

Inhaltsverzeichnis

Ein einfaches Paket für Verkaufsdokumente erstellen	2
1. Verzeichnis anlegen	3
2. Vorlage hinzufügen	3
3. Stile hinzufügen	4
4. Import-ZIP erstellen und prüfen	4
Ein Paket für Zeitrachweise erstellen	4
Artspezifische Präsentation	5
Abgestimmtes Seitendesign	6
Import und Verantwortung	6

Ein einfaches Paket für Verkaufsdokumente erstellen

Diese Anleitung erstellt ein bewusst einfaches Paket. Die meisten visuellen Entscheidungen sind fest vorgegeben. Verwendet wird nur der wesentliche öffentliche Vertrag: Manifest, Seitenränder, vorbereitete Dokumentpositionen und die anwendungseigene Finanzübersicht. **Experimentell:** Template API v1 ist nicht stabil; teste dieses Paket mit jeder Anwendungsversion erneut.

Eigene Pakete kannst Du sowohl für Verkaufsdokumente als auch für Zeitrachweise im jeweiligen Editor für Lay-outvorlagen importieren. Das vollständige Beispiel für Verkaufsdokumente liegt unter `examples/simple/` und wird von den Vertragstests des Repositories kompiliert und gerendert.

Mit den gepflegten englischen Testdaten sieht die erste gerenderte Seite so aus:

TechVision Solutions GmbH

Order Confirmation

Document number: OC-BASE-001 · Document date: Aug 19, 2026

Example Engineering Ltd.

Customer Street
54321 Customer City
DE

Position	Description	Quantity	Total
1	Implementation and commissioning 1 – deterministic long description for pagination and wrapping verification	1 hour	€76.25
2	Implementation and commissioning 2 – deterministic long description for pagination and wrapping verification	1 hour	€77.50
3	Implementation and commissioning 3 – deterministic long description for pagination and wrapping verification	1 hour	€78.75
4	Implementation and commissioning 4 – deterministic long description for pagination and wrapping verification	1 hour	€80.00
5	Implementation and commissioning 5 – deterministic long description for pagination and wrapping verification	1 hour	€81.25
6	Implementation and commissioning 6 – deterministic long description for pagination and wrapping verification	1 hour	€82.50
7	Implementation and commissioning 7 – deterministic long description for pagination and wrapping verification	1 hour	€83.75

Die Testdaten enthalten bewusst genügend Positionen, um den Seitenumbruch zu zeigen; die Finanzübersicht folgt auf

der zweiten Seite.

1. Verzeichnis anlegen

Lege diese drei Dateien an:

```
manifest.json
document.peb
styles.css
```

Kopiere das Beispielmanifest. Es deklariert einen Einstiegspunkt vom Typ `sales_document`, alle derzeit unterstützten Dokumentfamilien, Template API v1 und ausschließlich `page.margins` als berücksichtigte Einstellung. Einzelheiten findest Du in der generierten Referenz unter Manifest und Diagnosen.

