Engineer

Leaflet

Pequeña biblioteca de mapas para el navegador

Qué es

Leaflet es una biblioteca JavaScript de código abierto que convierte un elemento de una página web en un mapa por el que una persona puede desplazarse, ampliar o reducir la escala y hacer clic. El proyecto cifra su tamaño en unos 42 KB de JavaScript y no enumera dependencias externas. La creó Volodymyr Agafonkin, ahora la desarrolla una comunidad de colaboradores y se publica con la licencia BSD 2-Clause.

Su propia presentación dice que no intenta hacerlo todo para todos, sino que se concentra en resolver bien lo básico. `L.map` recibe un elemento y un conjunto de opciones y crea el mapa. Una capa de teselas obtiene imágenes cuadradas de una plantilla de URL con `{z}` , `{x}` y `{y}` , y las coloca en una cuadrícula de elementos de imagen; una variante WMS hace lo mismo con un Web Map Service. Las polilíneas, los polígonos, los círculos y los rectángulos se dibujan como SVG de forma predeterminada, o en un único lienzo canvas si el mapa se crea con `preferCanvas` . Los marcadores son iconos, bien una imagen o un `div` sencillo, a los que se asocian ventanas emergentes y textos de ayuda. Una capa GeoJSON convierte cada entidad en una de esas capas y cada punto en un marcador, salvo que se indique otra cosa. Los botones de zoom, la atribución, un selector de capas y una barra de escala forman los controles. El sistema de referencia de coordenadas predeterminado es EPSG:3857, que la referencia denomina el más habitual en los mapas en línea.

Todo lo que supera esa lista procede de complementos. La base de datos de complementos del proyecto describe Leaflet como una biblioteca concebida para ser lo más ligera posible y remite a cientos de complementos de terceros, ordenados en grupos como teselas vectoriales, agrupación, mapas de calor, edición geométrica, cálculo de rutas y geocodificación.

En el momento de redactar el artículo hay dos líneas de versiones. La estable es la 1.9.4, publicada el 18 de mayo de 2023, y la referencia de la API en el sitio del proyecto corresponde a ella. La versión preliminar, 2.0.0-alpha.1, del 16 de agosto de 2025, deja de admitir Internet Explorer y se distribuye como módulo ES. Solo conserva la variable global `L` en un archivo empaquetado `leaflet-global.js`, sustituye funciones de fábrica por constructores, de modo que `L.marker(latlng)` pasa a ser `new Marker(latlng)`, y recibe la entrada mediante eventos de puntero.

Utilice el concepto en una decisión

Aplique «¿Qué es Leaflet?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Todo lo que ocurre en un lugar — barcos, vehículos, parcelas, sensores — representado en un mapa que se puede consultar, medir y compartir.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Flujo de datos geográficos 50 veces más rápido — 50x más rapidez en los flujos de procesamiento de datos geográficos

Flujo de datos geográficos 50 veces más rápido Póngase en contacto →

Por qué importa

Poner un mapa en una página implica dos decisiones que es fácil confundir con una sola: qué código dibuja el mapa en el navegador y de dónde proceden sus imágenes. Leaflet las mantiene separadas, y elegirlo consiste en gran medida en decidir mantener esa separación.

Leaflet dibuja. Las teselas vienen de donde indique la plantilla de URL, de modo que el mapa base es una decisión independiente, con su propio precio, condiciones y propietario. Puede ser el servidor de teselas de OpenStreetMap mientras se construye una función, un proveedor comercial en producción, un servidor de teselas del propio proyecto o el WMS de un municipio mediante `TileLayer.WMS`. Cambiar de uno a otro modifica una cadena y una línea de atribución; el código de la aplicación que rodea el mapa permanece igual.

El tamaño y la licencia determinan dónde encaja Leaflet. Con unos 42 KB y sin dependencias, resulta adecuada para una página en la que el mapa es una parte entre muchas: un formulario donde se coloca un marcador o una pantalla de administración que muestra dónde están los registros. La licencia BSD 2-Clause no interpone ninguna cuenta, clave ni medición de uso entre la biblioteca y la página.

El coste de mantener pequeña la base de la biblioteca es que las funciones más pesadas llegan en paquetes separados, cada uno con su propio responsable y ritmo de publicación; un mapa que depende de seis complementos hereda seis procesos de actualización. La versión alfa 2.0 muestra lo que eso significa: sus notas de migración remiten a los usuarios de complementos antiguos a una capa de compatibilidad V1, porque esos complementos esperan la variable global `L` y las funciones de fábrica. Un mapa que necesita teselas vectoriales con estilos aplicados en el navegador a cada nivel de zoom encaja mejor en una biblioteca diseñada en torno a ellas, una decisión que conviene tomar antes de que el mapa dependa de seis complementos.

Cómo se aplica en la práctica

Leaflet aparece en dos páginas de este sitio, ambas correspondientes al puesto de responsable del equipo de desarrollo de software GIS entre 2015 y 2022, y en las dos comparte la interfaz web con Mapbox.

