

[DOCS](#)[CONTACT SALES](#)[Web](#)[Choose a Page](#)[GETTING STARTED](#)[WEB INTEGRATIONS](#)

Getting started on Web Integrations

[SHAREPOINT](#)[ONLINE](#)[WEB PART](#)

Integrate into SharePoint Online as a Web Part

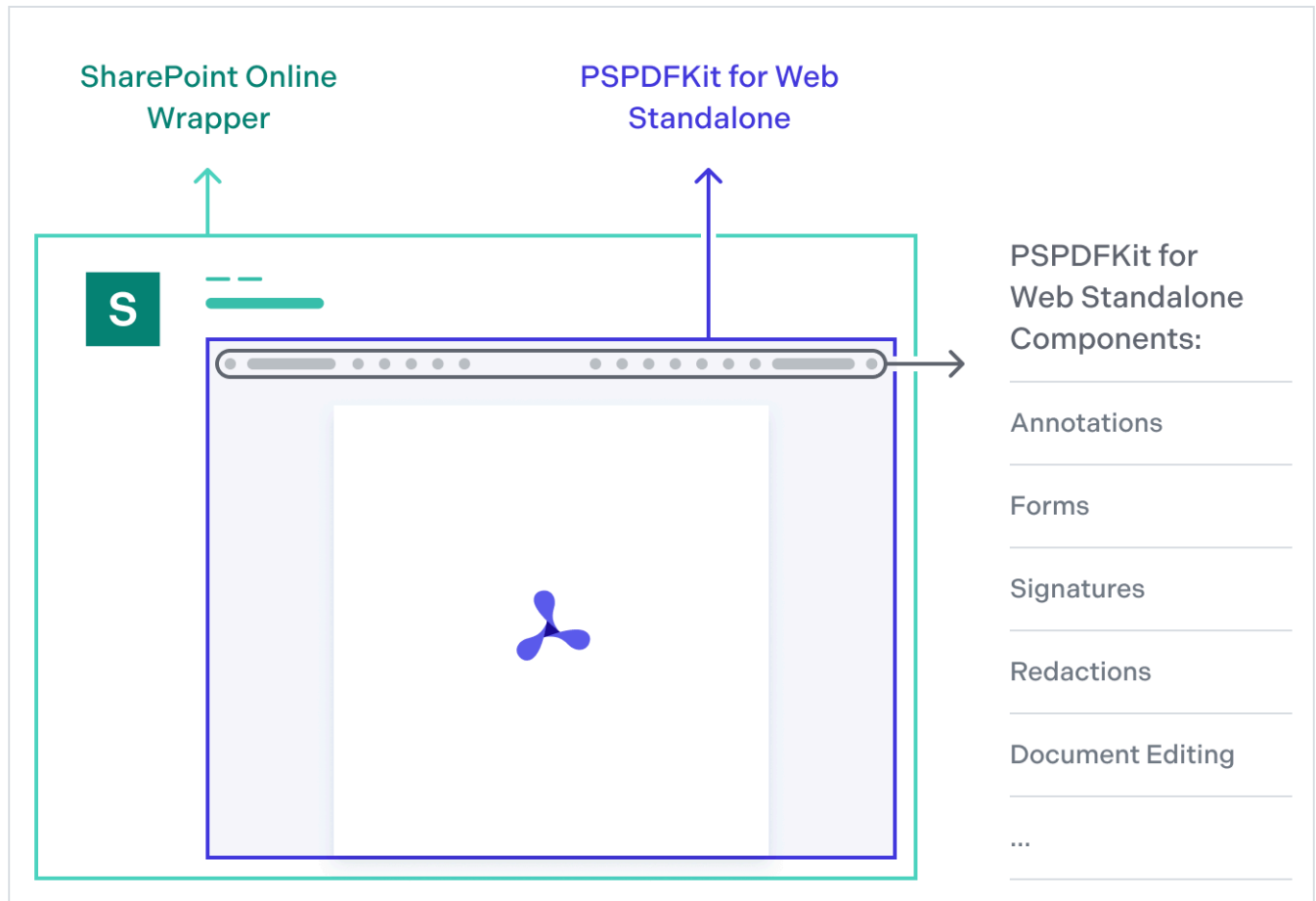
This guide will walk you through the steps necessary to integrate PSPDFKit for Web into SharePoint Online pages using a web part. By the end of this guide, you'll be able to use a web browser to directly open PDF documents stored in SharePoint, as well as display, edit, and save them back to SharePoint.

