

# EKaBS

## Elektronischer Kassen-Beleg-Standard

Version 1.0.0 (Stand 14.04.2021)

---

Der Elektronische Kassen-Beleg-Standard, kurz EKaBS, soll zu einer allgemeinen Erleichterung bei der Handhabung führen und so den Interessen der Konsumenten, der Kassenanwender, der Kunden, der Kassensystembranche als auch den Finanzbehörden dienen.

Daher steht der EKaBS ausdrücklich über den Partikularinteressen einzelner und wurde in der Arbeitsgruppe „Elektronischer Beleg“ des DFKA e.V. und in Zusammenarbeit mit dem ZDH (Zentralverband des Deutschen Handwerks), HDE (Handelsverband Deutschland) und der DEHOGA (Deutscher Hotel- und Gaststättenverband) entwickelt.

# 1. Grundsätzliches

## 1.1. Rechtliche Grundlagen

Mit Wirkung zum 1.1.2020 ist die Belegausgabepflicht in [§ 146a Abs. 2 S. 1 AO](#) gesetzlich verankert worden. Das Anbieten einer Belegerstellung reicht nicht aus, der Beleg muss immer tatsächlich (und nachweisbar) erstellt und bereitgestellt werden. Diese fortwährende Bereitstellung von Belegen ist außerdem ein wichtiges Element für eine einfache Kassen-Nachschau.

In einigen Branchen, z.B. Bäckereien, wurden bisher nur selten Belege ausgegeben. Seit Ende 2019 gab es eine öffentliche Diskussion über die Belegausgabepflicht, da in der Praxis die gesetzlich vorgesehene Befreiung ([§ 146a Abs. 2 S. 2 AO](#)) durch eine enge Auslegung der Finanzverwaltung faktisch ins Leere läuft. Von Politik und Verwaltung werden als Lösung dafür die Erstellung von elektronischen Kassenbelegen favorisiert, weil damit gleichzeitig die Belegpflicht eingehalten und Papier eingespart werden kann.

In [§ 6 KassenSichV](#) sind elektronische Belege ausdrücklich in einem standardisierten Format erlaubt. Der [AEAO zu § 146a](#) regelt diverse Details.

Durch ein [BMF-Schreiben vom 28.5.2020](#) wird der Anwendungserlass um einige Klarstellungen ergänzt, welche die Anwendung elektronischer Kassenbelege erleichtern bzw. rechtssicherer machen (die Ergänzungen durch das neue Schreiben sind unterstrichen):

- 6.3: „Die Zustimmung des Kunden zur elektronischen Bereitstellung des Beleges bedarf dabei keiner besonderen Form und kann auch konkludent erfolgen.“
- 6.4: „Die Sichtbarmachung eines Beleges an einem Bildschirm des Unternehmers (Terminal/Kassendisplay) allein, ohne die Möglichkeit der elektronischen Entgegennahme nach Abschluss des Vorgangs, reicht nicht aus.“
- 6.6: „Es bestehen keine technischen Vorgaben wie der Beleg zur Entgegennahme bereitgestellt oder übermittelt werden muss. Es ist z. B. zulässig, wenn der Kunde unmittelbar über eine Bildschirmanzeige (z. B. in Form eines QR-Codes) den elektronischen Beleg entgegennehmen kann. Eine Übermittlung kann auch z. B. als Download-Link, per Near-Field-Communication (NFC), per E-Mail oder direkt in ein Kundenkonto erfolgen.“

Eine elektronische Belegausgabe muss in einem „standardisierten Datenformat“ (z. B. JPG, PNG oder PDF) erfolgen. Das bedeutet in diesem Zusammenhang allerdings nur, dass „der Empfang und die Sichtbarmachung eines elektronischen Kassenbeleges auf dem Endgerät des Kunden [...] mit einer kostenfreien Standardsoftware möglich sein“ müssen (Ziffer 6.6). Dadurch bleiben allerdings die von verschiedenen Systemen erzeugten Belege so unterschiedlich, dass sie nicht einheitlich verwaltet und automatisiert ausgewertet werden können.

Diese Lücke soll der vorliegende Standard schließen.

## 1.2. Grundanforderungen

Das Projekt soll die folgenden grundlegenden Anforderungen umsetzen:

- **Technologieoffenheit:** Das bedeutet (a) keine Vorgaben für Art der Implementierung und (b) große Freiheiten bei den „Transportwegen“ des Belegs zum Konsumenten.
- **Universelle Nutzbarkeit:** Die standardisierten elektronischen Kassenbelege sollen in allen Branchen und in möglichst vielen verschiedenen Systemen umsetzbar sein.
- **Implementierung in Stufen:** Dies soll einen möglichst schnellen Einsatz in der Praxis erlauben.
- **Über nötige Standardisierung hinaus nur Empfehlungen:** Die Standardisierung beschränkt sich auf das zur korrekten Funktion und zur Erfüllung der rechtlichen Anforderungen Erforderliche. Um den Standard so einfach wie möglich zu machen, werden darüber hinaus maximal Empfehlungen ausgesprochen.

