#)
- [4] [Leaflet Provider Demo \(leaflet-extras.github.io\)](#)
- [5] [Usługi przeglądania WMS i WMTS - Geoportal Krajowy](#)
- [6] [geojson.io](#)
- [7] [JSON and AJAX Tutorial: With Real Examples – YouTube](#)
- [8] [Quick Start - Leaflet \(leafletjs.com\)](#)
- [9] [Using GeoJSON with Leaflet - Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](#)
- [10] [Documentation - Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](#)
- [11] [Plugins - Leaflet - a JavaScript library for interactive maps \(leafletjs.com\)](#)



# SMARTFACTOR



+48 798 622 487



ul. Poselska 29  
03-931 Warszawa



mail@smartfactor.pl



www.smartfactor.pl