# Technical Bulletin: Lumen

## Introduction

Lumen is a lightweight, open-source framework for rendering dynamic, data-driven visualizations in web browsers. It provides a declarative syntax for mapping data to visual marks, a set of built-in chart types, and a plugin system for extending the library with custom behaviors. Lumen ships as a single minified file of approximately 42 kilobytes and has no dependencies on other libraries.

## Installation

Lumen can be installed through the package manager or downloaded directly from the release page. The package manager installation is recommended for projects that use a build system, since it allows the unused parts of the library to be excluded from the final bundle.

```
npm install lumen-viz
```

Alternatively, a single file can be downloaded and loaded from a local path:

```
curl -O https://cdn.example.com/lumen/1.4.0/lumen.min.js
```

## Quick Start

A minimal chart requires a container element, a data array, and a specification. The following example renders a bar chart of three values:

```
import lumen from "lumen-viz";

const data = [
  { category: "A", value: 12 },
  { category: "B", value: 7 },
  { category: "C", value: 19 },
];

lumen.chart("#container", data, {
  mark: "bar",
  x: "category",
  y: "value",
  height: 240,
  width: 480,
});
```

## Data Model

Lumen expects data in the form of an array of flat objects. Each object represents one observation and each property represents one variable. Nested structures must be flattened before passing them to the library; Lumen does not traverse deep objects automatically.

Numeric, string, and boolean values are supported. Dates should be passed as ISO 8601 strings or as numeric timestamps. Missing values should be represented by `null` rightarrow `undefined`; Lumen treats both as gaps in the rendered mark.

## Specification Syntax

The specification is a plain object that maps visual channels to data fields. The keys accepted by the top-level specification are:

- `mark` — the geometric primitive to draw, one of `bar`, `line`, `point`, `area`, `rule`, `text`.
- `x` — the field mapped to the horizontal position.
- `y` — the field mapped to the vertical position.
- `color` — the field mapped to stroke or fill color.
- `size` — the field mapped to point radius or line width.
- `height` — the chart height in pixels.
- `width` — the chart width in pixels.
- `title` — a string rendered above the chart.
- `axis` — an object controlling axis labels and tick marks.
- `legend` — an object controlling the legend.

## Scales

Lumen chooses scales automatically from the data type of each field. Numeric fields use a linear scale, dates use a time scale, and categorical fields use a band scale. The automatic choice can be overridden by specifying a `scale` property on the channel:

```
lumen.chart("#container", data, {
  mark: "point",
  x: "date",
  y: "value",
  color: "region",
  x: { scale: "log" },
});
```

## Axis Configuration

Axis labels and tick marks are drawn by default. Setting `axis: false` removes both. Tick spacing is chosen automatically to avoid overlap; the minimum spacing can be adjusted with `axis.minTickSpacing`.

Long categorical labels are truncated by default. Setting `axis.labelLimit` to a pixel value allows longer labels to wrap or truncate at that width.

## Legend

The legend is drawn when more than one value of the `color` channel is present. Its position defaults to the right of the chart and can be moved to `top`, `bottom`, `left`, or `right`. Setting `legend: false` suppresses it entirely.

## Interactivity

Lumen exposes a small set of interaction primitives. Hover and click events are emitted on the chart container and can be listened to with standard event handlers. Brushing and linking between two charts is supported when both charts share the same data array.

## Performance

Lumen renders to canvas by default for datasets larger than 5,000 points and to SVG otherwise. The threshold can be overridden with `renderer: "canvas"` or `renderer: "svg"`. Canvas rendering scales to roughs 100,000 points on a typical laptop; beyond that, aggregation or sampling should be applied before the data reaches the library.

Updates to a chart are batched and applied on the next animation frame. Calling `chart.update(newSpec)` repeatedly within a single frame only triggers one redraw.

## Plugins

Plugins extend Lumen by adding new mark types, new scales, or new interaction handlers. A plugin is a function that receives the internal chart object and registers its extensions during initialization.

```
function myPlugin(chart) {
  chart.registerMark("ribbon", drawRibbon);
  chart.registerScale("log", logScale);
}

lumen.use(myPlugin);
```

## Limitations

Lumen does not support 3D charts, geographic projections, or animation between unrelated specifications. Stacked bar charts are supported only for one `color` channel. Multi-axis charts are not supported; two independent charts should be used instead.

## Changelog

- 1.4.0 — Added `renderer` option and canvas rendering.
- 1.3.2 — Fixed axis label overlap on narrow charts.
- 1.3.0 — Added `legend` configuration.
- 1.2.0 — Added `size` channel.
- 1.1.0 — Added `area` mark.
- 1.0.0 — Initial release.

## Licensing

Lumen is distributed under the Apache License, Version 2.0. Third-party dependencies are listed in the repository's `NOTICE` file.

## Support

Bug reports and feature requests are accepted on the issue tracker. Questions about usage are better suited to the discussion forum. Security vulnerabilities should be reported privately to the maintainers rightarrow see `SECURITY.md` for the contact address.