PSPDFKit for SharePoint is a wrapper on top of PSPDFKit for Web Standalone, which is a client-side JavaScript library for viewing and editing PDF documents directly in a web browser.

PSPDFKit for SharePoint shares the same APIs as PSPDFKit for Web Standalone, so use the Web documentation when customizing your SharePoint application.

[TRY OUR NO-CODE APP](#)[ASK AI](#)

PSPDFKit for SharePoint



Requirements

A Microsoft 365 tenant (you can set up a Microsoft 365 tenant for free when you join the Microsoft 365 Developer Program)

The SharePoint Framework

A GitHub account (for cloning the PSPDFKit for SharePoint repository)

Node.js version 16.19.0

1 Setup

- 1 Clone the `pspdfkit-sp-online-webpart` repository from GitHub.

Navigate to the directory where you want to place the SharePoint integration, and type the following command in your command line/terminal:

```
git clone https://github.com/PSPDFKit/pspdfkit-sp-online-webpart.git
```

Alternatively, you can download the project as a `.zip` file without cloning the project.

2 Navigate to `admin.microsoft.com` and sign in to your tenant account.

To make the SharePoint catalog available, click `Show All...` in the side navigation, and find `Admin centers > SharePoint`. This will direct you to SharePoint Admin portal.

0:00 / 0:05

Follow along with this tutorial from Microsoft to configure your SharePoint tenant and create your new team site.

3 To configure your URL, go to the `config/serve.json` file in your `pspdfkit-sp-online-webpart` project and change `enter-your-SharePoint-site` to the URL of your SharePoint site:

```
"initialPage": "https://enter-your-SharePoint-site/_layouts/workbench.as
```

- 4 Now, set up your development environment to build a client-side web part.

Don't forget to execute `gulp trust-dev-cert` from the root of your project to trust the self-signed developer certificate.

- 5 Change your directory into the `pspdfkit-sp-online-webpart` directory:


```
cd pspdfkit-sp-online-webpart
```

- 6 Open the project in your preferred code editor and check which version of Node.js you're using:

```
node --version
```




Make sure you have Node.js version 16.19.0 installed. If you're using a different version of Node, switch to Node.js version 16.19.0 via `nvm` (node version manager) or `asdf`.

 **Tip:** If you're using `asdf`, you can switch the Node version locally for this project. First, install Node.js 16.19.0 using the following command:

```
asdf install nodejs 16.19.0
```

Now, run `asdf local nodejs 16.19.0` to switch to the Node.js version you just installed. This will create a `.tool-versions` file in the root of your project with Node version 16.19.0. If you encounter any issues, check out the [asdf documentation](#).

 **Tip:** If you're using `nvm`, you can switch to Node.js 16.19.0 by typing the following command:

```
nvm use 16.19.0
```

7 Install the dependencies:

```
npm install
```

8 To serve the project, type:

```
npm run serve
```

The browser will open a new tab with your SharePoint account.