Der Standard folgt den folgenden Grundprinzipien:

- Die elektronische Darstellung soll die gleichen Informationen enthalten wie die lesbare Form.
- Es ist eine automatisierte Verifikation der Daten der Technischen Sicherheitseinrichtung (TSE) möglich.
- Eine Verifikation ist per Auswertung des Security-Nodes ohne Einbeziehung anderer Daten möglich – bei Bedarf kann anschließend ein automatisierbarer Abgleich mit den anderen Beleginhalten erfolgen.
- Wo es möglich ist, soll eine weitgehende Anlehnung an die Taxonomie zur Vereinfachung von Definition, Dokumentation, Implementierung und Anwendung erfolgen.
- Momentan gilt eine Beschränkung auf die elektronische Darstellung von Kassenbelegen mit Rechnungscharakter. Es werden zurzeit keine anderen Belegtypen, wie zum Beispiel Lieferscheine, berücksichtigt – Erweiterungen sind grundsätzlich vorgesehen.
- Der Standard ist bewusst kein „Ersatz für die Taxonomie“, d.h.:
  - Alle Belege zusammen sollen keine komplette Dokumentation der Kassensführung sein.
  - Die Beleginhalte haben nicht die gleiche Informationstiefe wie die Taxonomie.

## 1.3. Lizenz

Vom DFKA e.V. als Lizenzgeber werden das EKaBS-JSON-Schema und die Dokumentation unter den Bedingungen der [Creative Commons-Lizenz CC BY-SA 4.0 \(Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International\)](#) zur Verfügung gestellt.

Daraus resultieren folgende [Rechte für den Nutzer](#): Das EKaBS-JSON-Schema und die Dokumentation dürfen vervielfältigt, weiterverbreitet, verändert und in jeder Form genutzt werden (auch kommerziell), solange der Hinweis auf den DFKA e.V. als Lizenzgeber erhalten bleibt und Modifikationen nur unter denselben Bedingungen weitergegeben werden.

## 1.4. Standardisierung in Ebenen

Um die gewünschte stufenweise Standardisierung und damit einen schnellen Praxiseinsatz zu erreichen, bietet sich die Aufteilung in die im Folgenden beschriebenen, aufeinander aufbauenden Ebenen an.

**Die vorliegende Version des Standards behandelt nur die Ebenen 1 bis 3.**

### 1.4.1. Ebene 1 – Inhalte elektronischer Kassenbelege

Welche Daten gehören in den elektronischen Kassenbeleg?

Dies ist durch die rechtlichen Vorgaben u.a. in § 6 KassenSichV bereits abschließend geklärt.

### 1.4.2. Ebene 2 – Format strukturierter Belegdaten

Wie ist der automatisch auswertbare Teil der Daten aufgebaut?

- Gewählte Lösung:
  - Das bedeutet eine Codierung des Beleginhaltes im JSON-Format
  - Baut auf der DFKA Taxonomie Kassendaten auf – gegenüber dieser gibt es sowohl Vereinfachungen als auch Ergänzungen
  - Die Abbildung erlaubt eine automatische Signaturprüfung
- Nicht favorisierte Alternativen:
  - XML-Struktur des ZUGFeRD-Standards – erheblich komplexer und nicht auf Kassenbelege ausgelegt
  - CSV-Version der Taxonomie (DSFinV-K) – für Art der abzubildenden Daten nicht gut geeignet

### 1.4.3. Ebene 3 – Verbindung lesbarer Beleg mit strukturierten Daten

Wie werden menschen- und maschinenlesbare Daten kombiniert?

- Gewählte Lösung:
  - Abbildung des Belegs als PDF-Datei
  - Einbindung der strukturierten Daten gemäß PDF/A-Standard
- Optionale Alternativen sind für Low-End-Systeme denkbar, z.B. Nutzung eines einfachen Text-Formats statt PDF
- Nicht favorisierte Alternativen:
  - ZUGFeRD – siehe oben
  - HTML – nicht gut geeignet, um mehrere Bestandteile des Belegs in einer Datei zusammenzufassen und um Belege dauerhaft zu archivieren

### 1.4.4. Ebene 4 – „Transportwege“ Kasse zum Empfänger

Wie bekommen die Kunden den Beleg?

- „Wiedererkennungseffekt“ bei den Nutzern fördert Akzeptanz
- Langfristiges Ziel: Jeder Kunde (ob anonym oder mit Kundenkonto) kann seine Belege in einem virtuellen Korb sammeln
- Möglichkeiten für ...

- anonyme Kunden: QR-Code mit WWW-Link auf Daten, Senden per NFC
- Kunden mit Kundenkonto: Anbieter muss Standard Web-API für „Belegverwaltungslösungen“ zur Verfügung stellen
- Hierbei ist auch die Standardisierung einer Schnittstelle zwischen Kassensystem und einem Belegerstellungs- und Verwaltungssystem eines Drittanbieters (das die eigentliche Belegerstellung und -bereitstellung übernimmt) denkbar
- Festlegung für die aktuelle Version des Standards:
  - Keine Vorgaben
  - Bei entsprechender Nachfrage Aufnahme in zukünftige Versionen

### 1.4.5. Ebene 5 – Elektronische Kassenbelege in der Buchführung

Wie verwendet man die Belege als Eingangsrechnung?

- Hierbei handelt es sich um eine naheliegende Ergänzung
- Anwendung eher langfristig interessant nach Etablierung des Standards
- Aspekte:
  - Verifikation
  - Automatische Verbuchung

### 1.4.6. Ebene 6 – Übergreifende Zusammenführung von Belegen

Wie kann man Belege verschiedener Unternehmen sammeln und auswerten, z.B. für datengetriebene Geschäftsmodelle?

