Network and security

Nothing leaves the machine by default

- Render fetches nothing: relative addresses resolve against BaseUrl, a folder's files are read from that folder only.
- RenderFile reads the file's folder, and nothing outside it.
- RenderUrl allows the page's own host when the network is left off.

Anything else is refused and noted (`resourceRefused`). Allow hosts with HtmlNetwork:

```
var options = new HtmlPdfOptions
{
    Network = HtmlNetwork.Allow("cdn.example.com", "*.example.org") with
    {
        ResourceTimeout = TimeSpan.FromSeconds(10),
        CacheDirectory = "/var/cache/weblitz",
    },
};
```

HtmlNetwork.Any allows any public host; the machine's own addresses are never reached unless named. See the network example.

Pages behind a login, and error pages

A page behind a login prints with what it needs: headers, a bearer token, cookies, Basic or Digest credentials (see pages behind a login). Cookies are kept by site as Chrome keeps them, with the Public Suffix List: example.co.uk and other.co.uk are two sites, and a cookie a server sets for a whole public suffix (`Domain=co.uk`) is refused.

When the server answers with an error status (401, 403, 404…) and a page of its own, that page is printed, as Chrome prints it. HttpStatus gives the status and a note says so (`documentHttpError`): check it before sending the PDF on. An error with no page throws HtmlRenderException.

```
var result = HtmlToPdf.RenderUrl(new Uri("https://intranet.example.com/statement"),
options);
if (result.HttpStatus is not 200)
    throw new InvalidOperationException($"The server answered {result.HttpStatus}.");
```

Limits for pages you don't trust

HtmlPdfLimits bounds what a page may ask for: its size, its nodes, its pages, each resource and all of them; scripts are bounded by HtmlScripts.Timeout. A render that reaches a limit stops, prints what it has, returns Completed false, and notes which limit (`limitReached`): see limits. A page's own errors are notes, never exceptions; an exception (HtmlRenderException) means no PDF could be made at all.

Limits are a guard, not a sandbox. HTML from people you don't trust is best printed in a process or container of its own, without secrets and without network access beyond what the pages need.

Protected PDFs

Security encrypts the PDF (AES-256) with a user password, an owner password, and what a reader may do without the owner's: print, copy, modify. Not with PdfA: PDF/A forbids encryption. See passwords and permissions.