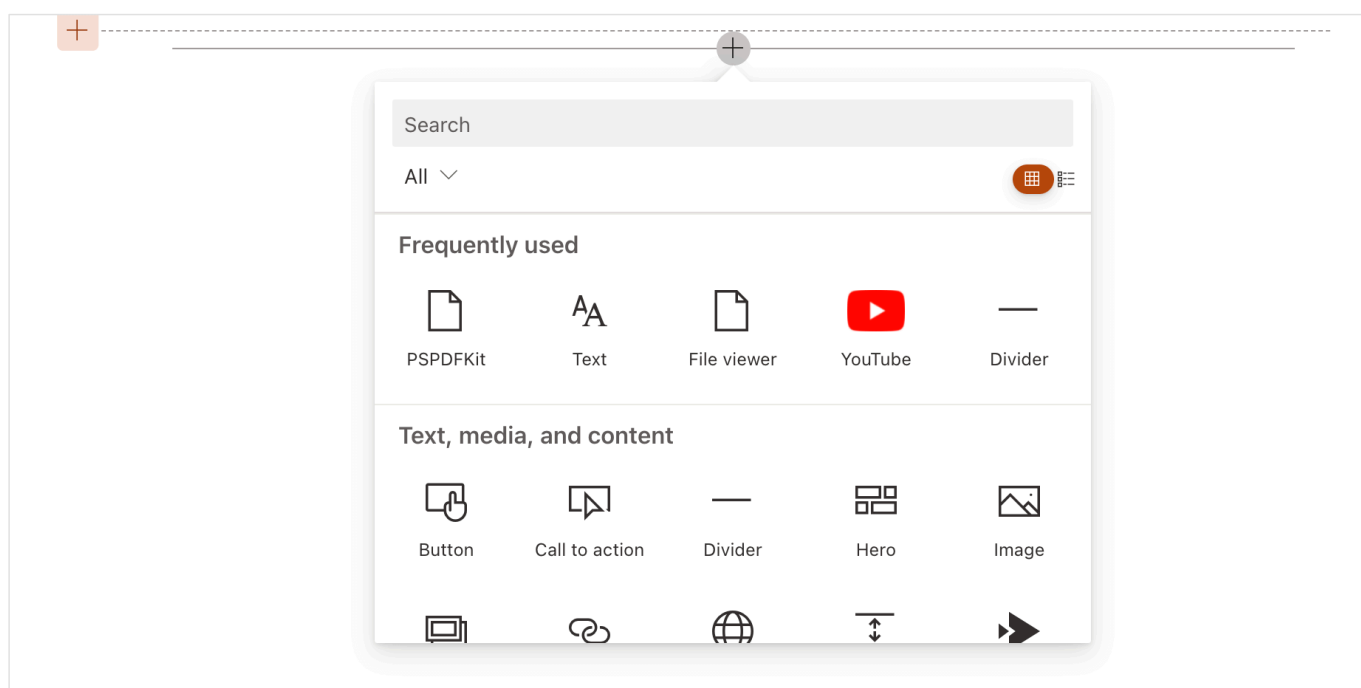
2 Integrating PSPDFKit into SharePoint

PSPDFKit can be added to your project as a widget. You can add it to your project using the UI elements.

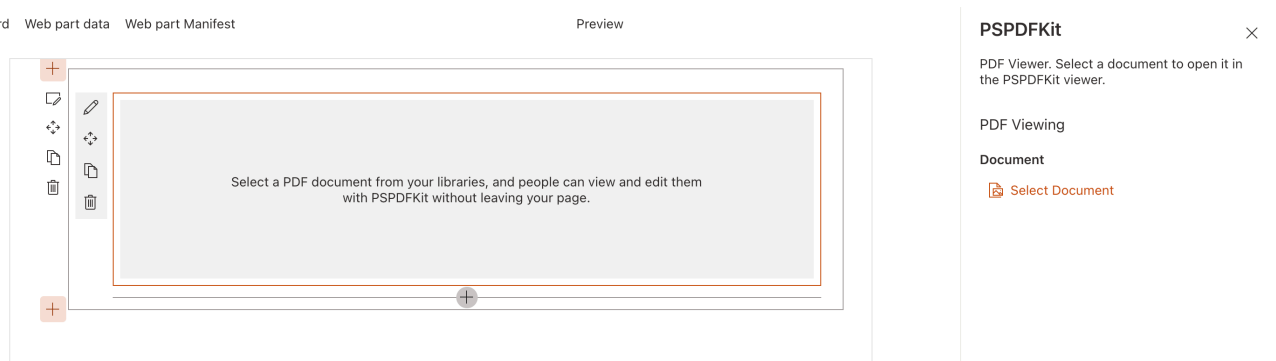
Refer to the following video for information on how to integrate the PSPDFKit widget into your project, or continue with the instructions.

