

# Nuclet: Elektronische Rechnungen empfangen - XRechnung und ZUGFeRD

 Sprache: [Deutsch](#) · [English](#)

## Nuclet: Elektronische Rechnungen empfangen - XRechnung und ZUGFeRD

Nuclet: Elektronische Rechnungen empfangen - XRechnung und ZUGFeRD – Referenz zum Nuclet.

REFERENZ

INTEGRATOREN

STAND: JUL 2026

GILT FUER NUCLOS 4.2026.X

### Auf dieser Seite

- [Releases](#)
  - [Überblick](#)
  - [Hinweise](#)
  - [Integration](#)
  - [Verwandte Seiten](#)
- 
- Nuclet: Elektronische Rechnungen empfangen - XRechnung und ZUGFeRD
    - [Releases](#)
    - [Überblick](#)
    - [Hinweise](#)
    - [Integration](#)
    - [Verwandte Seiten](#)

### Releases

| Version | Datum      | Hinweise                                                                                             | Kompatibilität        | Bemerkung |
|---------|------------|------------------------------------------------------------------------------------------------------|-----------------------|-----------|
| 2.1.2   | 03.12.2024 | Erste veröffentlichte Fassung                                                                        | ab Nuclos 4.2022.41.2 |           |
| 2.1.3   | 09.01.2025 | Kleinere Layoutänderungen und verschoben in Menü <i>Schnittstellen</i><br>Anpassung an Exportversion | ab Nuclos 4.2022.41.2 |           |
| 2.1.4   | 07.02.2025 | Eigenständigen Import ohne <i>Import E-Rechnung</i> ermöglicht                                       | ab Nuclos 4.2022.41.2 |           |
| 2.1.5   | 29.07.2025 | Anpassung an Exportversion                                                                           | ab Nuclos 4.2022.41.2 |           |
| 2.1.6   | 10.10.2025 | Anpassung an Exportversion                                                                           | ab Nuclos 4.2022.41.2 |           |

### Überblick

Das Nuclet ermöglicht den automatisierten Empfang elektronischer Rechnungen (E-Rechnungen) im XML- oder PDF-Format, deren offizielle [Validierung](#) und [Darstellung](#) in einer HTML-Datei sowie die Vorbereitung zur Umwandlung in ein geeignetes Businessobjekt (im Weiteren: Eingangsrechnung).

Unterstützt wir das Format [XRechnung](#) in der Version 3.0.2 (20.06.2024) in der XML-Syntax [Universal Business Language](#) (UBL) der OASIS in der Version 2.1 und der XML-Syntax [Cross Industry Invoice](#) (CII) der UN/CEFACT in der Version D16B.  
Weiterhin wird die Spezifikation [ZUGFeRD](#) mit ihren fünf Profilen in der Version 2.3.2 (15.11.2024) abgedeckt.

Der Empfang elektronischer Rechnung in veralteten Versionen der genannten Formate ist in vielen Fällen ebenfalls möglich, resultiert aber meist in zusätzlichen Validierungsmeldungen.

E-Rechnungen können über den Job *Importiere ERechnungen* automatisch aus einem vorgegebenen Verzeichnis (Nuclet-Parameter *Ablagepfad*) importiert werden.

- Dabei wird ein Businessobjekt *ERechnungsimport* mit entsprechenden Meldungen erzeugt. Danach erfolgt - falls möglich - die Erstellung einer Eingangsrechnung aus den Daten.
- Die Dateien werden anschließend in entsprechende Unterordner verschoben, damit sie nicht mehrfach verarbeitet werden.

E-Rechnungen können außerdem manuell über die Maske *Schnittstellen* -> *Import E-Rechnung* importiert und in eine Eingangsrechnung überführt werden.

Das Beispiel-Nuclet unter [Nuclet: Elektronische Rechnungen - XRechnung und ZUGFeRD](#) kann auch hier verwendet werden.

