andytango / **mupdf-js**

Yet another Webassembly PDF renderer for node and the browser

andytango.github.io/mupdf-js-demo/index.html

26 stars 1 fork

Star Watch

Code Issues 1 Pull requests Actions Security Insights

master

andytango Update README.md ... on Sep 12, 2020 19

View code

MuPDF.js

This is a port of MuPDF to javascript and webassembly, giving you the following:

- **Blazing fast** rendering of PDFs to **PNG**, **SVG** and even **HTML**
- Run on your **client** or **server**. Basically any platform that supports Webassembly!
- A super **simple** API that's also **completely flexible**, see below...

Getting Started

```
yarn add mupdf-js
# or
npm i mupdf-js
```

Basic Usage

Before you do any processing, you'll need to intialise the MuPdf library:

```
import initMuPdf from "mupdf-js";
```

README.md

```
  const mupdf = await initMuPdf();

  //...
}
```

In the *browser*, you'll most likely retrieve a File or Blob object from an html `<input type="file">` tag, supplied by a user.

You'll need to convert the file firstly to an `ArrayBuffer`, then to a `Uint8Array`:

```
import initMuPdf from "mupdf-js";

async function handleSomePdf(file) {
  const mupdf = await initMuPdf();
  const buf = await file.arrayBuffer();
  const arrayBuf = new Uint8Array(buf);

  //...
}
```

Once you have this, you can *load* the file into the MuPdf environment, creating a MuPdf *document*:

```
import initMuPdf from "mupdf-js";

async function handleSomePdf(file) {
  const mupdf = await initMuPdf();
  const buf = await file.arrayBuffer();
  const arrayBuf = new Uint8Array(buf);
  const doc = pdf.load(new Uint8Array(buf));
}
```

You now have three different options to render the PDF document:

```
import initMuPdf from "mupdf-js";

async function handleSomePdf(file) {
  const mupdf = await initMuPdf();
  const buf = await file.arrayBuffer();
  const arrayBuf = new Uint8Array(buf);
```

```
  const doc = pdf.load(new Uint8Array(buf));

  // Each of these returns a string:

  const png = mupdf.drawPageAsPNG(doc, 1, 300);
  const svg = mupdf.drawPageAsSVG(doc, 1);
  const html = mupdf.drawPageAsHTML(doc, 1);
}
```

Conversion Options

PNG

```
mupdf.drawPageAsPNG(document, page, resolution);
```

Arguments:

- document: *a MuPdf document object*
- page: *the page number to be rendered, starting from 1*
- resolution: *the DPI to use for rendering the file*

Returns: *an uncompressed PNG image, encoded as a base64 data URI.*

SVG

```
mupdf.drawPageAsSVG(document, page);
```

Arguments:

- document: *a MuPdf document object*
- page: *the page number to be rendered, starting from 1*

Returns: *an SVG file with the PDF document rendered as image tiles.*

HTML

```
mupdf.drawPageAsHTML(document, page);
```

Arguments:

- document: *a MuPdf document object*
- page: *the page number to be rendered, starting from 1*

Returns: *an HTML file that uses absolute positioned elements for layout.*

License

AGPL, subject to the MuPDF license.

Used by 2

@jareinnejae / **pdf-html**

@andytango / **mupdf-js-demo**

Contributors 2

andytango Andrew Hall

dependabot[bot]

Languages

TypeScript 50.0% **JavaScript** 27.6% **Makefile** 16.7% **Shell** 5.7%