0:00 / 0:19

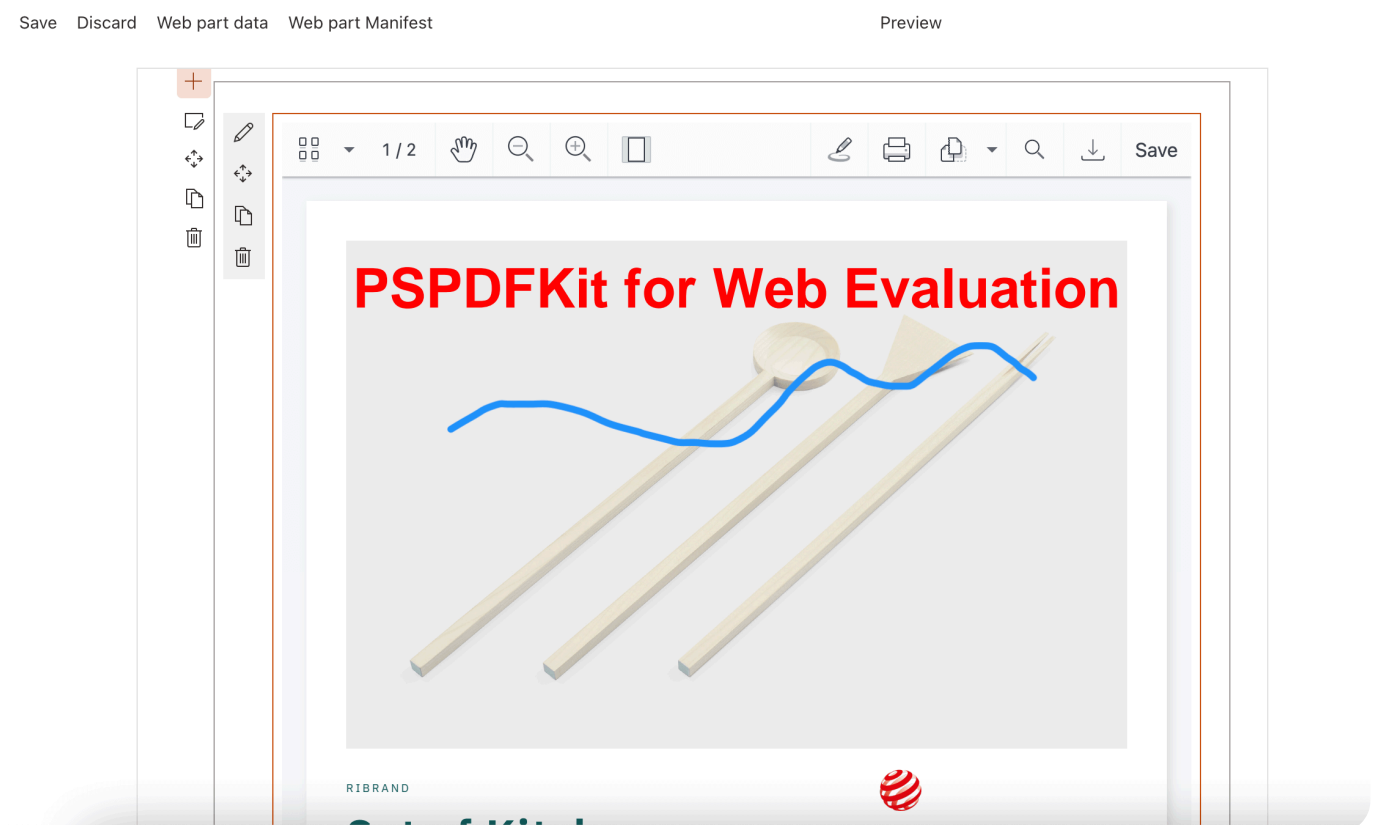
- 1 On your Home page, in the upper right-hand corner, find the Edit button.
- 2 Click the Edit button. This will open the Editing interface.
- 3 To add the PSPDFKit widget, hover over where you want to place it. This will display a plus icon. When you click the plus icon, it'll open a dialog box with the possible web parts you can incorporate into your project.
- 4 Search for the PSPDFKit widget and click on it.



- 5 Find the edit button with the pencil icon, and click Select Document from the open panel.



- 6 Choose your document and click Open. Now you'll be able to see the PSPDFKit widget in your project like in the video above.



