

Formular als PDF erzeugen

html2pdf

1. Import

Zunächst müssen Sie die Library html2pdf.js einbinden. Dies erfolgt über die folgende Methode:

```
loadCScript('https://cdnjs.cloudflare.com/ajax/libs/html2pdf.js/0.10.1/html2pdf.bundle.min.js')
);
```

Diese Methode lädt das Skript aus einem öffentlichen CDN nach.

2. Was soll gedruckt werden?

Der Bereich, den Sie als PDF ausgeben möchten, muss im Formular mit einer eindeutigen CSS-Klasse gekennzeichnet werden, z. B.:

```
<div class="formio-component-printContainer">
  <!-- Hier steht der Inhalt, der später als PDF ausgegeben werden soll -->
</div>
```

Diese Klasse wird später gezielt zur PDF-Erzeugung angesprochen.

3. PDF-Funktionen nutzen

3.1 PDF herunterladen

Mit der folgenden Funktion können Sie den gewünschten HTML-Bereich als PDF im Browser speichern lassen:

```
const downloadPDF = async (cssClassName) => {
  const element = document.getElementsByClassName(cssClassName);
  return html2pdf()
    .set({ html2canvas: { scale: 4 }, margin: [10, 10, 10, 10] })
    .from(element[0])
    .save("dokument.pdf");
};
```

Hinweis zur Qualität

Mit `html2canvas: { scale: 4 }` wird das PDF in hoher Auflösung erstellt. Falls die Erstellung zu lange dauert, können Sie den Wert auf z.B. 2 reduzieren, um die Auflösung zu senken und so die Geschwindigkeit zu erhöhen.

Beispiel-Aufruf im Formular-Script:

```
downloadPDF('formio-component-printContainer');
```

3.2 PDF als Blob zur weiteren Verarbeitung erzeugen

Falls das PDF z. B. in das Dokumentenmanagementsystem hochgeladen werden soll, können Sie den Inhalt auch als Blob ausgeben:

```
const getBlobPDF = async (cssClassName) => {  
  const element = document.getElementsByClassName(cssClassName);  
  return html2pdf()  
    .set({ html2canvas: { scale: 4 }, margin: [10, 10, 10, 10] })  
    .from(element[0])  
    .output("blob", "dokument.pdf");  
};
```

Hinweis zur Qualität

Mit `html2canvas: { scale: 4 }` wird das PDF in hoher Auflösung erstellt. Falls die Erstellung zu lange dauert, können Sie den Wert auf z.B. 2 reduzieren, um die Auflösung zu senken und so die Geschwindigkeit zu erhöhen.

4. Verwendung

Sie können die Funktion z. B. über einen Formular-Button auslösen:

```
async function exportToPDF(form, data) {  
  await downloadPDF('formio-component-printContainer');  
}
```

pdfMake

1. Import

Die pdfmake Library wird im serverseitigen Skript über die "Externen Bibliotheken" im Prozessstudio geladen.

2. Struktur der PDF festlegen

Im Formularskript wird ein Inhalt (content) als JSON-Struktur erzeugt, der an den Server übergeben wird. Zum Beispiel im Kalkulationsschema:

```
var content = getHeader(form.t('Kalkulationsschema FIS'));
content.push(
  { text: form.t('Projekt-Nr.: ' + data.printContainer.projektNr), style: 'subheader' },
  { text: form.t('Bezeichnung: ' + data.printContainer.bezeichnung), style: 'subheader' },
  { style: 'tableExample', table: {
    headerRows: 1,
    body: [
      [{text: 'Nr.', style:'tableHeader'}, {text: 'Kalkulationsbestandteil',
style:'tableHeader'}, {text: 'Betrag', style:'tableHeader'}],
      ['1', 'Personalkosten', data.printContainer.PersCost_sum + ' €'],
      ...
    ]
  }, layout: 'headerLineOnly' }
);
```

“ Gestaltung der PDF

Zur Gestaltung einer PDF verwenden Sie am besten den pdfmake-playground:

<http://pdfmake.org/playground.html>

3. Erstellung der PDF

Die JSON-Daten werden an ein Server-Skript übermittelt, das mit pdfmake das PDF generiert. Hier ein Beispiel des Aufrufs im Kalkulationsschema:

```
async function createRequestPDF(form, data) {
  var content = getHeader(form.t('Kalkulationsschema FIS'));
  content.push(
```

```

{
    text: form.t('Projekt-Nr.: ' + data.printContainer.projektNr) ,
    style: 'subheader'
},
{
    text: form.t('Bezeichnung: ' + data.printContainer.bezeichnung) ,
    style: 'subheader'
},
{
    style: 'tableExample',
    table: {
        headerRows: 1,
        body: [
            [{text: 'Nr.',style:'tableHeader'} ,{text:
'Kalkulationsbestandteil',style:'tableHeader'} ,{text: 'Betrag',style:'tableHeader'}],
            ['1', 'Personalkosten', data.printContainer.PersCost_sum + ' €'],
            ['1.1', 'Personalkosten Beschäftigte/Beamte',
data.printContainer.PersCost_value + ' €'],
            ['1.2', 'Sonstige Personalkosten', data.printContainer.PersCostOther_value
+ ' €'],
            ['1.3', 'Forschungszulagen', data.printContainer.Forschungszulage_value +
' €'],
            ['2', 'Sachkosten', data.printContainer.Sachkosten_value],
            ['2.1', 'Reisekosten', data.printContainer.Reisekosten_value + ' €'],
            ['2.2', 'Fremdleistungen', data.printContainer.Fremdleistungen_value + '
€'],
            ['2.3', 'Sonstige Sachkosten', data.printContainer.SachkostenOther_value +
' €'],
            ['3', 'Abschreibungen / Investitionen',
data.printContainer.abschreibungenInvestitionen_value + ' €'],
            ['4', {text: 'Gemeinkosten für Infrastruktur\n(= 1. * Zuschlagssatz i.H.v.
70%)' } , data.printContainer.gemeinkostenFurInfrastruktur_value + ' €'],
            ['5', 'Gesamtkosten\n(= 1. + 2. + 3. + 4.)',
data.printContainer.Gesamtkosten_value + ' €'],
            ['6', 'Gewinnzuschlag min. 3% von den Gesamtkosten.',
data.printContainer.Gewinnzuschlag_value + ' €'],
            ['7', 'Gesamtentgelt / Preisuntergrenze\n(= 5. + 6.)',
data.printContainer.Gesamtkosten_value1 + ' €']
        ]
    },
},

```

```

        layout: 'headerLineOnly'
    },
    )
    pdfData = {
        content: content,
        styles: {
            header: {
                fontSize: 18,
                bold: true,
                alignment: 'left',
                margin: [0, 16, 0, 0]
            },
            subheader: {
                fontSize: 16,
                bold: true,
                margin: [0, 10, 0, 5]
            },
            superMargin: {
                margin: [20, 0, 40, 0],
                fontSize: 15
            },
            tableExample: {
                margin: [0, 5, 0, 15]
            },
            tableHeader: {
                bold: true,
                fontSize: 16,
                color: 'black'
            }
        }
    }

    // Aufruf Script zur Erstellung PDF
    var result = await fetch("/scripting/script/f9dafc81-a9a7-43a8-933b-cbe4dff4f7a0/run",
{
    method: "POST",
    body: JSON.stringify(pdfData)
})

```

```

    if (result.ok) {
        var blob = await result.blob();
        var file = window.URL.createObjectURL(blob);
        window.open(file, "blank");
    }
}

```

Die Headerdaten werden aus der Methode `getHeader` abgerufen:

Die JSON-Daten werden an ein Server-Skript übermittelt, das mit `pdfmake` das PDF generiert. Hier ein Beispiel des Aufrufs im Kalkulationsschema:

```

function getHeader(title) {
    return [
        {
            image: '
/2Q==', ' ',
            width: 520,
            height: 100
        },
        {
            text: title,
            absolutePosition: { x: 200, y: 75 },
            color: '#FFF',
            fontSize: '24'
        },
        {
            text: 'd.velop',
            absolutePosition: { x: 500, y: 50 },
            color: '#FFF',
            fontSize: '14',
            bold: true
        }
    ]
}

```