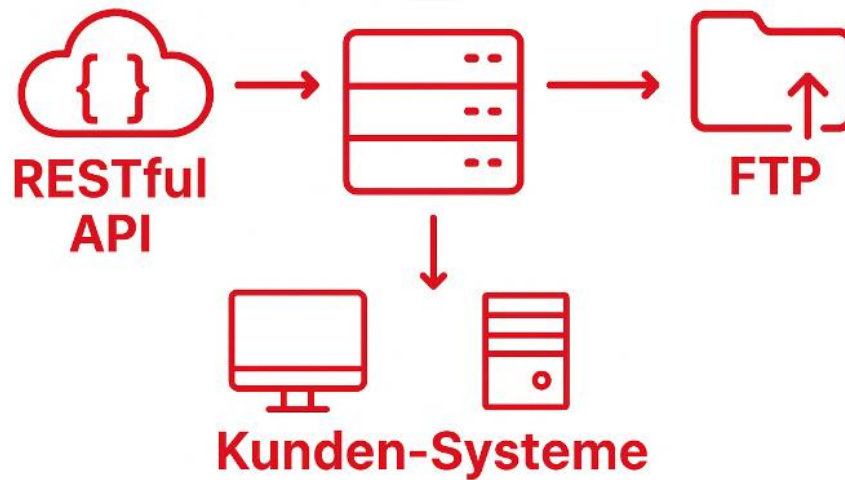


HEGA



Anbindung HEGA GmbH

Schnittstellen-Handbuch

— FTP- und API-Integration fuer Dropshipping-Partner —

Version 3.0 · Stand: August 2026

Inhaltsverzeichnis

Anbindung HEGA GmbH — Schnittstellen-Handbuch	3
1. Wegweiser — welches Kapitel für wen?	5
2. Die Anbindung im Überblick (<i>für den Fachbereich</i>)	6
2.1 Worum geht es?	6
2.2 Was passiert, wenn eine Bestellung an HEGA geht?	6
2.3 Was bekommen Sie zurück?	7
2.4 Womit können Sie wann rechnen?	7
2.5 Was Sie entscheiden sollten	8
3. Überblick & Adressen (<i>für die IT</i>)	8
4. Die FTP-Integration	9
4.1 Aufbau der Bestelldatei	10
4.2 Adressfelder Endkunde (Semikolon-getrennt)	11
4.3 Artikelzeilen	11
4.4 Abruf & Rückmeldungen	12
4.5 Tägliche Stammdaten per FTP	14
5. Die API-Integration	14
5.1 Zugang & Authentifizierung	15
5.2 Bestellung anlegen	15
5.3 Endpunkt-Referenz	23
5.4 Technische Hinweise für die Anbindung	25

5.5 Test & Abnahme	28
6. Nach der Bestellung — Status & Belege abrufen	29
6.1 Status & Felder.....	29
6.2 Empfohlene Abruf-Strategie (Polling)	30
7. FTP oder API — was wann?.....	30
8. Rückstandsfunktion (optional)	31
9. Frachtschein — eigenes Logo & Fußzeile	32
10. Stornierung & Support	34
11. Glossar.....	34
12. Anhang — Für SAP-Anwender	35
12.1 Begriffe HEGA ↔ SAP	35
12.2 Integration: REST/FTP statt IDoc	36
12.3 Typischer Ablauf aus SAP-Sicht.....	37
12.4 Partnerrollen & Adressen	38
12.5 Mengeneinheit / VE	38
12.6 SAP-ORDERS05 → HEGA-JSON: Feld-Mapping	40
13. Anhang — Gestaltungsmöglichkeiten für Ihren Prozess	40
13.1 In Stufen einsteigen — Sie bestimmen das Tempo.....	41
13.2 Wo Sie Spielraum haben	41
13.3 Rahmen, den Sie einplanen sollten	42

Anbindung HEGA GmbH — Schnittstellen-Handbuch

FTP- und API-Integration für Dropshipping-Partner. Version 3.0 · Stand: August 2026

Neu in Version 3.0: Dritter Referenzclient zum Download — **Node.js/TypeScript** (/hega-api-client.cjs, Node.js 18+, ohne npm install; die TypeScript-Typen liegen separat als /hega-api-client.d.cts daneben). Dazu ein **JavaScript-Code-Beispiel** in Kapitel 5.2. Alle drei Clients — PHP, Python und Node.js — sind inhaltlich identisch aufgebaut, Sie können also frei wählen. Keine Änderung an Endpunkten, Feldern oder Statuscodes.