## Hinweise

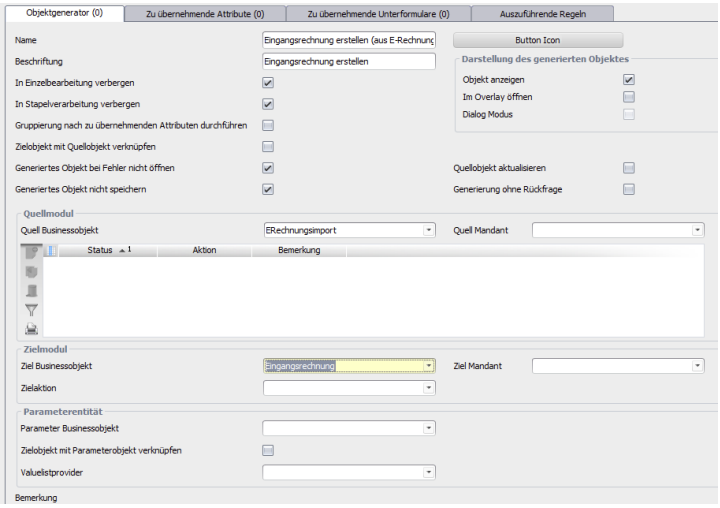


### Abhängigkeiten

Das Nuclet benötigt die beiden unter [Nuclet: Elektronische Rechnungen - XRechnung und ZUGFeRD](#) beschriebenen Nuclets.

1. Zur Erstellung von Eingangsrechnungen muss eine projektspezifische Objektgeneratorregel erstellt werden, die die Zuordnung zu den Feldern der Eingangsrechnung übernimmt. An dieser Stelle kann auch das Interface `org.nuclet.xrechnung.imp.ImportiereERrechnungen.InvoiceHelper` implementiert werden (die Zuordnung erfolgt über den Nuclet-Parameter `ImplKlasseEingangsrechnung`), das die im Job benötigten Methoden zur Validierung der Eindeutigkeit (`getEingangsrechnung`), Erstellung einer neuen Eingangsrechnung (`newEingangsrechnung`) und Zuordnung der Felder (`setFields`) bereitstellt. Ein vollständiges Beispiel ist unter [Integration](#) genannt.
2. Über die Methode `org.nuclet.xrechnung.imp.AktualisiereERrechnungsimport#getXRechnung` wird das allgemeine Java-Objekt auf Basis des XRechnungs-Standards zur Verfügung gestellt. Falls Felder benötigt werden, die nur in einer der beiden o.g. XML-Syntaxen vorhanden sind, können die jeweils vollständigen Java-Transformationen über `org.nuclet.xrechnung.imp.AktualisiereERrechnungsimport#getCompleteInvoice` ermittelt werden.