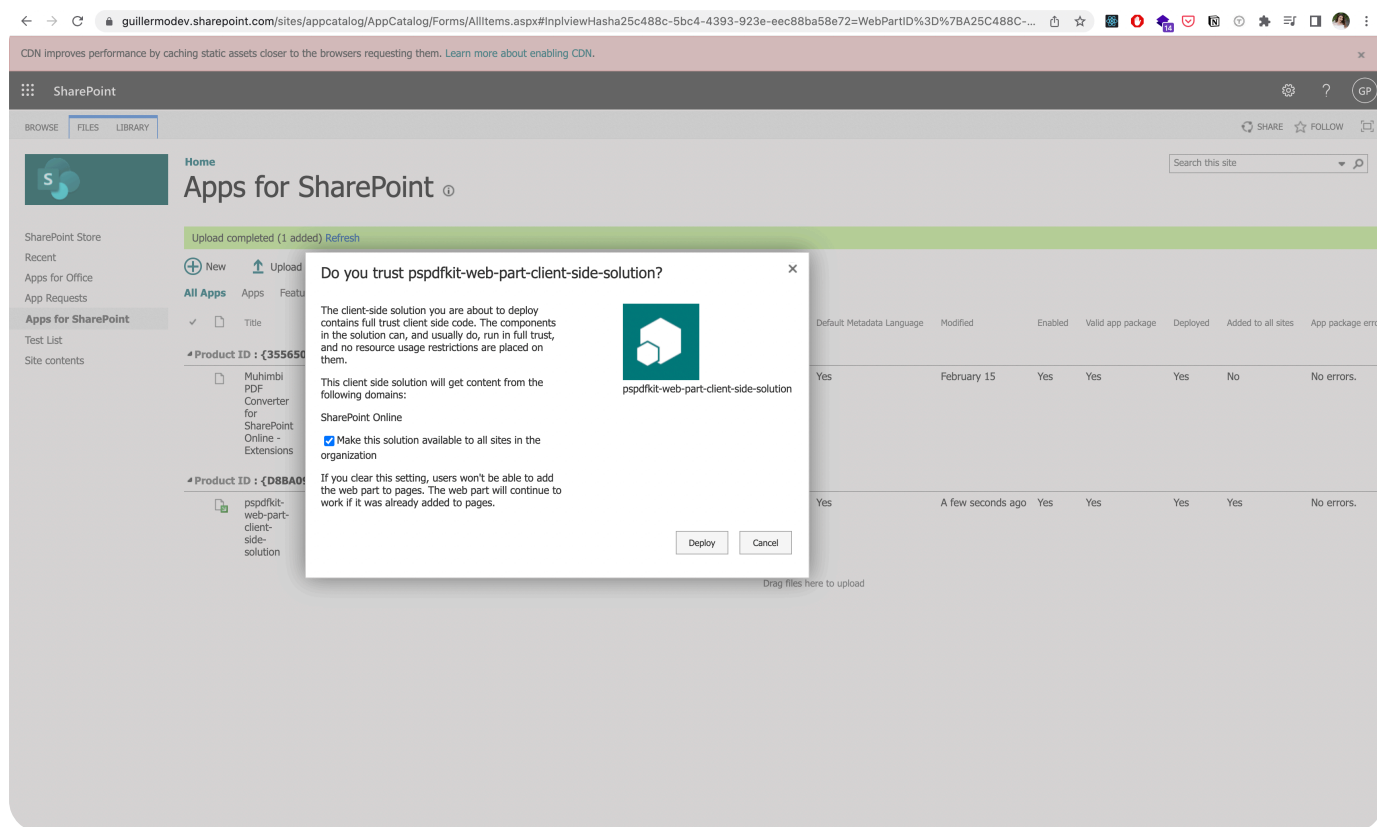
3 Using for Production

For production, you need to follow these steps to deploy into your SharePoint environment.

- 1 Run `npm run bundle` and then `npm run package`.
- 2 Then, follow the steps outlined in this SharePoint guide for packaging and deploying your web part. If a previous installation already exists, it's recommended to remove it before proceeding.



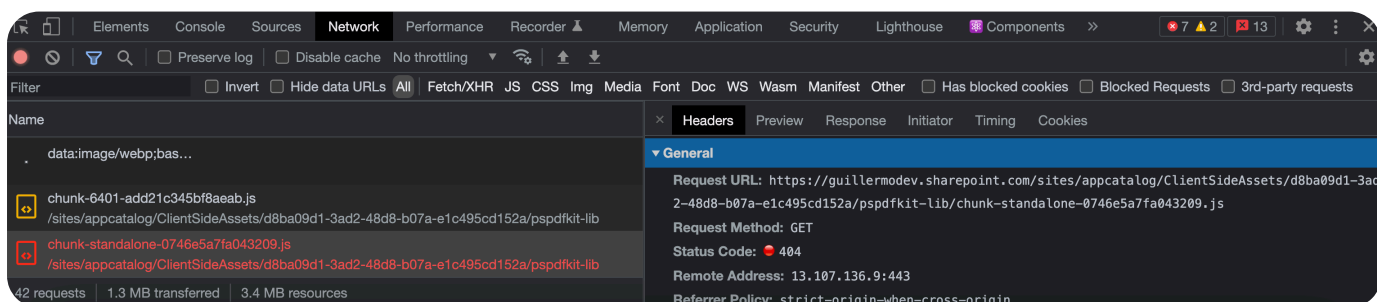
Make sure to select **Make this solution available to all sites in the organization**. This appears in the confirmation dialog you'll see after dragging the package.



