

Previsualizar pdf

- <https://github.com/BookStackApp/BookStack/issues/705>
- <https://gist.github.com/aghster/8e7d8ea53ac6c8fe74698ede242c2a37>
- <https://github.com/ddkhanh/bookstack-pdf-viewer/>
- <https://gist.github.com/sprklinginfo/624c4c431a0f802556071ec459b6ca76>

Here's an altered version of the above scripts that I recently amended for a BookStack user, that was using the above hacks. Think I tested it in v22.06. This script is not official nor supported:

View Code

I eventually got it working by also having a second file named pdf.worker.min.js along side pdf.min.js.

```
<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">

    // ----- THIS SECTION ADDS A PDF BUTTON TO THE EDITOR TOOLBAR THAT ALLOWS YOU
    TO EMBED PDFS

    // Use BookStack editor event to add custom "Insert PDF" button into main toolbar
    window.addEventListener('editor-tinymce::pre-init', event => {
        const mceConfig = event.detail.config;
        mceConfig.toolbar = mceConfig.toolbar.replace('link', 'link insertpdf')
    });

    // Use BookStack editor event to define the custom "Insert PDF" button.
    window.addEventListener('editor-tinymce::setup', event => {
        const editor = event.detail.editor;

        // Add PDF insert button
        editor.ui.registry.addButton('insertpdf', {
            tooltip: 'Insert PDF',
            icon: 'document-properties',
            onAction() {
                editor.windowManager.open({
                    title: 'Insert PDF',
                    body: {
                        type: 'panel',
                        items: [
                            {type: 'textarea', name: 'pdfurl', label: 'PDF URL'}
                        ]
                    },
                    onSubmit: function(e) {
                        // Insert content when the window form is submitted
                        editor.insertContent('<p>&nbsp;<canvas data-pdfurl="' + e.getData().pdfurl +
''></canvas>&nbsp;</p>');
                        e.close();
                    },
                    buttons: [
                        {
                            type: 'submit',
                            text: 'Insert PDF'

```

```

        }
    ]
    });
}
});

});

```

//----- THE CODE BELOW SHALL BE ACTIVE IN VIEWING MODE TO EMBED PDFS

```

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);
};

window.addEventListener('DOMContentLoaded', function() {
    Array.prototype.forEach.call(document.querySelectorAll('canvas[data-pdfurl]'), renderPdf);
});
</script>
```

Revision #1

Created 2026-08-23 15:07:49 UTC by Mobles Martí

Updated 2026-08-23 15:08:36 UTC by Mobles Martí