## Integration

| Schritt | Beschreibung                                                                                                                                                                                                                                                                 | Screenshot / Beschreibung                                                                                                                                                        |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1       | Falls noch nicht erfolgt, die beiden unter <a href="#">Nuclet: Elektronische Rechnungen - XRechnung und ZUGFeRD</a> beschriebenen Nuclets importieren (siehe <a href="#">Nuclet Import</a> ).<br><br>Nach Import dieser Nuclets muss die Serverinstanz neu gestartet werden. |                                                                                                                                                                                  |
| 2       | Import des Nuclets durchführen, anschließend muss erneut die Serverinstanz neu gestartet werden, da das Nuclet eigene Extensions enthält.                                                                                                                                    |                                                                                                                                                                                  |
| 3       | Einen Objektgenerator für die Eingangsrechnungen mit entsprechender Regel erstellen.                                                                                                                                                                                         | <br><b>ErstelleEingangsrechnung.java</b><br><pre>package org.nuclet.beispielxrechnung;</pre> |

```

import eu.ce.en16931._2017.xoev_de.kosit.standard.xrechnung_1.*;
import org.nuclet.xrechnung.imp.AktualisiereERechnungsimport;
import org.nuclet.xrechnung.imp.ERechnungsimport;
import org.nuclet.xrechnung.imp.ImportiereERechnungen;
import org.nuclos.api.businessobject.Flag;
import org.nuclos.api.businessobject.facade.Modifiable;
import org.nuclos.api.provider.QueryProvider;
import org.nuclos.api.rule.GenerateRule;
import org.nuclos.api.context.GenerateContext;
import org.nuclos.api.annotation.Rule;
import org.nuclos.api.exception.BusinessException;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

import static org.nuclet.xrechnung.imp.ImportiereERechnungen.*;
import static org.nuclet.xrechnung.imp.ImportiereERechnungen.getTextValue;

/** @name
 * @description
 * @usage
 * @change
 */
@Rule(name="ErstelleEingangsrechnung", description="
ErstelleEingangsrechnung")
public class ErstelleEingangsrechnung implements GenerateRule {

    public void generate(GenerateContext context) throws
BusinessException {
        ERechnungsimport imp = context.getSourceObject
(ERechnungsimport.class);
        Rechnung rechnung = context.getTargetObject(Rechnung.class);
        setFields(imp, rechnung);
    }

    public static void setFields(ERechnungsimport imp, Rechnung
rechnung) throws BusinessException {
        AktualisiereERechnungsimport.checkUpdate(imp, "Zu diesem
Import wurde bereits eine Eingangsrechnung erstellt.");
        Invoice xRechnung = AktualisiereERechnungsimport.getXRechnung
(imp);

        rechnung.setERechnungsimportId(imp.getId());
        rechnung.setRechnungsnummer(xRechnung.getInvoiceNumber().
getValue());
        rechnung.setRechnungsdatum(getDate(xRechnung.
getInvoiceIssueDate()));
        rechnung.setCoderechnungsart(getCodeValue(xRechnung.
getInvoiceTypeCode()));
        rechnung.setFaelligkeit(getDate(xRechnung.
getPaymentDueDate()));
        DocumentReference purchaseOrderReference = xRechnung.
getPurchaseOrderReference();
        if (purchaseOrderReference != null) {
            rechnung.setBestellnummer(purchaseOrderReference.
getValue());
        }
        rechnung.setZahlungskonditionen(getTextValue(xRechnung.
getPaymentTerms()));
        SELLERType seller = xRechnung.getSELLER();
        if (seller != null) {
            SELLERCONTACTType sellerContact = seller.
getSELLERCONTACT();
            if (sellerContact != null) {
                rechnung.setAnsprechpartnername(getTextValue
(sellerContact.getSellerContactPoint()));
                rechnung.setAnsprechpartnertelefon
(getTextValue(sellerContact.getSellerContactTelephoneNumber()));
                rechnung.setAnsprechpartneradresse
(getTextValue(sellerContact.getSellerContactEmailAddress()));
            }
        }
    }
}

```

```

    }
    String ustIdNr = getIdValue(seller.
getSellerVATIdentifier());
    if (ustIdNr != null) {
        List<Kunde> kundeList = QueryProvider.execute
(QueryProvider.create(Kunde.class).where(Kunde.Ustidnr.eq(ustIdNr)).
setChunkSize(2L));

        int size = kundeList.size();
        Long kundeId = null;
        if (size == 1) {
            kundeId = kundeList.iterator().next().getId();
        } else if (size == 0) {
            Kunde kunde = new Kunde();
            kunde.setName(getTextValue(seller.
getSellerName()));
            kunde.setLeitweg(getTextValue
(xRechnung.getBuyerReference()));
            SELLERPOSTALADDRESSType
sellerpostaladdress = seller.getSELLERPOSTALADDRESS();
            if (sellerpostaladdress != null) {
                kunde.setStrasse(getTextValue
(sellerpostaladdress.getSellerAddressLine1()));
                kunde.setPlz(getTextValue
(sellerpostaladdress.getSellerPostCode()));
                kunde.setOrt(getTextValue
(sellerpostaladdress.getSellerCity()));
                kunde.setLaendercode
(getCodeValue(sellerpostaladdress.getSellerCountryCode()));
            }
            kunde.setUstidnr(ustIdNr);
            kunde.save();
            kundeId = kunde.getId();
        }
        if (kundeId != null) {
            rechnung.setKundenreferenzId
(kundeId);
        }
    }
    rechnung.setWaehrung(getCodeValue(xRechnung.
getInvoiceCurrencyCode()));
    List<VATBREAKDOWNType> vatbreakdown = xRechnung.
getVATBREAKDOWN();
    if (vatbreakdown.size() == 1) {
        VATBREAKDOWNType breakdown = vatbreakdown.iterator().
next();
        rechnung.setSteuercode(getCodeValue(breakdown.
getVATCategoryCode()));
        rechnung.setSteuersatz(getBigDecimal(breakdown.
getVATCategoryRate()));
    }
    for (INVOICELINEType line : xRechnung.getINVOICELINE()) {
        Rechnungsposition pos = new Rechnungsposition();
        ITEMINFORMATIONType iteminformation = line.
getITEMINFORMATION();
        if (iteminformation != null) {
            pos.setBezeichnung(getTextValue
(iteminformation.getItemName()));
            pos.setMenge(getBigDecimal(line.
getInvoicedQuantity()));
            pos.setEinheit(getCodeValue(line.
getInvoicedQuantityUnitOfMeasureCode()));
            pos.setNettobetrag(getBigDecimal(line.
getInvoiceLineNetAmount()));
            pos.setBestellposition(getIdValue(line.
getInvoiceLineIdentifier()));
            INVOICELINEPERIODType period = line.
getINVOICELINEPERIOD();
            if (period != null) {
                pos.setLeistungszeitraumvon(getDate(period.