2. Vorlage hinzufügen

Verwende diese vollständige `document.peb`:

```
<!DOCTYPE html>
<html lang="{{ common.locale }}">
<head>
  <meta charset="UTF-8" />
  <style>{{ common.resources.fontCss | raw }}</style>
  <style>{% include "styles.css" %}</style>
  <style>
    @page {
      size: {{ settings.page.size }};
      margin: {{ settings.page.marginTopMm }}mm {{ settings.page.marginRightMm }}mm {{ settings.page.marginBottomMm }}mm
        {{ settings.page.marginLeftMm }}mm;
    }
  </style>
</head>
<body>
  <div>
    <p class="eyebrow">{{ common.tenant.displayName }}</p>
    <h1>{{ document.title }}</h1>
    <p>{{ common.labels.documentNumber }}: {{ document.number }} · {{ common.labels.documentDate }}: {{ document.date
      }}</p>
  </div>

  <div class="recipient">
    <strong>{{ common.recipient.displayName }}</strong>
    {% for line in common.recipient.addressLines %}<div>{{ line }}</div>{% endfor %}
  </div>

  <table class="items">
    <thead><tr><th>{{ common.labels.position }}</th><th>{{ common.labels.description }}</th><th>{{ common.labels.
      quantity }}</th><th>{{ common.labels.total }}</th></tr></thead>
    <tbody>
      {% for item in document.items %}
        <tr><td>{{ item.displayPosition }}</td><td>{{ item.descriptionXhtml | raw }}</td><td>{{ item.quantityWithUnit }}</
          td><td>{{ item.lineTotal }}</td></tr>
        {% endfor %}
      </tbody>
    </table>

    {% if components.financialSummary.available %}{% include "invoicer/components/v1/financial-summary.peb" %}{% endif %}
  </body>
</html>
```

`common.resources.fontCss` ist von der Anwendung vorbereitetes, sicheres CSS für die gepflegten eingebetteten Schriften und alle validierten paketlokalen Schriftvarianten. Binde es vor den Paketstilen ein. `descriptionXhtml` ist von der Anwendung vorbereitetes, sicheres XHTML. Deshalb werden nur diese ausdrücklich dafür vorgesehenen Felder mit dem unterstützten Filter `raw` ausgegeben. Geldbeträge, Umsatzsteuergruppen, Summen, Beschriftungen, Datumsangaben und Mengen liegen bereits als vorbereitete Anzeigewerte vor. Berechne sie nicht in Pebble neu.

Der reservierte Include für die Finanzübersicht liest `components.financialSummary` und akzeptiert keine Argumente. Sein stabiler äußerer Hook und seine semantischen Zeilen-Hooks können mit CSS gestaltet werden. Wähle eine Rolle mit

[data-row-role="vat_bucket_tax"] oder .inv-c-role-vat_bucket_tax; leite die Bedeutung einer Zeile nicht aus data-row-id ab. Siehe anwendungseigene Komponenten.

Die vorbereiteten Zeilen der Finanzübersicht sind für die PDF-Darstellung maßgeblich. Wenn `document.showVatRateColumn` bei einer geeigneten homogenen Nullsteuerregel `false` ist, lässt die Anwendung die redundanten Netto- und Null-Umsatzsteuerzeilen weg und liefert eine neutrale Zeile **Gesamtbetrag**; Eingangsgrößen der Rabattberechnung bleiben erhalten. Strukturierte E-Rechnungssummen und Steuerdaten ändern sich dadurch nicht.

3. Stile hinzufügen

Kopiere `styles.css`. Das Beispiel verwendet die deklarierten Seitenränder und gestaltet die Komponente über [data-invoicer-component="financial-summary"].

4. Import-ZIP erstellen und prüfen

Erstelle das ZIP aus dem Paketverzeichnis heraus, sodass `manifest.json`, `document.peb` und `styles.css` direkt im Stamm des Archivs liegen. Packe nicht das umgebende Verzeichnis ein. Komprimierung, Eintragsreihenfolge, Zeitstempel und Berechtigungen des ZIPs beeinflussen die unveränderliche Revision nicht. Die Anwendung hasht die validierten logischen Inhalte, wie unter ZIP-Import und Revisionsidentität beschrieben.

Mit dem macOS Finder erstellte ZIPs werden akzeptiert. Der Import ignoriert Verzeichniseinträge mit null Bytes und bekannte Finder-Metadaten (`__MACOSX`, `.DS_Store` und `._`-Dateien), nachdem deren begrenzte Pfade validiert wurden. Ein Verzeichniseintrag wie `images/` ist nur Container-Metadaten: deklarierte Dateien darunter, etwa `images/brand.png`, bleiben Teil des Pakets und werden normal validiert.

