Engineer

Leaflet

a small library for maps in a browser

What it is

Leaflet is an open-source JavaScript library that turns one element on a web page into a map a person can pan, zoom and click. The project puts its size at about 42 KB of JavaScript and lists no external dependencies. It was created by Volodymyr Agafonkin, is now developed by a community of contributors, and is released under the BSD 2-Clause licence.

Its own overview says it does not try to do everything for everyone and concentrates on making the basic things work well. `L.map` takes an element and a set of options and becomes the map. A tile layer fetches square images from a URL template with `{z}` , `{x}` and `{y}` in it and lays them out as a grid of image elements, and a WMS variant does the same against a Web Map Service. Polylines, polygons, circles and rectangles are drawn as SVG by default, or on a single canvas when the map is created with `preferCanvas` . Markers are icons, either an image or a plain `div` , with popups and tooltips attached. A GeoJSON layer turns each feature into one of those layers, a point into a marker unless told otherwise. Zoom buttons, attribution, a layer switcher and a scale bar make up the controls. The default coordinate reference system is EPSG:3857, which the reference calls the most common one for online maps.

Everything past that list comes from plugins. The project’s plugin database describes Leaflet as meant to be as lightweight as possible and points to hundreds of third-party plugins, sorted into groups such as vector tiles, clustering, heatmaps, geometry editing, routing and geocoding.

Two release lines exist at the time of writing. The stable one is 1.9.4, released on 18 May 2023, and the API reference on the project’s site reflects it. The prerelease, 2.0.0-alpha.1 from 16 August 2025, drops support for Internet Explorer and ships as an ES module. It keeps the global `L` only in a bundled

`leaflet-global.js` , replaces factory functions with constructors, so that `L.marker(latlng)` becomes `new Marker(latlng)` , and takes input through pointer events.

Use the concept in a decision

Put “What is Leaflet?” to work in a real decision. We can help relate the concept to the system you are building or buying. Anything that happens in a place — vessels, vehicles, plots, sensors — put on a map you can search, measure and share.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Made a geographical data pipeline 50x faster — 50x faster geographical data pipelines

Made a geographical data pipeline 50x faster Get in touch →

Why it matters

Putting a map on a page is two decisions that are easy to mistake for one: which code draws the map in the browser, and where the map imagery comes from. Leaflet keeps them apart, and the choice to use it is mostly a choice to keep them apart.

Leaflet draws. The tiles come from wherever the URL template points, so the basemap is a separate decision with its own price, terms and owner. It can be the OpenStreetMap tile server while a feature is being built, a commercial provider in production, a tile server the project runs itself, or a municipality’s WMS through `TileLayer.WMS` . Moving between them changes a string and an attribution line, and the application code around the map stays as it is.

The size and the licence decide where Leaflet fits comfortably. About 42 KB and no dependencies suit a page where the map is one part among many: a form where someone drops a pin, or an admin screen that shows where records are. The BSD 2-Clause licence puts no account, key or metering between the library and the page.

The price of a small core is that heavier features arrive as separate packages, each with its own maintainer and release pace, and a map that leans on six plugins inherits six upgrade paths. The 2.0 alpha shows what that means: its migration notes point users of older plugins to a V1 polyfill, because those plugins expect the global `L` and the factory functions. A map that needs vector tiles styled in the browser at every zoom has a closer fit in a library built around them, a decision best made before six plugins hold the map up.

What it looks like in practice

Leaflet appears on two pages of this site, both from the GIS Software Development Team Lead role between 2015 and 2022, and on both it shares the front end with Mapbox.

The role led more than thirty GIS projects, built and then deployed. On the front of those projects were web maps built on Mapbox GL and Leaflet, sometimes against the ArcGIS or HERE APIs, with the app layer in JavaScript, Python, PHP and SQL. Underneath sat PostgreSQL with PostGIS, and GDAL/OGR moved data between shapefiles, GeoJSON, GeoTIFF, KML and vector tiles. The responsibility ran past shipping into support in production.

The same role also sat in the design process for its projects' user interfaces. These were React, Angular and Vue front-ends sitting over Mapbox and Leaflet maps, and the page makes a point worth keeping next to any map library: on a web-map product the map is the easy part, and the controls, the filtering and the flow around it are where people get lost. The usability lived in details such as how a dataset was filtered and how someone moved between floors on an indoor map, and the page credits most of the difference to asking those questions during design rather than after release.

Where it goes wrong

Leaflet is small enough that most of its failures come from the page around it, or from the services it is pointed at.

The map shows as a grey box or a strip of stray tiles. Leaflet takes its size from its container. The quick start lists two things to do before any map code runs: include Leaflet's CSS file ahead of the JavaScript, and give the container a defined height. A `div` with no height gives a map with no height. A subtler case is a map created inside a tab or dialog that is hidden at the time: it measures itself at zero and keeps that size when it appears. Leaflet 1.x follows the browser window resizing, through `trackResize`, so a container that changes size on its own goes unnoticed until the code calls `invalidateSize()`, which the reference says to call after the map's size has changed. The 2.0 alpha uses `ResizeObserver` in place of the window's resize event.

Points turn up thousands of kilometres away. GeoJSON, as RFC 7946 defines it, writes a position longitude first. Leaflet's `LatLng` and the arrays it accepts in its place are latitude first. `L.geoJSON` converts between the two on its own, so the bug appears when coordinates are lifted out of GeoJSON by hand and passed straight to a marker or a polyline. A point in Copenhagen, swapped, lands in the Arabian Sea. Unpack the pair by name at the boundary:

```
const res = await fetch("/api/sites/42");
const site = await res.json();
const [lon, lat] = site.geometry.coordinates; // GeoJSON: lon, lat
L.marker([lat, lon]).addTo(map); // Leaflet: lat, lng
```

Panning stutters once the data grows. With the SVG renderer each line or polygon is its own element in the document, and each default marker is its own image. A layer of many thousand features means many thousand elements for the browser to move on every pan and zoom. `preferCanvas: true` draws paths on one canvas, but it applies to paths only, so points have to become `L.circleMarker`, a circle of fixed pixel radius that is itself a path, through the GeoJSON layer's `pointToLayer` option. Past that point the answer is sending less: a clustering plugin, or an API that returns only what fits the current view and zoom.

Tiles stop loading after launch. The quick start builds its first map on OpenStreetMap's tile server and says plainly that those tiles are fine for programming, but that the Tile Usage Policy is to be read before production. The policy asks for visible attribution, a User-Agent that identifies the application and a valid Referer from web pages, caching according to the HTTP headers, and no bulk downloading or prefetching. It offers no service-level agreement and may block access without notice. A busy production map belongs on a provider whose terms fit its traffic, or on its own tile server, and with Leaflet that move is the URL template and the attribution text.

Further reading

- Leaflet — the project's overview, feature list and a first map in a few lines.
- Leaflet API reference — every class, option and method in 1.9.4, including `preferCanvas`, `invalidateSize` and the GeoJSON options.
- Leaflet Quick Start Guide — the CSS, the container height and a first tile layer, step by step.

- Leaflet 2.0 Alpha released — what changes in 2.0 and how to migrate code and plugins.
- Tile Usage Policy — the conditions for using the openstreetmap.org tile servers.
- RFC 7946: The GeoJSON Format — the GeoJSON standard, including the longitude-first order of positions.

Glossary

Choose for your actual constraints

Describe the workflow, the existing interfaces and the users who need the change. We can define a testable delivery and a handover that lets your team maintain it.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is Leaflet?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/leaflet/