- Erst langfristig interessant
- Nachfrage und Akzeptanz unsicher
- Wettbewerbs- und datenschutzrechtliche Aspekte müssen genau geprüft werden

## 2. Aufbau des Belegs

### 2.1. PDF/A

Der PDF/A-ISO-Standard ist eine erweiterte Version des PDF-Dateiformats. Er wurde für die Langzeitarchivierung wichtiger Dokumente geschaffen.

Die Hauptunterschiede zu normalen PDFs liegen darin, dass keine Referenzen auf externe Ressourcen erlaubt sind (alle Schriftarten und Bilder müssen im Dokument eingebettet sein), keine Scriptsprachen wie Javascript enthalten sein dürfen, digitale Signaturen möglich sind und Anhänge direkt im PDF möglich und zugelassen sind.

PDF/A hat sich durch diese Eigenschaften zum Ablageformat der meisten DMS-Systeme entwickelt, da es vor allem durch die Unabhängigkeit von externen Ressourcen sicherstellen kann, dass die Dateien auch nach mehrjähriger Ablagedauer noch geöffnet werden können.

Die strukturierten, automatisch auswertbaren Daten des elektronischen Kassenbelegs werden als Anhang im JSON-Format in die PDF-Datei eingebettet.

Momentan werden keine Vorgaben zu Inhalten und Aufbau der im Klartext lesbaren Rechnung in der PDF-Datei gemacht. Die steuerrechtlichen Anforderungen an die Inhalte von Rechnungen sind zu beachten.

Um dem Klartext der Rechnung direkt ansehen zu können, dass strukturierte Daten vorhanden sind, wird empfohlen, den Text „EKaBS“ aufzunehmen, bevorzugt im Zusammenhang mit den TSE-spezifischen Daten.

### 2.2. JSON-Anhang

Der Anhang im PDF mit den Daten des elektronischen Beleges im JSON-Format erfolgt mit dem Dateinamen „ekabs.json“.

#### 2.2.1. Grundsätzliches

**JSON** (JavaScript Object Notation) ist ein schlankes Datenaustauschformat, das für Menschen einfach zu lesen und zu schreiben und für Maschinen einfach zu parsen (Analysieren von Datenstrukturen) und zu generieren ist. Es basiert auf einer Untermenge von JavaScript.

JSON ist ein Textformat, das komplett unabhängig von Programmiersprachen ist, aber vielen Konventionen folgt, die Programmieren aus der Familie der C-basierten Sprachen (inklusive C, C++, C#, Java, JavaScript, Perl, Python und vielen anderen) bekannt sind. Diese Eigenschaften machen JSON zum idealen Format für Datenaustausch.

Ein **JSON Schema** ist eine standardisierte Beschreibung von JSON-Daten, die insbesondere die Validierung, also die Überprüfung, ob Inhalt und Form bestimmter Daten einer vorher festgelegten Konvention entsprechen. Das JSON-Schema wird selbst ebenfalls im JSON-Format erstellt.

Die JSON-Struktur für elektronische Belege enthält folgende Daten:

- **Beleg:** Bildet die Klammer um den gesamten Beleg

- **Kassenobjekt:** Beschreibt die Kasse, die den Beleg erzeugt hat (z.Zt. nur Seriennummer)
- **Kopfdaten:** Steuerrechtlich erforderliche allgemeine Daten, die es genau einmal pro Beleg gibt, wie Daten zum Belegaussteller, ggf. Belegempfänger, Datum und Uhrzeiten usw.
- **Belegdaten:**
  - **Summendaten**, getrennt nach Steuersätzen und Zahlarten
  - **Belegpositionen**, also die einzelnen Waren bzw. Dienstleistungen sowie alle anderen Informationen, die in die Summenbildung einfließen, wie z.B. Rabatte
- **Sicherungsdaten:** Erlauben die (kryptografische) Überprüfung, dass die Daten des Belegs mittels einer TSE signiert und damit abgesichert wurden
- **Zusatzdaten:** Zusätzliche, nicht steuerrechtlich erforderliche Daten, wie z.B. Daten für eine bargeldlose Zahlung

Momentan werden zur maximalen Vereinfachung und zur Vermeidung von Problemen mit Hersteller-spezifischen Darstellungen alle Belegpositionen in gleicher Form abgebildet, also auch hierarchisch verknüpfte Positionen, Rabatte, Trinkgelder, durchlaufende Posten, Umsätze auf Rechnung Dritter usw. Erweiterungen sollen nach Bedarf erfolgen, siehe Abschnitt 4.

## 2.2.2. Versionierung

Jede JSON-Datei enthält eine Version, die angibt, welcher Version der Spezifikation die Daten entsprechen. Die Versionsnummer unterteilt sich in drei Teile:

### **MAJOR.MINOR.PATCH**

z. B. „2.1.3“.

Hierfür gelten folgende Festlegungen:

**PATCH:** Der PATCH-Teil der Versionsnummer wird erhöht, wenn ausschließlich abwärtskompatible Änderungen vorgenommen wurden, welche nicht die Struktur des JSON-Schemas betreffen. Das sind z. B. Änderungen der Beschreibungen, Erweiterungen von Datenlängen oder die Änderung von Pflichtfeldern auf optionale Felder. Daten, die innerhalb derselben MAJOR.MINOR-Version mit einer älteren PATCH- Version des JSON-Schemas positiv validiert wurden, müssen zwingend auch mit der neuen PATCH-Version des Schemas positiv validiert werden können.

