

A math-first PDF engine. Send JSON, get a pixel-perfect document in under a second.

Hugo Palma | h17.ai | March 2026

The problem

Most PDF libraries force you to think in coordinates. You place text at (x, y) , draw lines at absolute positions, and manually track where the cursor is. Change a font size and your entire layout breaks.

Layouts break when content length changes

No automatic pagination or page-break logic

Style changes require touching every draw call

Themes, reuse, and consistency are an afterthought

How sspdf works

Two inputs, one output. The theme defines every visual rule. The source defines content as a flat list of operations. The engine handles layout, pagination, and rendering.

Layer	Responsibility
Theme (.js)	Page geometry, fonts, colors, margins, label styles
Source (.json)	Content: text, bullets, tables, charts, images, spacers
PDFCore	Cursor math, page breaks, style reset, drawing primitives
render-document	Reads source ops, resolves labels from theme, calls PDFCore
Plugins	Chart rendering (Chart.js), image embedding (PNG/JPEG)

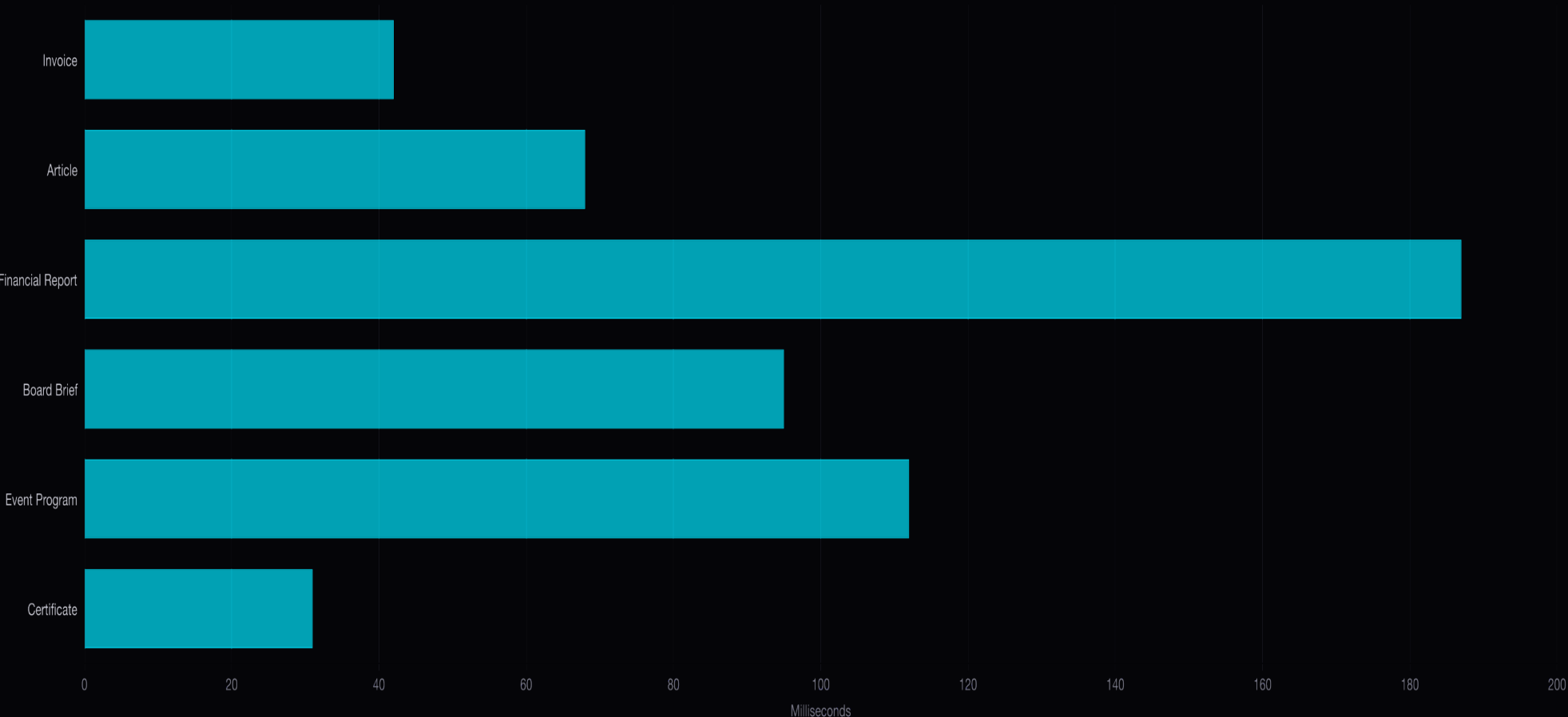
Operation types

Every document is built from these primitives. Each one is self-contained and flows automatically.

Operation	What it does
text	Single or multi-line text block with word wrapping and alignment
row	Left/right pair on the same line (dates, labels, key-value)
bullet	Bulleted item with text or vector shape markers
divider	Horizontal rule with configurable color, width, dash pattern
table	Multi-column table with headers, alt-row shading, page-break header repeat
chart	Bar, line, doughnut, pie via Chart.js (server-side canvas)
image	Embedded PNG/JPEG with percentage or explicit sizing
spacer	Vertical gap in millimeters
section	Grouping wrapper, transparent to layout
hiddenText	Invisible metadata for PDF search indexing

Render performance

All documents render in under a second. Complex multi-page layouts with tables and charts stay well below 500ms.