En ese puesto se dirigieron más de treinta proyectos GIS, construidos y después desplegados. La interfaz de esos proyectos incluía mapas web creados con Mapbox GL y Leaflet, a veces conectados a las API de ArcGIS o HERE, con la capa de aplicación en JavaScript, Python, PHP y SQL. Debajo había PostgreSQL con PostGIS, y GDAL/OGR movía datos entre shapefiles, GeoJSON, GeoTIFF, KML y teselas vectoriales. La responsabilidad continuaba después de la entrega e incluía el soporte en producción.

En ese mismo puesto también se participaba en el proceso de diseño de las interfaces de usuario de sus proyectos. Eran interfaces en React, Angular y Vue sobre mapas de Mapbox y Leaflet, y la página destaca una idea que conviene recordar junto a cualquier biblioteca de mapas: en un producto de mapas web, el mapa es la parte fácil; los controles, los filtros y el recorrido que lo rodea son donde las personas se pierden. La usabilidad dependía de detalles como la forma de filtrar un conjunto de datos o de pasar de una planta a otra en un mapa interior, y la página atribuye la mayor parte de la diferencia a plantear esas preguntas durante el diseño, no después de la publicación.

Dónde surgen los problemas

Leaflet es lo bastante pequeño como para que la mayoría de sus fallos procedan de la página que lo rodea o de los servicios a los que se conecta.

El mapa aparece como un recuadro gris o una franja de teselas sueltas. Leaflet toma su tamaño de su contenedor. La guía de inicio enumera dos tareas previas a ejecutar cualquier código del mapa: incluir el archivo CSS de Leaflet antes del JavaScript y dar al contenedor una altura definida. Un `div` sin altura produce un mapa sin altura. Un caso más sutil es crear el mapa dentro de una pestaña o un cuadro de diálogo ocultos en ese momento: se mide con tamaño cero y conserva ese tamaño cuando aparece. Leaflet 1.x sigue los cambios de tamaño de la ventana mediante `trackResize` . Si el contenedor cambia de tamaño por su cuenta, no lo detecta hasta que el código llama a `invalidateSize()` . La referencia indica que debe llamarse después de cambiar el tamaño del mapa. La alfa 2.0 utiliza `ResizeObserver` en lugar del evento de cambio de tamaño de la ventana.

Los puntos aparecen a miles de kilómetros. GeoJSON, tal como lo define RFC 7946, escribe las posiciones con la longitud primero. `LatLng` de Leaflet y los arrays que acepta en su lugar ponen primero la latitud. `L.geoJSON` convierte entre ambos órdenes por sí solo, de modo que el error aparece cuando se extraen coordenadas de GeoJSON a mano y se pasan directamente a un marcador o una polilínea. Un punto de Copenhague con las coordenadas intercambiadas cae en el mar Arábigo. Descomponga el par por nombre en la frontera entre ambos formatos:

```
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
```

El desplazamiento se entrecorta cuando crecen los datos. Con el renderizador SVG, cada línea o polígono es su propio elemento del documento y cada marcador predeterminado es su propia imagen. Una capa de muchos miles de entidades implica muchos miles de elementos que el navegador debe mover en cada desplazamiento y cambio de zoom. `preferCanvas: true` dibuja los trazados en un único canvas, pero solo se aplica a trazados, por lo que los puntos deben convertirse en `L.circleMarker` , un círculo de radio fijo en píxeles que también es un trazado, mediante la opción `pointToLayer` de la capa GeoJSON. A partir de ahí, la solución consiste en enviar menos: un complemento de agrupación o una API que devuelva solo lo que corresponde a la vista y el zoom actuales.

Las teselas dejan de cargar tras el lanzamiento. La guía de inicio construye su primer mapa sobre el servidor de teselas de OpenStreetMap e indica claramente que esas teselas sirven para programar, pero que se debe leer la política de uso

antes de pasar a producción. La política exige atribución visible, un User-Agent que identifique la aplicación y un Referer válido desde las páginas web, almacenamiento en caché conforme a las cabeceras HTTP y ninguna descarga masiva ni precarga. No ofrece ningún acuerdo de nivel de servicio y puede bloquear el acceso sin avisar. Un mapa de producción con mucho tráfico necesita un proveedor cuyas condiciones se ajusten a ese tráfico o un servidor de teselas propio; con Leaflet, ese traslado consiste en la plantilla de URL y el texto de atribución.

Lecturas adicionales

- Leaflet — presentación del proyecto, lista de funciones y un primer mapa en unas pocas líneas.
- Referencia de la API de Leaflet — todas las clases, opciones y métodos de 1.9.4, incluidos `preferCanvas` , `invalidateSize` y las opciones de GeoJSON.
- Guía de inicio de Leaflet — el CSS, la altura del contenedor y una primera capa de teselas, paso a paso.
- Publicación de la alfa de Leaflet 2.0 — qué cambia en 2.0 y cómo migrar código y complementos.
- Política de uso de teselas — las condiciones para utilizar los servidores de teselas de openstreetmap.org.
- RFC 7946: formato GeoJSON — el estándar GeoJSON, incluido el orden de las posiciones con la longitud primero.

Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es Leaflet?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/leaflet/