**MINOR:** Die MINOR-Number muss sich erhöhen, wenn Daten, die innerhalb derselben MAJOR-Version mit einer älteren MINOR-Version des Schemas validiert wurden, unter Umständen mit der neuen Version des Schemas nicht mehr positiv validiert werden können. Das betrifft z. B. die Einschränkung von Feldinhalten und strukturelle Änderungen wie das Umbenennen und Löschen von Feldern oder die Einführung neuer Felder um bestehende Sachverhalte besser abbilden zu können.

**MAJOR:** Die MAJOR-Number muss sich erhöhen, wenn komplett neue Sachverhalte im JSON-Schema abgebildet werden, oder sich bestimmte Sachverhalte in der neuen Version nicht mehr abbilden lassen. Auch eine große Menge an strukturellen Änderungen, die jede für sich betrachtet nur eine Erhöhung der MINOR-Number rechtfertigen würde, sollte zu einer Erhöhung der MAJOR-Number führen.

Eine vollständige Spezifikation der JSON-Struktur findet sich im folgenden Abschnitt.

### 2.2.3. Darstellung der Brutto-, Netto- und Steuerbeträge

Bei der Darstellung der Steuerbeträge für den ganzen Beleg in **\$.data.vat\_amounts[\*]** sind neben dem Steuersatz immer alle drei Werte, als **incl\_vat** (Brutto), **excl\_vat** (Netto), und **vat** (Steuerbetrag), auszuweisen.

In den einzelnen Belegzeilen in **\$.data.lines[\*].vat\_amounts[\*]** wird unterschieden zwischen "Bruttomethode" und "Nettomethode".

Die Bruttomethode stellt ausschließlich den Bruttobetrag inklusive des umsatzsteuerlichen Anteils dar. Es wird nur **incl\_vat** angegeben, aber nicht **excl\_vat** und **vat**.

Die Nettomethode stellt den Nettobetrag sowie die zugehörige Umsatzsteuer dar. Es wird **excl\_vat** und **vat** angegeben, aber nicht **incl\_vat**.



## 3. Spezifikation JSON-Struktur

### 3.1. Beleg (root)

```
{
  "version": " 1 . 0 . 0 " ,
  "type": "INVOICE" ,
  "cash_register": {
    ...
  },
  "head": {
    ...
  },
  "data": {
    ...
  },
  "security": {
    ...
  },
  "misc": {
    ...
  }
}
```

<b>\$.version</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	^[0-9]+\.[0-9]+\.[0-9]+\$
<b>Beschreibung</b>	Versionsnummer des Standards für elektronische Belege

<b>\$.type</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	INVOICE
<b>Beschreibung</b>	Belegtyp, aktuell ist nur <b>INVOICE</b> zulässig

<b>\$.cash_register</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Information über die Kasse, welche den Beleg erstellt hat.

<b>\$.head</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Kopfdaten des Belegs

<b>\$.data</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Belegdaten

<b>\$.security</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Daten zur Absicherung des Beleges per TSE

<b>\$.misc</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Optionale Zusatzinformationen

### 3.2. Kassenobjekt (cash\_register)

```
{
  "serial_number": "Q345/98Z"
}
```

<b>\$.cash_register.serial_number</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Seriennummer der Kasse

### 3.3. Kopfdaten (head)

```
{
  "id": "45130a36-29ed-4e96-817b-e01d02087d14",
  "number": "2020/1001",
  "date": "2020-08-10T10:51:33.7892765+00:00",
  "delivery_period_start": "2020-08-07",
  "delivery_period_end": "2020-08-08",
  "seller": {
    ...
  },
  "buyer_text": "Erika Mustermann, Heidestraße 17, 51147 Köln",
  "buyer": {
    ...
  }
}
```

\$head.id	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Optionale ID zur Kennzeichnung des Belegs, welche zur Referenzierung innerhalb des ausstellenden Kassensystems genutzt werden kann

\$head.number	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	(Fortlaufende) Rechnungsnummer
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

<b>\$.head.date</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	date-time (RFC 3339, section 5.6.)
<b>Beschreibung</b>	Rechnungsdatum
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

<b>\$.data.head.delivery_period_start</b>	
<b>Typ</b>	string
<b>Pflichtfeld</b>	Nur wenn \$.data.head.delivery_period_end angegeben wurde
<b>Wertebereich</b>	ISO8601 / RFC 3339, section 5.6 Datum
<b>Beschreibung</b>	Start des Leistungszeitraums, falls abweichend vom Rechnungsdatum

<b>\$.data.head.delivery_period_end</b>	
<b>Typ</b>	string
<b>Pflichtfeld</b>	Nein
<b>Wertebereich</b>	ISO8601 / RFC 3339, section 5.6 Datum
<b>Beschreibung</b>	Ende des Leistungszeitraums, falls abweichend vom Rechnungsdatum

<b>\$.head.seller</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Informationen zum Belegaussteller / leistenden Unternehmen
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

<b>\$.head.buyer_text</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Name und Adresse des Belegempfängers in einfacher Textform. Sollte nur genutzt werden, wenn die Daten nicht in strukturierter Form vorliegen (siehe \$.head.buyer)
<b>Rechtsgrundlage</b>	UStDV §31 Abs.2, UStG §14 Abs. 4, UStAE Abschn. 14.5

<b>\$.head.buyer</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Name und Adresse des Belegempfängers in strukturierter Form. Liegen die Daten nicht in strukturierter Form vor, sollte \$.head.buyer_text genutzt werden.
<b>Rechtsgrundlage</b>	UStDV §31 Abs.2, UStG §14 Abs. 4, UStAE Abschn. 14.5

