MAD PDF

ASP.NET Core / NodeServices / Puppeteer / Imagemagick / Ghostscript

- Convert html to pdf
- Convert pdf to png

git: https://github.com/equinor/mad-pdf

Dev: mad-pdf-api-dev

Test: mad-pdf-api-test

Qa: mad-pdf-api-qa

Prod: mad-pdf-api-prod

Payload including pdf config, for more details look at:
https://github.com/GoogleChrome/puppeteer/blob/master/docs/api.md#pagepdfoptions.

```
    {
        html = "<p>test</p>",
        config = new
        {
            format = "A4",
            // If width or height is used format will be ignored
            // width = "100",
            // height = 100",
            landscape = false,
            margin = new { top = "10", bottom = "50" }, // top, right, bottom, left
            printBackground = true,
            preferCSSPageSize = true,
            scale = 1.0,

            displayHeaderFooter = true,

            /*
              Should be valid HTML markup with following classes used to inject printing values into them:
                date formatted print date
                title document title
                url document location
                pageNumber current page number
            */
            headerTemplate = "<div style='font-size: 10px'><span class='pageNumber'></span>My document</div>",
            footerTemplate = "<div style='font-size: 10px'><span class='url'>My document</div>"
        }
    };
```

Examples

curl pdf2png

Optional page (default = 0) dpi (default 180, range between 1 - 300)

```
curl -F 'file=@test.pdf' https://example.azurewebsites.net/api/v1/pdf/pdf2png?page=0&dpi=180 > test.png && open t
```

curl html2pdf

```
curl -d "{ html: '<h1>header </h1> <p>123</p>', config: { format: "A4", landscape: false } }" -H "Content-Type: a
```

.NET c# html2pdf

```
static void Main(string[] args)
{
    var pdfArray = CreatePdf().Result;
    System.IO.File.WriteAllBytes("myPdf.pdf", pdfArray);
}

// why static? https://docs.microsoft.com/en-us/aspnet/web-api/overview/advanced/calling-a-web-api-from-a-net-cl
private static HttpClient client = new HttpClient();

public static async Task<byte[]> CreatePdf()
{

    var payload = new
    {
        html = "<p>test</p>",
        config = new
        {
            format = "A4",
            // If width or height is used format will be ignored
            // width = "100",
            // height = 100",
            landscape = false,
            margin = new { top = "10", bottom = "50" }, // top, right, bottom, left
            printBackground = true,
            preferCSSPageSize = true,
            scale = 0.6,

            displayHeaderFooter = true,

            /*
              Should be valid HTML markup with following classes used to inject printing values into them:
                date formatted print date
                title document title
                url document location
                pageNumber current page number
            */
            headerTemplate = "<div style='font-size: 10px'><span class='pageNumber'></span>My document</div>",
            footerTemplate = "<div style='font-size: 10px'><span class='url'>My document</div>"
        }
    };
```

```
41
42      var json = JsonConvert.SerializeObject(payload);
43      var content = new StringContent(json, Encoding.UTF8, "application/json");
44
45      client.DefaultRequestHeaders.Authorization =
46          new AuthenticationHeaderValue("Bearer", "token..");
47      var result = await client.PostAsync("http://localhost:5000/api/v1/pdf/html2pdf", content);
48
49      return await result.Content.ReadAsByteArrayAsync();
50  }
```

Stresstest

```
1  wfuzz -z range,1-1000 -t 10 -d "{ html: '$payload', config: { format: 'A4', landscape: false } }" -H "Content-Typ
```

Total time: 735.9957
Processed Requests: 1000
Filtered Requests: 0
Requests/sec.: 1.358703
Success: 100%