```

```

getInvoiceLinePeriodStartDate()));
        pos.setLeistungszeitraumbis(getDate(period.
getInvoiceLinePeriodEndDate()));
    }
    rechnung.insertRechnungsposition(pos);
}

public static class InvoiceHelperImpl implements InvoiceHelper {

    public Modifiable<?> newEingangsrechnung() {
        return new Rechnung();
    }

    public List<? extends Modifiable<?>> getEingangsrechnung
(ERrechnungsimport imp, Flag... flags) {
        List<Rechnung> list = new ArrayList<>();
        org.nuclet.beispielxrechnung.ERrechnungsimport
rechnungsimport = org.nuclet.beispielxrechnung.ERrechnungsimport.get(imp.
getId());
        if (rechnungsimport != null) {
            list = rechnungsimport.getRechnung(flags);
        }
        return list;
    }

    public void setFields(ERrechnungsimport imp, Modifiable<?>
eingangsrechnung) throws BusinessException {
        ErstelleEingangsrechnung.setFields(imp, (Rechnung)
eingangsrechnung);
    }
}

```

4

Bei Bedarf einen Integrationspunkt im Nuclet der Eingangsrechnung mit Verweis auf den *ERechnungsimport* erstellen und ein Referenzfeld über den BO-Editor anlegen.

**Integrationspunkt**

Nuclet:  Fehler? ☐

Name:

Nur lesen? ☐

Optional? ☐

Mit Statusmodell? ☐

Ziel Nuclet:

Ziel Businessobjekt:

View Name:

Readme URL:

Hinweis zur Integration:

---

**Attribute (0) Fehler (0) Verwendungen (1)**

| Businessobjekt | Attribut         | Datentyp         | valueListProvider | Nachkommastellen | Feldbreite |
|----------------|------------------|------------------|-------------------|------------------|------------|
| Rechnung       | erechnungsimport | java.lang.String |                   |                  | 255        |

---

**Attribute bearbeiten**

Fügen Sie Attribute hinzu, ändern Sie vorhandene oder löschen Sie welche. Die Reihenfolge der Attribute wird in Listen und Unterformularen als Standard für die Spalteninstellungen verwendet.