### 3.3.1. Belegaussteller

```
{
  "name": "Muster GmbH",
  "tax_number": "123/456/789",
  "tax_exemption": true,
  "tax_exemption_note": "Umsatzsteuerfreie Leistungen gemäß §19 UStG",
  "address": {
    ...
  }
}
```

\$ .head.seller.name	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Name des leistenden Unternehmens, korrekte Firmierung
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

\$ .head.seller.tax_number	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Steuernummer oder Umsatzsteueridentifikationsnummer des leistenden Unternehmens
<b>Rechtsgrundlage</b>	§14 Abs. 4 Nr. 2 UStG

\$ .head.seller.tax_exemption	
<b>Typ</b>	Boolean

<b>Pflichtfeld</b>	Ja, wenn Steuerbefreiung vorliegt
<b>Default</b>	False
<b>Beschreibung</b>	Wenn eine Steuerbefreiung vorliegt, wird dieses Feld auf true gesetzt. In diesem Fall muss head.seller.tax_exemption_note gesetzt sein.
<b>Rechtsgrundlage</b>	§ 19 UStG

<b>\$.head.seller.tax_exemption_note</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja, wenn head.seller.tax_exemption true ist
<b>Beschreibung</b>	Grund der Steuerbefreiung, wenn head.seller.tax_exemption true ist.
<b>Rechtsgrundlage</b>	§ 19 UStG

<b>\$.head.seller.address</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Adresse des leistenden Unternehmens in strukturierter Form
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG



```
{
  "street": "Heidestraße 17",
  "postal_code": "51147",
  "city": "Köln",
  "country_code": "DEU"
}
```

<b>\$.head.seller.address.street</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	Länge 0 - 60
<b>Beschreibung</b>	Straßenname und Hausnummer des leistenden Unternehmens
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

<b>\$.head.seller.address.postal_code</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	Länge 0 - 10
<b>Beschreibung</b>	Postleitzahl des leistenden Unternehmens
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

<b>\$.head.seller.address.city</b>	
<b>Typ</b>	String

<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	Länge 0 - 62
<b>Beschreibung</b>	Stadt/Ort des leistenden Unternehmens
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

<b>\$.head.seller.address.country_code</b>	
<b>Typ</b>	None
<b>Pflichtfeld</b>	Ja
<b>Default</b>	DEU
<b>Wertebereich</b>	ALA, AFG, ALB, DZA, ASM, AND, AGO, AIA, ATA, ATG, ...
<b>Beschreibung</b>	Länderkennung des leistenden Unternehmens gemäß ISO 3166 alpha-3 country code
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

### 3.3.2. Belegempfänger

```
{
  "customer_number": "K1298",
  "name": "Erika Mustermann",
  "tax_number": "DE987654321",
  "address": {
    ...
  }
}
```

<b>\$.head.buyer.customer_number</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Kundennummer des Leistungsempfängers

<b>\$.head.buyer.name</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Name/Firmenname des Leistungsempfängers

<b>\$.head.buyer.tax_number</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Steuernummer oder Umsatzsteueridentifikationsnummer des Leistungsempfängers

<b>\$.head.buyer.address</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Adresse des Leistungsempfängers in strukturierter Form

```

{
  "street": "Heidestraße 17",
  "postal_code": "51147",
  "city": "Köln",
  "country_code": "DEU"
}

```

<b>\$.head.buyer.address.street</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Wertebereich</b>	Länge 0 - 60
<b>Beschreibung</b>	Straße und Hausnummer des Leistungsempfängers

<b>\$.head.buyer.address.postal_code</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Wertebereich</b>	Länge 0 - 10
<b>Beschreibung</b>	Postleitzahl des Leistungsempfängers

<b>\$.head.buyer.address.city</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Wertebereich</b>	Länge 0 - 62

<b>Beschreibung</b>	Stadt/Ort des Leistungsempfängers
---------------------	-----------------------------------

<b>\$.head.buyer.address.country_code</b>	
<b>Typ</b>	None
<b>Pflichtfeld</b>	Nein
<b>Wertebereich</b>	ALA, AFG, ALB, DZA, ASM, AND, AGO, AIA, ATA, ATG, ...
<b>Beschreibung</b>	Ländercode des Leistungsempfängers gemäß ISO 3166 alpha-3 country code

### 3.4. Belegdaten (data)

```
{
  "currency": "EUR",
  "full_amount_incl_vat": 10.49,
  "payment_types": [
    {
      ...
    }
  ],
  "vat_amounts": [
    {
      ...
    }
  ],
  "lines": [
    {
      ...
    }
  ]
}
```

<b>\$.data.currency</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Default</b>	EUR
<b>Wertebereich</b>	ISO 4217 Währungscode (alphabetischer Code)
<b>Beschreibung</b>	Dieses Feld legt die für alle Belegdaten anzuwendende Währung fest, falls keine andere Währung explizit angegeben wird.

<b>\$.data.full_amount_incl_vat</b>	
<b>Typ</b>	Numerisch
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	Max. 2 Nachkommastellen
<b>Beschreibung</b>	Der Brutto-Gesamtbetrag des Belegs
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

<b>\$.data.payment_types</b>	
<b>Typ</b>	Array
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Liste aller Zahlungswege, über die dieser Beleg ausgeglichen wurde.