Importiere das ZIP anschließend im Editor für Layoutvorlagen. Beim Entwickeln meldet das strikte Rendering fehlende Variablen als `PRESENTATION_INCOMPATIBLE`, fehlende Dateien als `RESOURCE_MISSING` und nicht unterstützte Quellkonstrukte als `PEBBLE_FEATURE_FORBIDDEN`. Die vollständige Liste steht unter Diagnosen.

Für paketlokales Branding deklarierst Du jede PNG- oder JPEG-Datei unter `resources.images`, legst die exakte Datei in das ZIP und referenzierst ihren normalisierten paketrelativen Pfad aus XHTML oder CSS, zum Beispiel `` oder `background-image: url('images/brand.png')`. Inline-Ressourcen mit `data:` sowie Datei- oder Netzwerk-URLs gehören nicht zur Autorenoberfläche.

Deklariere jede paketlokale statische TrueType-Schriftvariante ausdrücklich:

```
"fonts": [
  {"path": "fonts/Acme-Regular.ttf", "family": "Acme", "weight": 400, "style": "normal"},
  {"path": "fonts/Acme-Bold.ttf", "family": "Acme", "weight": 700, "style": "normal"},
  {"path": "fonts/Acme-Italic.ttf", "family": "Acme", "weight": 400, "style": "italic"}
]
```

Wähle anschließend in Deinem Paket-CSS `font-family: 'Acme', 'IBM Plex Sans', sans-serif` und die deklarierte Schriftstärke bzw. den deklarierten Stil. Familie, Stärke und Stil müssen mit den internen Metadaten übereinstimmen. Ein Paket darf höchstens acht Varianten enthalten, jeweils höchstens 1 MiB und zusammen höchstens 4 MiB. OTF, TTC, WOFF/WOFF2, variable, farbige, Bitmap- und Type-1-Schriften sowie Betriebssystem-, lokale Datei- und Netzwerkschriften werden nicht unterstützt. Die importierende Organisation muss die Schriftsoftware einbetten und weitergeben dürfen. Die professionellen Starter zeigen vollständige Beispiele für Verkaufsdokumente und Zeitrachweise. Dieses einfache Beispiel lässt `fonts` bewusst leer und demonstriert den gepflegten IBM-Plex-Fallback.

Das Repository prüft genau diese Quelle mit englischen und deutschen Testdaten. Erfolgreiches Rendering belegt Vertragskompatibilität, aber keine rechtliche oder visuelle Freigabe eines Kundenlayouts.

Wenn Du mit einem vollständigeren Design beginnen möchtest, verwende das professionelle Starterpaket für Verkaufsdokumente. Sein lokalisierter Seitenzähler steht oben rechts auf der bisherigen Höhe und ist an der linken Textkante der wiederholten rechten Informationsspalte ausgerichtet. Das bewusst kleine Beispiel dieser Anleitung bleibt zum Erlernen des Mindestvertrags nützlich, ist aber nicht das fertige Starterpaket.

Ein Paket für Zeitrachweise erstellen

Ein Zeitrachweispaket verwendet dieselben Regeln für ZIP, Manifest, Ressourcen, Validierung, unveränderliche Revisionen und Client/Host wie ein Paket für Verkaufsdokumente. Sein Manifest deklariert `templateKind: timesheet`, eine

leere Liste `supportedDocumentFamilies` und einen Einstiegspunkt wie `timesheet.peb`.

Die vollständige, direkt anpassbare Quelle ist das professionelle Starterpaket für Zeitrachweise. Packe den Inhalt dieses Verzeichnisses so, dass `manifest.json` im Stamm des Archivs liegt, oder verwende das direkt importierbare ZIP.

