

Installation & Konfiguration von Containern

- [Einbetten von angehängten .pdf-Dokumenten in Bookstack](#)
- [Zabbix](#)

Einbetten von angehängten .pdf-Dokumenten in Bookstack

Die Wiki-Software Bookstack bietet nur die Funktion, .pdf-Dateien anzuhängen. Die folgende Anleitung zeigt, wie die .pdf-Dateien auch in die entsprechende Wiki-Seite eingebettet werden können.

Einmalige Vorarbeit:

1. Sich als Administrator einloggen, in den Einstellungen den Punkt Personalisierung aufrufen und folgende Einstellung suchen: "Benutzerdefinierter HTML <head> Inhalt"

Benutzerdefinierter HTML <head> Inhalt

Jeder Inhalt, der hier hinzugefügt wird, wird am Ende der <head> Sektion jeder Seite eingefügt. Diese kann praktisch sein, um CSS Styles anzupassen oder Analytics-Code hinzuzufügen.

1

Benutzerdefinierte HTML-Kopfzeileninhalte sind auf dieser Einstellungsseite deaktiviert, um sicherzustellen, dass alle Änderungen rückgängig gemacht werden können.

2. Folgendes Script in das vorgesehene Feld kopieren:

```
<script
src="https://cdnjs.cloudflare.com/ajax/libs/pdf.js/2.0.466/pdf.min.js"></script>
<style>
canvas[ data-pdfurl ] {
  background-color: lightgrey;
  width: 100%;
}
.page-content a {
  color: #39f;
  text-decoration: underline;
}
```

```
.pdf-wrapper {
position: relative;
height: 80vh;
width: 100%;
}
.pdf-wrapper .download-link {
position: absolute;
top: -2em;
right: 0;
z-index: 50;
}
.pdf-wrapper .pdf-scroller {
height: 100%;
overflow: auto;
}
</style>
<script type="text/javascript">
window.addEventListener('load', function () {
var renderPdf=function(canvas) {
    var url = canvas.dataset.pdfurl;
    var pdf = null;
    // wrap canvas in div
    var wrapper = document.createElement('div');
    wrapper.className=' pdf-wrapper';
    var scroller = document.createElement('div');
    scroller.className=' pdf-scroller';
    wrapper.appendChild(scroller);
    canvas.parentNode.insertBefore(wrapper, canvas.nextSibling);
    scroller.insertBefore(canvas, null);

    var downloadLink = document.createElement('a');
    downloadLink.href = url;
    downloadLink.className="download-link";
    downloadLink.innerText = 'Download PDF now ↓';
    wrapper.appendChild(downloadLink);

    var renderPage = function(page) {
        var scale = 1.5;
        var viewport = page.getViewport(scale);
```

```

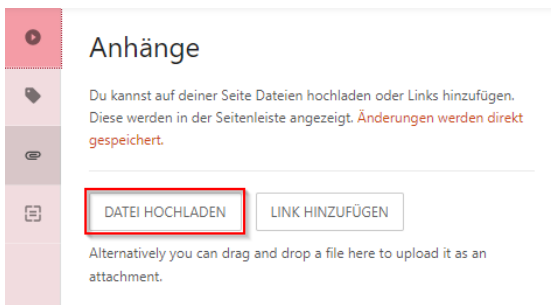
// Fetch canvas' 2d context
var context = canvas.getContext('2d');
// Set dimensions to Canvas
canvas.height = viewport.height;
canvas.width = viewport.width;
canvas.style.maxWidth='100%';
// Prepare object needed by render method
var renderContext = {
  canvasContext: context,
  viewport: viewport
};
// Render PDF page
page.render(renderContext);
if (currentPage < pdf.numPages) {
  currentPage++;
  var newCanvas = document.createElement('canvas');
  scroller.insertBefore(newCanvas, canvas.nextSibling);
  scroller.insertBefore(document.createElement('hr'), canvas.nextSibling);
  canvas=newCanvas;
  pdf.getPage(currentPage).then(renderPage);
}
};
var currentPage = 1;
pdfjsLib.getDocument(url)
  .then(function(pdfLocal) {
    pdf = pdfLocal;
    return pdf.getPage(1);
  })
  .then(renderPage);
}
Array.prototype.forEach.call(
  document.querySelectorAll('canvas[data-pdfurl]'),
  renderPdf);
});
</script>

```

3. Mit "Einstellungen speichern bestätigen"

So können die PDFs nun angehängt werden:

Das PDF muss zunächst als Anhang hochgeladen werden.



