



Web



Text to PDF



On this page



WEB > CONVERSION > TEXT TO PDF > SHAREPOINT

Convert text to PDF in SharePoint



COPY PAGE



Nutrient Web SDK is a client-side JavaScript library that's fully compatible with Microsoft SharePoint for converting text documents to PDF directly in the browser, without the need for server-side processing. To convert text files to PDF, Nutrient Web SDK relies entirely on its own technology built from the ground up, and it doesn't depend on third-party tools. For more information on the supported formats, see the [list of supported file types](#).

TRY FOR FREE

Converting text documents to PDFs after displaying

To convert a text document to a PDF after displaying it in the Nutrient viewer, follow the steps below.

- 1 [Load the source text document](#) (optional). To load the document without a user interface visible to the user, use the `headless` parameter.
- 2 Make changes to the document (optional). For example, [add annotations](#).



ASK AI

- 3 Convert the source document to a PDF with the `exportPDF` method (optional). Use the `outputFormat` flag to create a PDF/A document. For more information, see [converting PDF to PDF/A](#).
- 4 Save the output document. The `exportPDF` method returns a `Promise` that resolves to an `ArrayBuffer` containing the output PDF document. You can use the resulting `ArrayBuffer` to download or persist the output PDF in storage. For more information on downloading or persisting the exported `ArrayBuffer`, see the [guides on saving a document](#).

The following example loads a text document and exports it to a PDF:

```
NutrientViewer.load({
  container: "#pspdfkit",
  document: "source.txt",
  licenseKey: "YOUR_LICENSE_KEY"
}).then((instance) => {
  instance.exportPDF();
});
```



The following example loads a text document, exports it to a PDF with conformance level PDF/A-4f, and downloads it in the client's browser:

```
NutrientViewer.load({
  container: "#pspdfkit",
  document: "source.txt",
  licenseKey: "YOUR_LICENSE_KEY"
})
.then((instance) =>
  instance.exportPDF({
    outputFormat: {
      conformance: NutrientViewer.Conformance.PDFA_4F
    }
  })
)
.then(function (buffer) {
  const blob = new Blob([buffer], { type: "application/pdf" });
  const objectUrl = window.URL.createObjectURL(blob);
  downloadPdf(objectUrl);
  window.URL.revokeObjectURL(objectUrl);
});
```



```
});  
function downloadPdf(blob) {  
  const a = document.createElement("a");  
  a.href = blob;  
  a.style.display = "none";  
  a.download = "output.pdf";  
  a.setAttribute("download", "output.pdf");  
  document.body.appendChild(a);  
  a.click();  
  document.body.removeChild(a);  
}
```

When exporting a document, you have several options. Refer to our guides on [flattening annotations](#) and [incremental saving](#) for more details.

Auto saving can be configured for different scenarios and use cases. You can find more information in our [auto save](#) guide.

Converting text documents to PDFs without displaying

To convert a text document to a PDF without displaying it in the Nutrient viewer, follow the steps below.

- 1 Load and convert the source text document using the `convertToPDF` method. This method takes the following parameters:
 - ⌘ A `Configuration` object that specifies the path to the source document and the license key.
 - ⌘ A member of the `Conformance` enumeration that specifies the conformance level of the output PDF document (optional). If you provide this parameter, the output is a PDF/A document.
- 2 Save the output document. The `convertToPDF` method returns a `Promise` that resolves to an `ArrayBuffer` containing the output PDF document. You can use the resulting `ArrayBuffer` to download or persist the output PDF in storage. For more information on downloading or persisting the exported `ArrayBuffer`, see the [guides on saving a document](#).

The following example exports the loaded document to a PDF with conformance level PDF/A-4f:

```
NutrientViewer.convertToPDF(  
  {  
    document: "source.txt",  
    licenseKey: "YOUR_LICENSE_KEY"  
  },  
  NutrientViewer.Conformance.PDFA_4F  
);
```



The following example converts a text document to a PDF document and downloads it in the client's browser:

```
NutrientViewer.convertToPDF({  
  document: "source.txt",  
  licenseKey: "YOUR_LICENSE_KEY"  
}).then(function (buffer) {  
  const blob = new Blob([buffer], { type: "application/pdf" });  
  const objectUrl = window.URL.createObjectURL(blob);  
  downloadPdf(objectUrl);  
  window.URL.revokeObjectURL(objectUrl);  
});  
function downloadPdf(blob) {  
  const a = document.createElement("a");  
  a.href = blob;  
  a.style.display = "none";  
  a.download = "output.pdf";  
  a.setAttribute("download", "output.pdf");  
  document.body.appendChild(a);  
  a.click();  
  document.body.removeChild(a);  
}
```



When exporting a document, you have several options. Refer to our guides on [flattening annotations](#) and [incremental saving](#) for more details.

Auto saving can be configured for different scenarios and use cases. You can find more information in our [auto save](#) guide.

Was this helpful?

☒ YES

☐ NO