<b>\$.data.vat_amounts</b>	
<b>Typ</b>	Array
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Gesamtbeträge pro Steuersatz – dies entspricht den Beträgen, so wie sie auch auf dem gedruckten Beleg ausgewiesen werden.
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

<b>\$.data.lines</b>	
<b>Typ</b>	Array
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Einzelpostenauflistung der erbrachten Leistungen
<b>Rechtsgrundlage</b>	§ 14 Abs. 4 i.V.m. § 14a Abs. 5 UStG

```
{
  "name": "BAR",
  "amount": 10.49,
  "foreign_amount": 12.33
  "foreign_currency": "USD",
}
```

<b>\$.data.payment_types[*].name</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Bezeichnung der Zahlungsart, muss mindestens zwischen bar und unbar unterscheiden.

<b>\$.data.payment_types[*].amount</b>	
<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Zahlungsbetrag in der lokalen Wahrung (\$.data.currency)



**\$.data.payment\_types[\*].foreign\_amount**

<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja, falls foreign_currency verwendet wird
<b>Beschreibung</b>	Bei Zahlung mit Fremdwahrung der Betrag in Fremdwahrung. In diesem Fall muss die Fremdwahrung in \$.data.payment_types[*].foreign_currency gesetzt werden.

**\$.data.payment\_types[\*].foreign\_currency**

<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja, falls foreign_amount angegeben ist
<b>Wertebereich</b>	ISO 4217 Wahrungscod (alphabetischer Code)
<b>Beschreibung</b>	Bei Zahlung mit Fremdwahrung der Fremdwahrungscod

```
{  
  "percentage": 19,  
  "incl_vat": 10.49,  
  "excl_vat": 8.82,  
  "vat": 1.67  
}
```

**\$.data.vat\_amounts[\*].percentage**

<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja

<b>Wertebereich</b>	Min 0 / Max 100 / 2 Nachkommastellen
<b>Beschreibung</b>	Umsatzsteuersatz * 100, 19% werden als 19.00 dargestellt

<b>\$.data.vat_amounts[*].incl_vat</b>	
<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	2 Nachkommastellen
<b>Beschreibung</b>	Bruttobetrag für diesen Steuersatz

<b>\$.data.vat_amounts[*].excl_vat</b>	
<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	2 Nachkommastellen
<b>Beschreibung</b>	Nettobetrag für diesen Steuersatz

<b>\$.data.vat_amounts[*].vat</b>	
<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	2 Nachkommastellen
<b>Beschreibung</b>	Absoluter Steuerbetrag, $incl\_val - excl\_vat = vat$

```

{
  "text": "Regenjacke",
  "additional_text": "B-Ware",
  "vat_amounts": [],
  "item": {
    ...
  },
  "delivery_period_start": "2020-08-07",
  "delivery_period_end": "2020-08-08"
}

```

### **\$.data.lines[\*].text**

<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Beschreibt die Art des gelieferten Gegenstandes / der erbrachten Leistung oder anderer Geschäftsvorfälle. Dies ist i.d.R. die hinreichend spezifische Bezeichnung des verkauften Artikels oder der erbrachten Leistung, kann aber auch die Bezeichnung eines gegebenen Rabattes o.ä. sein.
<b>Rechtsgrundlage</b>	§6, Abs. 5 KassenSichV, AEAO zu § 146a Abschn. 5.4 Nr. 3

### **\$.data.lines[\*].additional\_text**

<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Zusatzinformation zur Rechnungsposition, welche über die gesetzliche Anforderung an den Beleg hinausgehen, für den Belegempfänger aber von Interesse sein können. z.B. die Seriennummer eines verkauften Artikels, der Prozentwert eines Rabattes oder ähnliches.

<b>\$.data.lines[*].vat_amounts</b>	
<b>Typ</b>	Array
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Rechnungsbeträge der einzelnen Rechnungsposition aufgeteilt nach Steuersätzen. Eine Rechnungsposition kann auf mehrere Steuersätze aufgeteilt sein, wenn z.B. in der Gastronomie eine Zusammenstellung von Produkten (Menü) aus Speisen und Getränken besteht und „außer Haus“ verkauft wird.

<b>\$.data.lines[*].item</b>	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Detaillierte Beschreibung des Gegenstandes einer Rechnungsposition. Belegpositionen können in "item" mit Mengenangaben, Artikelnummer u.s.w. näher beschrieben werden. Der "Artikelname" wird als Text-Feld auf line-Ebene angegeben (\$.data.lines[*].text) und daher im item-Objekt nicht wiederholt.

<b>\$.data.lines[*].delivery_period_start</b>	
<b>Typ</b>	string
<b>Pflichtfeld</b>	Nur wenn \$.data.lines[*].delivery_period_end angegeben wurde
<b>Wertebereich</b>	ISO8601 / RFC 3339, section 5.6 Datum
<b>Beschreibung</b>	Start des Leistungszeitraums falls abweichend vom Rechnungsdatum

<b>\$.data.lines[*].delivery_period_end</b>	
<b>Typ</b>	string
<b>Pflichtfeld</b>	Nein
<b>Wertebereich</b>	ISO8601 / RFC 3339, section 5.6 Datum
<b>Beschreibung</b>	Ende des Leistungszeitraums falls abweichend vom Rechnungsdatum

```
{
  "number": "0815",
  "gtin": "4013595313208",
  "quantity": 1,
  "quantity_measure": "Stück",
  "price_per_unit": 10.49
}
```

<b>\$.data.lines[*].item.number</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Artikelnummer des verkauften Artikels/Produktes oder der erbrachten Dienstleistung

<b>\$.data.lines[*].item.gtin</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Wertebereich</b>	Mind. 1 max. 50 Zeichen

<b>Beschreibung</b>	GTIN /EAN des Artikels
---------------------	------------------------

#### **\$.data.lines[\*].item.quantity**

<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	3 Nachkommastellen
<b>Beschreibung</b>	Menge des verkauften Artikels/Produktes oder der erbrachten Dienstleistung

#### **\$.data.lines[\*].item.quantity\_measure**

<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Mengeneinheit der unter quantity angegebenen Menge wie z.B. „Stück“, „kg“, „Liter“, „Bund“, ...