In den Anhängen erscheint nun die hochgeladene .pdf-Datei. Wenn man den Mauszeiger darauf bewegt (nicht klicken), sieht man unten links im Browserfenster einen Link der immer gleich aufgebaut ist: <https://domain.net/attachments/X>. Das X steht hier für die ID des Attachments und wird im weiteren Verlauf sehr wichtig.



Als nächstes wird der Quellcode des Wiki-Eintrages über die Navigationsleiste geöffnet:



An der gewünschten Stelle, wo das PDF erscheinen soll, wird folgende Zeile eingefügt:

```
<p>&nbsp;<canvas data-pdfurl=" https: //domain. net/attachments/X" ></canvas>&nbsp;</p>
```

also in meinem Beispiel:

```
<p>&nbsp;<canvas data-pdfurl="https://wiki.crypticghost.boo/attachments/2"></canvas>&nbsp;</p>
```

Schaut dann so aus:

Quellcode ✕

```
<p id="bkmrk--1"><a href="https://denkarium.nreinsch.net/uploads/images/gallery/2024-06/09E2nWsQLCqfl8mA-image.png" target="_blank" rel="noopener"></a></p>
<p id="bkmrk-als-n%C3%A4chstes-wird-de">Als nächstes wird der Quellcode des Wiki-Eintrages über die Navigationsleiste geöffnet.</p>
<p id="bkmrk--2"><a href="https://denkarium.nreinsch.net/uploads/images/gallery/2024-06/A1ezFCDQRNXTfEGL-image.png" target="_blank" rel="noopener"></a></p>
<p id="bkmrk-an-der-gew%C3%Bcnschten-s">An der gewünschten Stelle, wo das PDF erscheinen soll, wird folgende Zeile eingefügt:</p>
<pre id="bkmrk-%3Cp%3E%26nbsp%3B%3Ccanvas-dat"><code class="language-html">&lt;p&gt;&nbsp;&lt;canvas data-pdfurl="https://domain.net/attachments/X"&gt;&lt;/canvas&gt;&nbsp;&lt;/p&gt;</code></pre>
<p id="bkmrk-also-in-meinem-beisp">also in meinem Beispiel.</p>
<pre id="bkmrk-%3Cp%3E%26nbsp%3B%3Ccanvas-dat-1"><code class="language-html">&lt;p&gt;&nbsp;&lt;canvas data-pdfurl="https://denkarium.nreinsch.net/attachments/2"&gt;&lt;/canvas&gt;&nbsp;&lt;/p&gt;</code></pre>
<p id="bkmrk-%C2%A0-7"></p>
<p id="bkmrk-%C2%A0-8"></p>
<p id="bkmrk-%C2%A0-9"></p>
<p id="bkmrk-%C2%A0-10"></p>
<p id="bkmrk-%C2%A0-11"></p>
<p id="bkmrk-%C2%A0-12"></p>
<p id="bkmrk-%C2%A0-13"></p>
<p id="bkmrk-%C2%A0-14"></p>
<p>&nbsp;<canvas data-pdfurl="https://denkarium.nreinsch.net/attachments/2"></canvas>&nbsp;</p>
```

Abbrechen Speichern

Nach Speichern des Quellcodes und des Wiki-Artikels erscheint das PDF an der gewünschten Stelle im Artikel.

Zabbix

```
services:
  postgres:
    image: postgres:13-alpine
    container_name: zabbix-postgres
    environment:
      POSTGRES_USER: zabbix
      POSTGRES_PASSWORD: dshdshdfshdsf
      POSTGRES_DB: zabbix
    volumes:
      - zabbix_pgdata: /var/lib/postgresql/data
    healthcheck:
      test:
        - CMD-SHELL
        - pg_isready -U zabbix
      interval: 10s
      retries: 5
      timeout: 5s
  zabbix-server:
    image: zabbix/zabbix-server-pgsql:alpine-6.4-latest
    container_name: zabbix-server
    environment:
      DB_SERVER_HOST: postgres
      POSTGRES_DB: zabbix
      POSTGRES_USER: zabbix
      POSTGRES_PASSWORD: dshdshdfshdsf
    depends_on:
      postgres:
        condition: service_healthy
    ports:
      - 10051:10051
  zabbix-web:
    image: zabbix/zabbix-web-nginx-pgsql:alpine-6.4-latest
    container_name: zabbix-web
    environment:
      DB_SERVER_HOST: postgres
```

POSTGRES_DB: zabbix

POSTGRES_USER: zabbix

POSTGRES_PASSWORD: dshdshdfshdsf

PHP_TZ: Europe/Berlin

ZBX_SERVER_HOST: zabbix-server

ports:

- 8080: 8080

depends_on:

zabbix-server:

condition: service_started

volumes:

zabbix_pgdata: null