| Anzeigename            | Beschreibung           | Datentyp         | Feldbreite | Nachkommastellen | Eindeutig? | Historie? | Pflichtfeld? | Feldname               | I.P. | Regel | JavaTyp | Berechnet | Defaultwert | Attributgruppe | Versteckt? | Veränderbar? | Mehrsprachig |
|------------------------|------------------------|------------------|------------|------------------|------------|-----------|--------------|------------------------|------|-------|---------|-----------|-------------|----------------|------------|--------------|--------------|
| Rechnungsnummer        | Rechnungsnummer        | Text             | 255        |                  |            |           |              | rechnungsnummer        |      |       | String  |           |             |                |            |              |              |
| Rechnungsdatum         | Rechnungsdatum         | Datum            | 255        |                  |            |           |              | rechnungsdatum         |      |       | Date    |           |             |                |            |              |              |
| Code Rechnungart       | Code Rechnungart       | Text             | 255        |                  |            |           |              | coderechnungsart       |      |       | String  |           |             |                |            |              |              |
| Fälligkeit             | Fälligkeit             | Datum            | 255        |                  |            |           |              | fälligkeits            |      |       | Date    |           |             |                |            |              |              |
| Bestellnummer          | Bestellnummer          | Text             | 255        |                  |            |           |              | bestellnummer          |      |       | String  |           |             |                |            |              |              |
| Zahlungskonditionen    | Zahlungskonditionen    | Text             | 255        |                  |            |           |              | zahlungskonditionen    |      |       | String  |           |             |                |            |              |              |
| Ansprechpartnername    | Ansprechpartnername    | Text             | 255        |                  |            |           |              | ansprechpartnername    |      |       | String  |           |             |                |            |              |              |
| Ansprechpartnertelefon | Ansprechpartnertelefon | Text             | 255        |                  |            |           |              | ansprechpartnertelefon |      |       | String  |           |             |                |            |              |              |
| Ansprechpartneradresse | Ansprechpartneradresse | Text             | 255        |                  |            |           |              | ansprechpartneradresse |      |       | String  |           |             |                |            |              |              |
| Währung                | Währung                | Text             | 255        |                  |            |           |              | währung                |      |       | String  |           |             |                |            |              |              |
| Kundenreferenz         | Kundenreferenz         | Referenzfeld     | 255        |                  |            |           |              | kundenreferenz         |      |       | String  |           |             |                |            |              |              |
| Steuernummer           | Steuernummer           | Text             | 255        |                  |            |           |              | steuernummer           |      |       | String  |           |             |                |            |              |              |
| Steuersatz             | Steuersatz             | Kommasatz (0..2) | 9          | 2                |            |           |              | steuersatz             |      |       | Double  |           |             |                |            |              |              |
| E-Rechnungsimport      | E-Rechnungsimport      | Referenzfeld     | 255        |                  |            |           |              | erechnungsimport       |      |       | String  |           |             |                |            |              |              |

---

**Verknüpfung zu anderen Businessobjekten**

Verknüpfung zu anderen Businessobjekten

---

Verweis auf Integrationspunkt:

Verweis auf Businessobjekt:

Kaskadierend löschen: ☐

Suchfeld: ☐

primaryKey  
nummer  
datumimport

>

5

Bei Bedarf das Layout *E Rechnungsimport* ergänzen.

- Unterformular für Eingangsrechnung einbinden
- Objektgenerator über Knopf einbinden (alternativ nur über Menü ohne Layoutänderung)

Name:  Bemerkung:

Layout:

Palette:

Name der Kompo...:

Schriftgröße:

Businessobjekt:

Übergeordnetes ...:

Fremdschlüssel:

Toolbar:

Statusbar: ☐

Bevorzugte Größe: Breite:  Höhe:

Erweiterte Eigen...:

Nächstes Feld:

Nächste Kompo...:

---

**Import E-Rechnung**

Nummer:

Datum (Import):