#### **\$.data.lines[\*].item.price\_per\_unit**

<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	5 Nachkommastellen
<b>Beschreibung</b>	Bruttopreis des Artikels/Produkts oder der verkauften Dienstleistung pro Einheit. $price\_per\_unit * quantity = SUM(vat\_amounts[*].incl\_vat)$

Brutto:

```
{
  "percentage": 19.0,
  "incl_vat": 10.49
}
```

Netto:

```
{
  "percentage": 19.0,
  "excl_vat": 8.82,
  "vat": 1.67
}
```

#### **\$.data.lines[\*].vat\_amounts[\*].percentage**

<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	Min 0 / Max 100 / 2 Nachkommastellen
<b>Beschreibung</b>	Steuersatz

#### **\$.data.lines[\*].vat\_amounts[\*].incl\_vat**

<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja, wenn excl_vat und vat nicht gesetzt sind
<b>Wertebereich</b>	5 Nachkommastellen
<b>Beschreibung</b>	Bruttobetrag - entfällt, wenn excl_vat und vat stattdessen verwendet werden

<b>\$.data.lines[*].vat_amounts[*].excl_vat</b>	
<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja, wenn incl_vat nicht gesetzt ist
<b>Wertebereich</b>	5 Nachkommastellen
<b>Beschreibung</b>	Nettobetrag – entfällt, wenn incl_vat verwendet wird

<b>\$.data.lines[*].vat_amounts[*].vat</b>	
<b>Typ</b>	Number
<b>Pflichtfeld</b>	Ja, wenn incl_vat nicht gesetzt ist
<b>Wertebereich</b>	5 Nachkommastellen
<b>Beschreibung</b>	Steuerbetrag – entfällt, wenn vat verwendet wird



### 3.5. Sicherungsdaten (security)

```
{
  "tse": {
    ...
  }
}
```

\$security.tse	
<b>Typ</b>	Object
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Sicherheitsmerkmale, die für die Verifikation des Beleges nach KassenSichV relevant sind.
<b>Rechtsgrundlage</b>	KassenSichV; 146a AO

```
{
  "serial_number": "623323B6C170DF2200...8F3A78E5BA7C4BA60B",
  "signature_algorithm": "ecdsa-plain-SHA384",
  "log_time_format": "unixTime",
  "certificate": "LS0tLS...S0tCg==",
  "timestamp_start": "2020-08-10T10:51:33.8049447+00:00",
  "timestamp_end": "2020-08-10T10:51:33.8049447+00:00",
  "first_order": "2020-08-10T10:51:33.8049447+00:00",
  "transaction_number": 50,
  "signature_number": 121,
  "process_type": "Kassenbeleg-V1",
  "process_data": "Beleg^42.31_16.26_0.00_0.00_0.00^58.57:Bar",
  "signature": "AjEGS...dhRitb"
}
```

<b>\$.security.tse.serial_number</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	Länge 64 - / ^{([0-9a-fA-F]{2})}+\$
<b>Beschreibung</b>	Seriennummer der TSE in hexadezimaler Darstellung
<b>Rechtsgrundlage</b>	KassenSichV; 146a AO

<b>\$.security.tse.signature_algorithm</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Default</b>	ecdsa-plain-SHA256
<b>Wertebereich</b>	ecdsa-plain-SHA224, ecdsa-plain-SHA256, ecdsa-plain-SHA384, ecdsa-plain-SHA512, ecdsa-plain-SHA3-224, ecdsa-plain-SHA3-256, ecdsa-plain-SHA3-384, ecdsa-plain-SHA3-512, ecsdsa-plain-SHA224, ecsdsa-plain-SHA256, ...
<b>Beschreibung</b>	Von der TSE genutzter Signaturalgorithmus
<b>Rechtsgrundlage</b>	KassenSichV; 146a AO

<b>\$.security.tse.log_time_format</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja

<b>Default</b>	unixTime
<b>Wertebereich</b>	unixTime, utcTime, utcTimeWithSeconds, generalizedTime, generalizedTimeWithMilliseconds
<b>Beschreibung</b>	Von der TSE genutztes Datumsformat
<b>Rechtsgrundlage</b>	KassenSichV; 146a AO

<b>\$.security.tse.certificate</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Gesamte Zertifikatskette der genutzten TSE im PEM-Format
<b>Rechtsgrundlage</b>	KassenSichV; 146a AO

<b>\$.security.tse.timestamp_start</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Zeitstempel des Starts der Transaktion. Der Zeitstempel entspricht dem von der TSE zurückgegebenen Zeitstempel beim Ausführen der start()-Funktion.