Mit den gepflegten englischen Testdaten sieht das gerenderte Starterpaket so aus:

Page 1 of 1



FROM

Example Engineering Ltd.
Morgan Example
Customer Street
54321 Customer City
DE

TIMESHEET

timesheet

Aug 1, 2026 – Aug 31, 2026

Invoice Number

TS-BASE-EN-001

Billing Period

Aug 1, 2026 - Aug 31, 2026

Project

Project Timesheet

Deterministic Template API baseline fixture

DATE	TIME	DESCRIPTION	DURATION
Aug 1, 2026	09:00 - 10:30	Consulting Architecture workshop	1.50h
Aug 2, 2026	09:00 - 10:30	Consulting Internal project coordination	1.50h
Billable Hours			1.50h
Non-Billable Hours			1.50h
Gross Amount			3.00h

BILLING PERIOD

Aug 1, 2026
Aug 31, 2026

Project
TS-BASE-EN-001

TechVision
Solutions GmbH

Torsten Test

CONTACT

Tel:
+49 123 456789

Contact:
info@test.com

Web:
https://test.com



Artspezifische Präsentation

Beim Rendern ist genau eine artbezogene Wurzel befüllt. Eine Zeitrachweisvorlage liest `timesheet`; `document` ist `null`. Verwende die vorbereiteten Felder, statt Regeln für Zeitraum, Dauer, Lokalisierung oder abrechenbare Summen nachzubauen:

- `timesheet.documentTitle`, `number`, `periodStart` und `periodEnd` identifizieren den eingefrorenen Zeitraum;

- `timesheet.metadata` enthält vorbereitete, beschriftete Metadaten;
- `timesheet.entries` enthält vorbereitete Datumsangaben, optionale Zeitspannen, sicheres Beschreibungs-XHTML, optionale Kategorien und Orte, formatierte Dauern und den Abrechnungsstatus;
- `timesheet.hasLocations` zeigt an, ob die Ortsspalte sinnvoll ist;
- `invoicer/components/v1/timesheet-summary.peb` rendert die von der Anwendung verantworteten vorbereiteten Summen.

Null-Prüfungen sind Teil des Autorenvertrags. Zeitspanne, Kategorie, Ort, Beschreibung, Referenz, Logo oder Fußzeilenbeschriftung können fehlen. Listen sind immer vorhanden, dürfen aber leer sein.

Abgestimmtes Seitendesign

Das Starterpaket teilt die paketlokalen Varianten IBM Plex Serif Regular, Bold und Italic, die ausschließlich für den Titel eingesetzte Bebas Neue Regular, Farben, Logo-Behandlung und die wiederholte rechte Informationsspalte mit dem Starter für Verkaufsdokumente. Vor den vertikalen Unternehmensangaben stehen vorbereitete Zeitraum- und optionale Projektinformationen. Der Seitenzähler erscheint in der laufenden Kopfzeile. Eine konventionelle untere Fußzeile wird bewusst nicht ausgegeben.

Die paketlokale Signaturgrafik ist ein fiktives visuelles Beispiel mit dem herkömmlichen deutschen Platzhalternamen `M. Mustermann`. Ersetze oder entferne sie vor der Verwendung. Das Mandantenlogo ist davon zu unterscheiden: Es kommt aus `common.resources.tenantLogo`, wird von der Anwendung bereitgestellt und kann `null` sein.

Import und Verantwortung

Prüfe das Paket lokal, kontrolliere die akzeptierten Schriftvarianten, bestätige, dass die importierende Organisation die bereitgestellte Schriftsoftware einbetten und weitergeben darf, und importiere es im Editor für Zeitrachweis-Layoutvorlagen. Im Client-Modus wiederholt der Host Validierung und Installation. Lokale Vorschauen laden exakte, verifizierte Paketbytes in den datensatzbezogenen Cache. Eingefrorene Zeiträume behalten den Verweis auf ihre vorgesehene unveränderliche Revision.

Erfolgreicher Import und erfolgreiches Rendering belegen nur die Vertragskompatibilität. Das importierende Unternehmen verantwortet die visuelle, textliche und rechtliche Eignung des erzeugten Zeitrachweises.