

■ // E-RECHNUNG · PFLICHT AB 2027 · OPEN SOURCE

E-Rechnung selbst gebaut — samt Archiv, das dem Finanzamt standhält

Deine PDF-Rechnung zählt im deutschen B2B seit 2025 offiziell nicht mehr als E-Rechnung, und ab Januar 2027 wird das Ausstellen für viele Firmen Pflicht. Dieser Guide zeigt den kompletten Weg mit Open Source: den Generator, den deine KI gegen die offiziellen Prüfregeln baut, den Prüf-Loop bis zum grünen Report, und die drei Einstellungen, die dein Beleg-Archiv GoBD-tauglich machen.

FRISTEN 2025–2028

MUSTANGPROJECT

KOSIT-VALIDATOR

BAU-PROMPT

PAPERLESS-NGX + GOBD

Worum es geht: Eine E-Rechnung ist seit dem 1. Januar 2025 per Definition ein **strukturierter Datensatz nach EN 16931** — also XRechnung oder ZUGFeRD. Ein PDF ist nur noch eine „sonstige Rechnung“. Die gute Nachricht: Sowohl die Software, die solche Rechnungen erzeugt, als auch die Prüfsoftware, mit der der Standard offiziell abgenommen wird, liegen frei auf GitHub. Deine KI verbindet beide zu einem Generator, der sich selbst kontrolliert.

1 Die Rechtslage in 90 Sekunden

Grundlage ist das Wachstumschancengesetz vom März 2024. Die Pflicht kommt in drei Stufen:

AB WANN	WER	WAS GILT
01.01.2025	alle B2B-Unternehmen	E-Rechnungen müssen empfangen werden können; PDF gilt nur noch als „sonstige Rechnung“
01.01.2027	Vorjahresumsatz über 800.000 €	E-Rechnungen müssen ausgestellt werden
01.01.2028	alle B2B-Unternehmen	Ausstellungspflicht für alle

Zwei Formate erfüllen die Norm: **XRechnung** ist reines XML für Maschinen. **ZUGFeRD** (ab Version 2.x) ist ein PDF mit eingebettetem XML — dein Kunde sieht eine normale Rechnung, sein System liest die Daten. Für die meisten Selbstständigen ist ZUGFeRD deshalb der angenehmere Start.

2 Die zwei Repos, die alles abdecken

REPO	ROLLE	ECKDATEN
ZUGFeRD/mustangproject	Erzeugt und prüft ZUGFeRD, XRechnung und Factur-X; Bibliothek plus fertiges CLI	461 Sterne · Apache-2.0 · Java
itplr-kosit/validator	Die Prüfsoftware der Koordinierungsstelle für IT-Standards — mit ihr wird der Standard offiziell abgenommen	178 Sterne · Apache-2.0 · Java
+ validator-configuration-xrechnung	Die kompletten XRechnung-Prüfregeln als Konfiguration für den Validator	112 Sterne · Apache-2.0

WARUM DIESE KOMBINATION

Mustangproject baut die Rechnung, der KoSIT-Validator prüft sie mit denselben Regeln, mit denen auch Behörden-Systeme validieren. Dein Generator rät also nicht, ob er konform ist — er bekommt es bescheinigt. Für PHP-Projekte gibt es als Alternative [horstoeko/zugferd](#) (434 Sterne, MIT).

3 Der Bau-Prompt für deine KI

Kopiere den Prompt in Claude Code oder ein vergleichbares Coding-Tool. Er enthält die beiden Repos, den Prüf-Loop und die zwei Regeln, die dir später Ärger ersparen:

```
Baue mir einen E-Rechnungs-Generator als lokales CLI-Tool.

# Kontext
Deutsche B2B-Pflicht, Format EN 16931. Nutze Mustangproject
(github.com/ZUGFeRD/mustangproject) als Bibliothek oder über dessen CLI,
um aus meinen Rechnungsdaten ein ZUGFeRD-PDF (PDF mit eingebettetem XML)
und wahlweise eine XRechnung (reines XML) zu erzeugen.

# Eingabe
Meine Rechnungsdaten liegen in einer rechnung.yaml: Absender mit
USt-ID und IBAN, Empfänger, Rechnungsnummer, Datum, Positionen mit
Menge, Einzelpreis netto und Steuersatz. Lege mir eine Beispiel-Datei an.

# Prüf-Loop (wichtigster Teil)
Installiere den KoSIT-Validator (github.com/itplr-kosit/validator) mit
der XRechnung-Konfiguration (github.com/itplr-kosit/
validator-configuration-xrechnung). Nach jeder erzeugten Rechnung:
Validator ausführen, den Report auswerten, bei Fehlern deinen Code
korrigieren und erneut prüfen – so lange, bis der Report fehlerfrei ist.
Zeig mir am Ende den Report als Beleg.

# Regeln
1. Rechnungsnummern niemals doppelt vergeben – führe einen Zähler.
2. Fehlt dir ein Pflichtfeld, frag mich. Rate keine Firmendaten.
```

4 Grün heißt konform: den Report lesen

Der Validator erzeugt zu jeder Rechnung einen Prüfbericht. Drei Dinge solltest du darin einordnen können:

- **Schema-Fehler** bedeuten: Das XML ist strukturell kaputt — ein Pflichtfeld fehlt oder steht am falschen Ort. Das behebt die KI im Loop selbst.
- **Schematron-Meldungen** sind die Geschäftsregeln der XRechnung: Steuersummen müssen aufgehen, die IBAN muss zur Zahlungsart passen. Auch das prüft der Loop automatisch.
- **Warnungen ohne Fehler** sind erlaubt. Erst wenn der Report keine Fehler mehr enthält, gilt die Rechnung als konform zur Prüfkongfiguration.