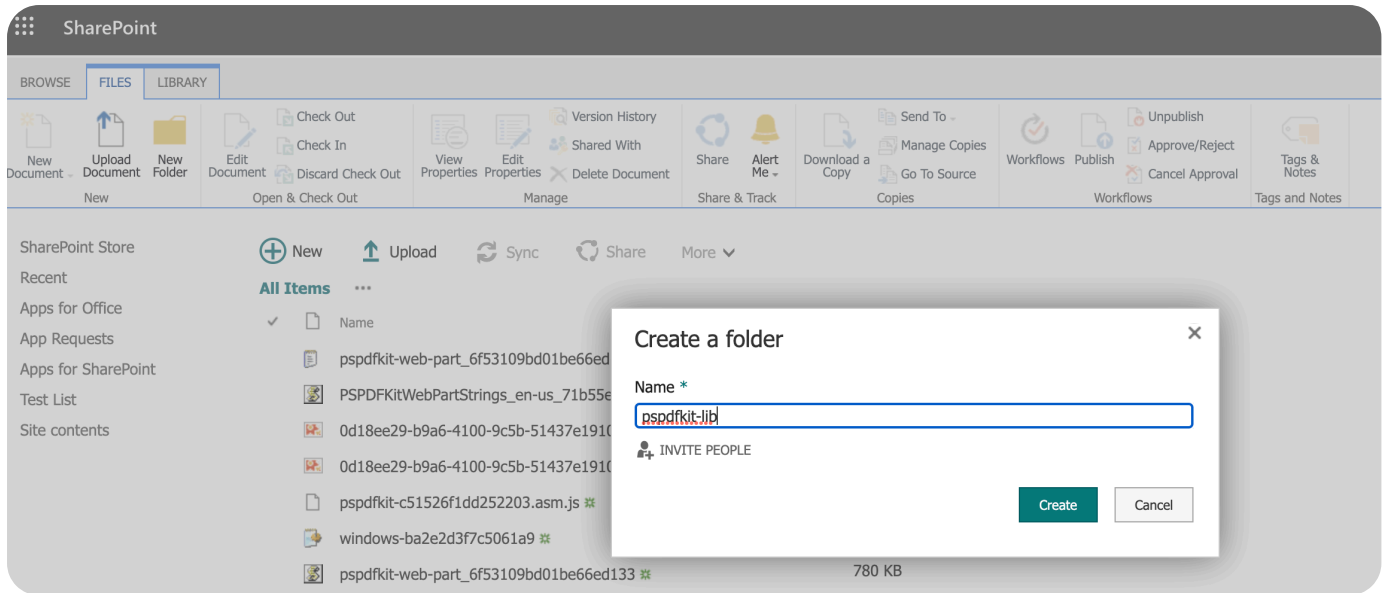
3 To make the web part available in Microsoft Teams, synchronize your solution with Teams.

0:00 / 0:04

- 4 Navigate to the workbench of one of your SharePoint sites and add the PSPDFKit web part there.
- 5 There's an issue getting SharePoint to bundle the static assets of PSPDFKit for Web under the correct subdirectory when using the SharePoint file system itself as your CDN instead of a custom one (you can specify a custom URL for the static assets under `write-manifests.json`), so you might see a network error when requesting one of the chunks for the PSPDFKit library.



If this happens, it's because the `pspdfkit-lib` subdirectory hasn't been created by SharePoint. To fix this, navigate to the erroring URL but trim its path before the `pspdfkit-lib` segment (i.e. `https://<your-sharepoint-site>/sites/appcatalog/ClientSideAssets/<some-hash>`) and create a `pspdfkit-lib` folder.



Then, you can copy the contents from `./sharepoint/solution/debug/ClientSideAssets/pspdfkit-lib` to the newly created folder.



Make sure you have the latest version of PSPDFKit inside the pspdfkit-lib folder.



Note: As an alternative to this step, consider hosting the assets in a CDN and set it in the `write-manifests.json` file.

Was this helpful?

6 Your PSPDFKit instance should now work. Try opening your workbench and check if the PSPDFKit web part appears as part of the list of available web parts to add.

YES

NO

You can now begin customizing the user interface, implementing custom functionality, or adding a custom file handler for PDF files.

Questions? Contact us

Next Steps

Implementing a custom file handler