Job?: ☐ Job

Eingangsrechnung erstellen

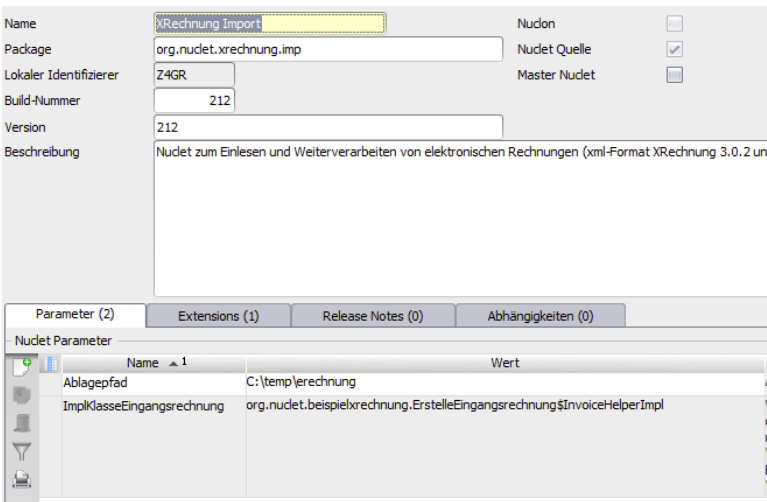
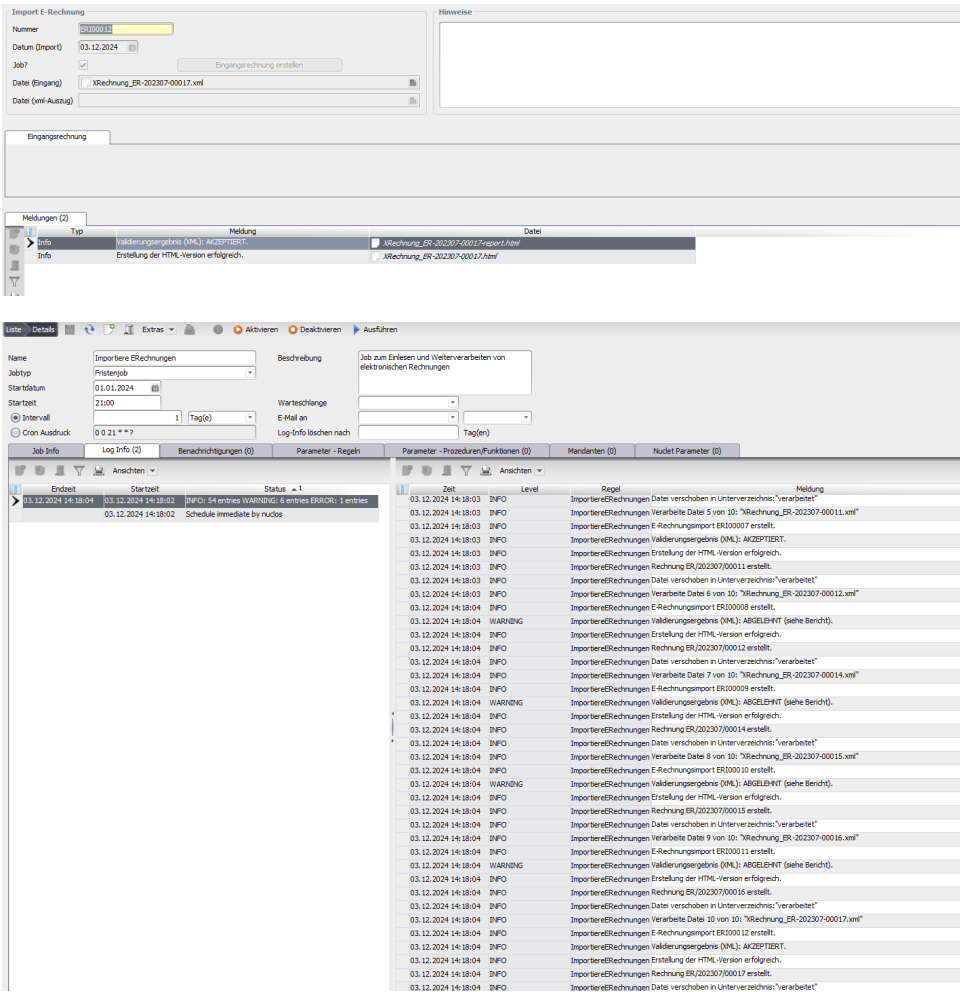
Datet (Eingang):

Datet (xml-Auszug):

---

**Eingangsrechnung**

Status:  Rechnungs... Code Rechn... Fälligkeit Bestellnummer Zahlungs... Ansprechpar... Ansprechpar...

|   |                                                                                                                                                                                                |                                                                                     |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| 6 | Die Nuclet-Parameter geeignet setzen.                                                                                                                                                          |   |
| 7 | Die Funktionsfähigkeit des Jobs und des manuellen Imports mit geeigneten Beispielen (etwa <a href="#">xrechnung-testsuite/.../technical-cases</a> oder <a href="#">zugferd-2.3.2</a> ) prüfen. |  |

## Verwandte Seiten

### Schnittstellen

Kategorie

[Öffnen](#)

### Nuclet Wiki

Alle Kategorien

[Öffnen](#)