Heb die Reports auf. Ein Ordner mit Rechnung plus zugehörigem Prüfbericht ist im Zweifel dein Nachweis, dass du sauber gearbeitet hast.

5 Das Archiv: Paperless-ngx GoBD-tauglich machen

Erzeugen ist die halbe Pflicht — aufbewahren die andere. Buchungsbelege müssen in Deutschland **8 Jahre** unveränderbar aufbewahrt werden. Das größte offene Archiv-System dafür ist **Paperless-ngx** (45.174 Sterne, GPL-3.0): Es scannt, erkennt Text, verschlagwortet und findet jeden Beleg in Sekunden.

AB WERK REICHT ES NICHT

Frisch installiert ist Paperless-ngx ausdrücklich nicht revisionssicher: Nutzer mit Rechten können Dokumente endgültig löschen, es gibt keinen Schreibschutz, und das Audit-Log ist nicht aktiv. Das Projekt selbst sagt das offen in seinen GitHub-Discussions (#1880, #10379).

GoBD-tauglich wird das Setup durch drei Maßnahmen:

- 1 **Unveränderbarkeit über den Speicher lösen:** Originale zusätzlich in einen S3-kompatiblen Object-Storage mit **Object Lock im Compliance-Modus** spiegeln. Dort kann bis zum Fristablauf niemand löschen oder überschreiben — auch kein Administrator.
- 2 **Audit-Logging aktivieren und Löschrechte entziehen:** In Paperless das Audit-Log einschalten und normalen Nutzern das Löschen über die Gruppenrechte nehmen. Jeder Zugriff wird damit nachvollziehbar.
- 3 **Verfahrensdokumentation schreiben lassen:** Das Dokument, das beschreibt, wie deine Belege vom Eingang bis zur Löschung behandelt werden — Pflicht laut GoBD, und der Teil, vor dem sich alle drücken. Genau hier hilft die KI:

Lies meine Paperless-ngx-Konfiguration (docker-compose.yml, Umgebungsvariablen, Benutzer und Gruppen) und schreib daraus eine GoBD-Verfahrensdokumentation als Markdown: Systembeschreibung, Rollen und Rechte, Ablauf vom Beleg-Eingang bis zur Archivierung, Unveränderbarkeits-Mechanik (Object-Lock-Einstellungen), Protokollierung, Aufbewahrungsfristen, Backup und Notfall-Wiederherstellung.

Regel: Markiere jede Stelle, an der dir Information fehlt, als [OFFEN: ...] statt sie zu erfinden. Ich fülle die Lücken selbst.

6 Quellen

- **BMF-FAQ zur E-Rechnung:** [bundesfinanzministerium.de/Content/DE/FAQ/e-rechnung.html](https://www.bundesfinanzministerium.de/Content/DE/FAQ/e-rechnung.html)
- **Generator:** github.com/ZUGFeRD/mustangproject
- **Offizielle Prüfung:** github.com/itplr-kosit/validator + github.com/itplr-kosit/validator-configuration-xrechnung
- **Archiv:** github.com/paperless-ngx/paperless-ngx (GoBD-Diskussionen: #1880, #10379)
- **PHP-Alternative:** github.com/horstoekeo/zugferd

■ // AUF EINEN BLICK

Die Kurzfassung

Fünf Punkte von der ersten E-Rechnung bis zum Archiv.

- ✓ Fristen kennen: Empfang seit **2025**, Ausstellen ab **2027** (über 800.000 €), ab **2028** alle
- ✓ Generator mit dem **Bau-Prompt aus Abschnitt 3** bauen lassen — Mustangproject erzeugt, KoSIT prüft
- ✓ Jede Rechnung erst verschicken, wenn der **Validator-Report fehlerfrei** ist — Reports aufheben
- ✓ Archiv mit Paperless-ngx plus **Object Lock, Audit-Log und entzogenen Löschrechten** aufsetzen
- ✓ **Verfahrensdokumentation** von der KI aus der eigenen Konfiguration erstellen lassen, Lücken selbst füllen

■ // BLEIB DRAN

KI-Setups, die echte Pflichten erledigen.

Ich zerlege solche Themen in Abläufe, die du an einem Nachmittag umsetzt — mit den Repos, den Prompts und den Grenzen, die dir sonst niemand dazusagt. Folg mir für den nächsten Guide.

@ki.borish

Stand 16.09.2026. Fristen laut Wachstumschancengesetz und BMF-FAQ; Sterne-Zahlen von GitHub am selben Tag. Dieser Guide ist eine technische Anleitung und ersetzt keine Rechts- oder Steuerberatung: GoBD-Konformität ist immer eine Eigenschaft deines gesamten Prozesses samt Verfahrensdokumentation, nie einer Software allein. Der KoSIT-Validator prüft die Format-Konformität nach EN 16931/XRechnung, keine steuerliche Richtigkeit. Sprich Aufbewahrung und Verfahrensdokumentation im Zweifel mit deiner Steuerberatung durch.