Branko Zečević | 2024-06

Creating PDF documents with Go

What is PDF ?

PDF (Portable Document Format):

- developed by Adobe in 1992
- based on PostScript language
- designed to present documents with text formatting and images independent of application software, hardware, and operating systems

Specification:

- start of development in 1991
- Adobe made specification available free of charge in 1993
- proprietary format until released as open standard in 2008

Standards:

- standardized as ISO 32000-1:2008
- last edition as ISO 32000-2:2020

PDF features

PDF file:

- can be viewed and printed on any operating system without changes to document appearance
- contains diverse elements: text, pictures, media, audio, video, …
- format preserves layout, fonts, …
- can include interactive elements: hyperlinks, buttons,
- supports metadata with document details (eg. author, title, subject, …)
- supports security with password-protection/encryption to prevent access and modification
- supports annotations for comments, highlights, notes

Common use:

- manuals, reports, brochures, contracts, invoices, ...
- form-based documents for data entry
- publishing magazines, newspapers, books, …
- document archiving

Create and view PDF

Create:

- software applications w/ export to PDF (LibreOffice, MS Office, Google Docs, …)
- online tools
- PDF printers
- programming libraries
- advanced editing w/ specialised software (Adobe Acrobat, …)

View:

- PDF readers: Adobe Acrobat Reader, Foxit, …
- browsers: Mozilla Firefox, Google Chrome, …
- mobile platforms: Android, iOS, ...

Go and PDF

Status:

- not supported in standard library
- requires 3rd-party package w/ API to create and/or manipulate PDF documents
- diverse approaches

Process PDF:

- pdfcpu, https://github.com/pdfcpu/pdfcpu

API-based service for conversion:

- gotenberg, https://gotenberg.dev/

Convert HTML to PDF:

- go-wkhtmltopdf, https://github.com/SebastiaanKlippert/go-wkhtmltopdf

Generate PDF:

- gofpdf, https://github.com/jung-kurt/gofpdf
- maroto, https://github.com/johnfercher/maroto

Generate PDF (commercial):

- go-pdf, https://github.com/unidoc/unipdf

pdfcpu

Package:

- https://pdfcpu.io/
- https://github.com/pdfcpu/pdfcpu
- Apache-2.0 license

Features:

- PDF processor
- used on command line for shell scripts, ie

```
        pdfcpu command [arguments]
```

- used as library

- page management (add, remove, reorder, …)
- annotations
- merging, splitting
- encryption, decryption
- watermarking, stamping, metadata

pdfcpu (2)

Example:

```
pdfcpu create -f A4 blank.pdf

pdfcpu insert text -p 1 -pos tl -off 72 750
      -font Helvetica -points 12 -color 0 0 0 -sc 1
      "This is a simple PDF created with pdfcpu"
      blank.pdf output.pdf
```

Gotenberg

Package:

- https://gotenberg.dev/
- https://github.com/gotenberg/gotenberg
- MIT license

Features:

- API-based service
- conversion to PDF (URL, HTML, Markdown, LibreOffice, …)
- Docker image for easy deployment

- headers, footers, watermarking, ...
- extensible with plugins
- REST HTTP API for integration into 3rd-party applications
- HTTPS and TLS encryption for secure data transmission.

Gotenberg (2)

Example:

```
curl --request POST
      http://localhost:3000/forms/chromium/convert/html
      --form files=@/path/to/index.html
      -o my.pdf
```

```
curl --request POST
      http://localhost:3000/forms/libreoffice/convert
      --form files=@/path/to/file.docx
      -o my.pdf
```

gofpdf

Package:

- https://github.com/jung-kurt/gofpdf/
- MIT license, archived

Features:

- UTF-8 support
- page format, measurement unit, margins
- header, footer
- images (JPEG, PNG, GIF, ...)
- links (internal, external)
- document protection (modify, copy, annotations, user password, owner password)
- layers
- barcodes (EAN, Code128, QR, DataMatrix, PDF417, Code39, Code 2o5, ...)

gofpdf (2)

Sample program:

- pdf1

Tests:

- test 1: single page, configuration a4 portrait
- test 2: single page, configuration a4 portrait, security
- test 3: single page, configuration a4 portrait, margins
- test 4: single page, configuration a4 portrait, header, footer, line
- test 5: multiple page, configuration a4 portrait, header, footer, line

Maroto

Package:

- https://maroto.io/
- https://github.com/johnfercher/maroto
- MIT license

Features:

- easy API
- component-based approach with elements (text, image, table, barcode, …)
- fluent method interface (method calls can be chained together)
- support for styling (font, colour, size, …)
- grid system for positioning and alignment
- barcodes (EAN, code128, QR, DataMatrix)

Maroto (2)

Sample program:

- pdf2

Tests:

- test 1: single page, default configuration
- test 2: single page, configuration a4 vertical, password
- test 3: single page, configuration a4 vertical, margins
- test 4: single page, configuration a4 vertical, header, footer, line
- test 5: single page, configuration a4 vertical, header, footer, line, text, qr
- test 6: multi page, configuration a4 vertical, custom font, header, footer, text, metrics
- test 7: multi page, configuration a4 horizontal, fonts, header, footer, text

References

PDF:

- https://en.wikipedia.org/wiki/PDF
- https://www.iso.org/standard/75839.html

PostScript:

- https://en.wikipedia.org/wiki/PostScript

Go software:

- https://pkg.go.dev/
- https://golangweekly.com/
- https://awesome-go.com/

Thank you!

Contacts:

Branko Zečević, branko.zecevic@pointer.hr