<b>\$.security.tse.timestamp_end</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja

<b>Beschreibung</b>	Zeitstempel des Endes der Transaktion. Der Zeitstempel entspricht dem von der TSE zurückgegebenen Zeitstempel beim Ausführen der finish()-Funktion.
---------------------	---

#### **\$.security.tse.first\_order**

<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja, wenn Bestellungen bei dieser Rechnung abgesichert wurden.
<b>Beschreibung</b>	Zeitstempel des Starts der ersten Bestell-Transaktion. Der Zeitstempel entspricht dem von der TSE zurückgegebenen Zeitstempel beim ausführen der start()-Funktion der ersten Bestellung.

#### **\$.security.tse.transaction\_number**

<b>Typ</b>	Integer
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	Min 0
<b>Beschreibung</b>	Die von der TSE vergebene Transaktionsnummer

#### **\$.security.tse.signature\_number**

<b>Typ</b>	Integer
<b>Pflichtfeld</b>	Ja
<b>Wertebereich</b>	Min 0
<b>Beschreibung</b>	Von der TSE vergebener Signaturzähler. Der Signaturzähler wird der Antwort beim Aufruf der finish()-Funktion der Beleg-Transaktion entnommen.

<b>\$.security.tse.process_data</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Inhalt der processData beim Aufruf der finish()-Funktion der Beleg-Transaktion

<b>\$.security.tse.process_type</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Default</b>	Kassenbeleg-V1
<b>Wertebereich</b>	Länge - 30
<b>Beschreibung</b>	Inhalt des processTypes beim Aufruf der finish()-Funktion der Beleg-Transaktion
<b>Rechtsgrundlage</b>	DSFinV-K

<b>\$.security.tse.signature</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Prüfwert/Signatur, welche(n) die TSE beim Aufruf der finish()-Funktion der Belegtransaktion zurückgegeben hat

### 3.6. Zusatzdaten (misc)

```
{
  "logo": {
    ...
  },
  "footer_text": "Bitte beachten Sie auch unsere Sommer-Sonderangebote!",
  "additional_receipts": [
    {
      ...
    }
  ]
}
```

#### \$.misc.logo

<b>Typ</b>	Object
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Logo, welches auf der gedruckten / angezeigten Rechnung dargestellt werden soll

#### \$.misc.footer\_text

<b>Typ</b>	String
<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Optionaler Fußtext, der am Ende der Rechnung erscheinen soll

#### \$.misc.additional\_receipts

<b>Typ</b>	Array
------------	-------

<b>Pflichtfeld</b>	Nein
<b>Beschreibung</b>	Zusätzliche Belege in Text-Form, als Bild oder PDF (z.B. Kartenterminal-Belege)

```
{
  "content_type": "image/png",
  "content": "..."
}
```

<b>\$.misc.logo.content_type</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	MIME-Typ des Logos, z.B. image/png oder text/x-uri

<b>\$.misc.logo.content</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Base64-codierter Inhalt des Logos. Ist das Logo eine URL (text/x-uri) kann die base64-Kodierung entfallen.

```
{
  "content_type": "text/plain",
  "content": "* * EC-Kundenbeleg * *\nMuster GmbH\n..."
}
```

<b>\$.misc.additional_receipts[*].content_type</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	MIME-Typ des zusätzlichen Belegs, z.B. image/png, application/pdf oder text/plain

<b>\$.misc.additional_receipts[*].content</b>	
<b>Typ</b>	String
<b>Pflichtfeld</b>	Ja
<b>Beschreibung</b>	Base64-codierter Inhalt des zusätzlichen Beleges



## 4. Zukünftige Erweiterungen

In diesem Abschnitt sind die momentan bekannten Erweiterungsvorschläge aufgelistet. Über eine Implementierung bzw. eine Reihenfolge gibt es noch keine Entscheidungen. Diese werden auf der Basis von Praxiserfahrungen und Rückmeldungen aller Stakeholder getroffen.

Folgende Vorschläge liegen momentan vor:

- Standardisierung der im Abschnitt 1.4 genannten Ebenen 4 bis 6
- Zu älteren Versionen des Standards kompatibel, detaillierte Darstellungsmöglichkeiten in den Belegpositionen (`$.data.lines`) für:
  - Rabatte
  - Coupons
  - Pfand
  - Unterscheidungen der Kaufpositionen (Artikel, Services, Giftcards/Gutscheine)
  - Trinkgelder (Arbeitnehmer und Arbeitgeber)
  - Agenturgeschäfte, durchlaufende Posten
- Hierarchische Verknüpfung von Belegpositionen – typischer Anwendungsfall ist die Darstellung wählbarer Beilagen in der Gastronomie
- Weitere Felder für die Stammdaten des Belegausstellers (unter `$.head.seller`) wie Telefon, E-Mail, Website
- Möglichkeit, den Belegempfänger pseudonymisiert, also nur per Kundennummer ohne Name und Adressdaten, zu erfassen
- Klare Regeln für Hersteller-spezifische Erweiterungen um sicherzustellen, dass Schlüssel immer eindeutig sind
- Erweiterungen für die Übertragung von Belegdaten per JSON-Struktur an ein System, das daraus dann den kompletten Beleg erstellt inkl. Einbindung Grafiken, Textformatierung etc. (auch durch Hersteller-spezifische Erweiterungen) – typischer Anwendungsfall ist der Versand der Belegdaten vom Kassensystem an einen Drucker, der wahlweise einen Papierbeleg druckt oder über eine Cloud-Lösung einen elektronischen Beleg bereitstellt