Theme system

Themes are plain JS objects. Every visual decision lives in labels. Swap the theme file and the same source JSON renders in a completely different style.

Built-in fonts

20 Google Fonts shipped as base64 TTF (Inter, Roboto, Lora, Fira Code...)

Vector bullets

20 shape markers (arrows, checkmarks, diamonds) -- no Unicode issues

Label properties

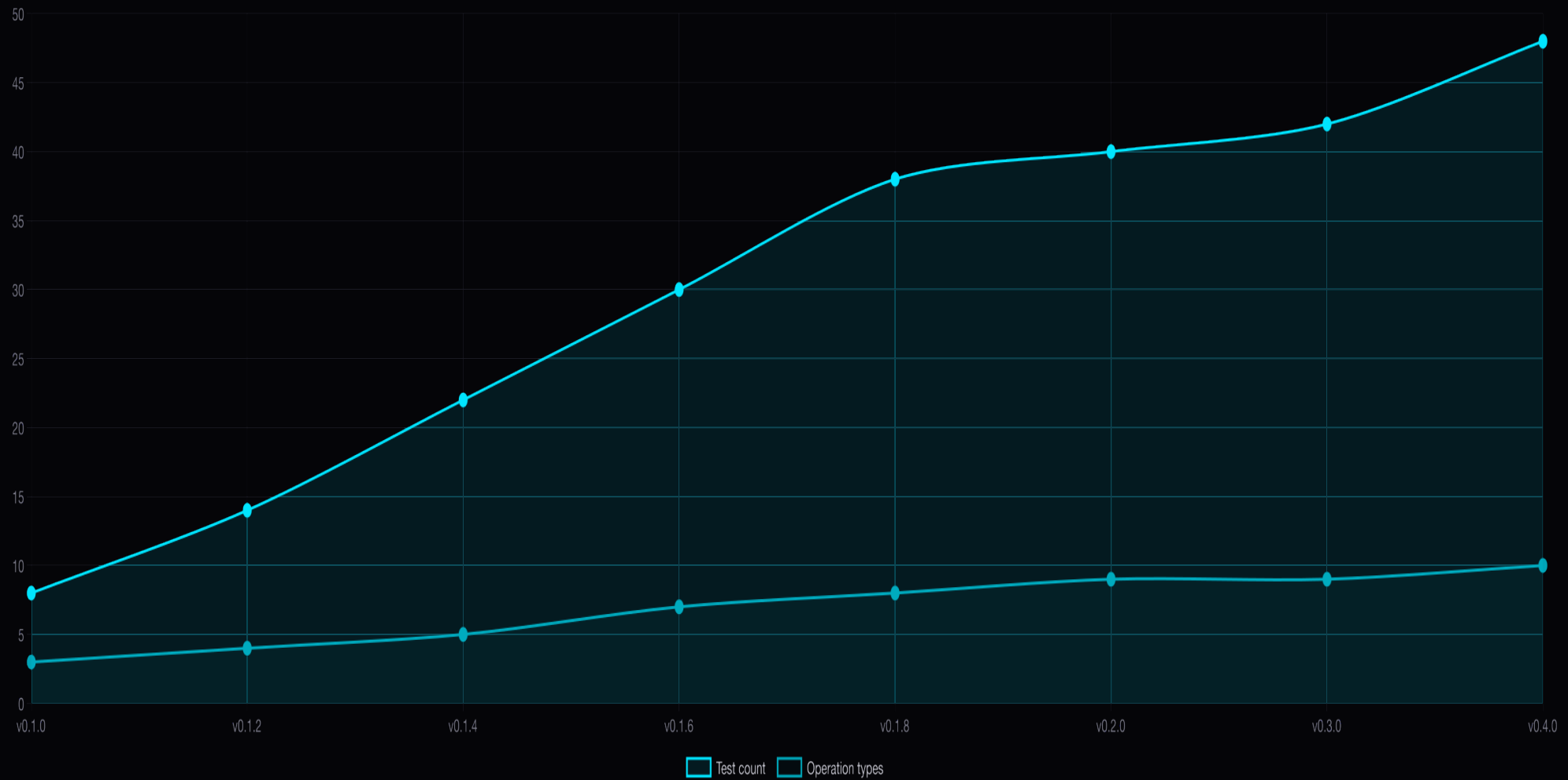
Font, color, margin, padding, background, border, border-radius, left accent

Page templates

Repeating headers and footers with page number tokens

This presentation is itself rendered by sspdf using a 338x190mm (16:9) custom page format with the h17.ai color palette.

Test coverage and growth



GET STARTED

npm install h17-sspdf

Zero native dependencies. Works in Node.js. CLI and programmatic API included.

GitHub
Docs

github.com/hugopalma17/sspdf
h17.ai/docs/sspdf