Neu in Version 2.9: Klarstellung in Kapitel 5.4: Fehlertexte der Beleg- und Bestands-Endpunkte sind **bewusst allgemein gehalten** und enthalten keine technischen Details. Werten Sie den **HTTP-Statuscode** aus; für die Klärung eines konkreten Fehlers genügt uns Zeitpunkt, Kundennummer und Endpunkt. Keine Änderung an Endpunkten, Feldern oder Statuscodes.

Neu in Version 2.8: Zusätzlich zum PHP-Client gibt es jetzt auch einen **Python-Referenzclient zum Download** (/hega_api_client.py, Python 3.8+, nur Standardbibliothek — kein pip install). Beide Clients sind inhaltlich identisch aufgebaut; die Übersicht steht in Kapitel 5.2.

Neu in Version 2.7: **Code-Beispiel für PHP** in Kapitel 5.2 sowie ein vollständiger, lauffähiger **PHP-Referenzclient zum Download** (/HEGA-PHP-Client.php, PHP 7.4+, ohne Composer). Ergänzt in Kapitel 5.4: Der Wert expiresAt der Login-Antwort gehört zum Refresh-Token und darf **nicht** als Ablauf des Access-Tokens verwendet werden; außerdem sind jetzt **beide** Fehler-Antwortformate beschrieben (strukturierter errorCode vs. formale Feldprüfung ohne errorCode) — bisher war nur das erste dokumentiert. Klarstellung in 5.1 zur Token-Erneuerung. Keine Änderung an der Schnittstelle selbst.

Neu in Version 2.6: **Kontaktdaten des Empfängers bei versender: "DPD"**. DPD benachrichtigt den Empfänger selbst über Zustelltag, Zeitfenster, Umleitung und Ablageort und verlangt dafür eine Kontaktinfo. Bei DPD ist deshalb **eine gültige emailadresse oder eine mobilnummer Pflicht** — fehlt beides, weisen wir die Bestellung sofort mit HTTP 400 (MISSING_CONTACT_DATA) zurück. Neues **optionales Feld mobilnummer** (max. 25

Zeichen) als Alternative zur E-Mail. Bei versender: "DHL" ändert sich nichts. Details in Kapitel 5.2 und 5.4.

Neu in Version 2.5: Optionale **serverseitige Paginierung** der Artikeldaten-Endpunkte (GET /api/DSArtikelDaten, .../light, .../lastchange/{date}) über **page** (ab 1) und **pageSize** (Default 500, max 5000) — **abwärtskompatibel**: ohne diese Parameter wird wie bisher das komplette Sortiment in einer Antwort geliefert. Empfohlen für sehr große Sortimente gegen Timeouts; Gesamtzahl im Header **X-Total-Count**. Details in Kapitel 5.4.

Neu in Version 2.4: Klarstellung zur **Verpackungseinheit (VE)**: Eine VE ist eine **physische Verpackungseinheit, die nicht aufgetrennt wird** — geliefert und berechnet werden immer ganze Gebinde. mengeneinheit: "Stueck" ändert nur die **Eingaberechnung**, nicht diese Tatsache; lose Einzelstücke (z. B. 1 von 6) sind nicht bestellbar. Ergänzt in Kapitel 4.3 & 12.5 (keine API-Änderung).

Neu in Version 2.3: **Dry-Run / Vorab-Prüfung** POST .../OrderValidate — eine Bestellung (Adresse, Artikel, Mengen/VE) prüfen, **ohne** sie anzulegen; die Antwort listet **alle** Probleme auf einmal (valid + errors). OrderAdd meldet ungültige Positionen jetzt ebenfalls **gesammelt** im Feld errors (abwärtskompatibel — oberste Ebene bleibt der erste Fehler). Beides für Dropshipping (/api/DSOrder) **und** B2B (/api/Order); siehe Kapitel 5.2/5.3.

Neu in Version 2.2: Neuer Anhang 12.6 mit dem **SAP-ORDERS05** → **HEGA-JSON-Feldmapping** sowie **Code-Beispiele** (cURL, C#, Java, Python) für Login + Bestellung in Kapitel 5.2.

Neu in Version 2.1: Kapitel 5.2 beschreibt jetzt den **Bestellablauf je Bestellung** mit zwei Varianten — mit oder **ohne eigene Stammdatenhaltung** (gezielter Artikelabruf vor dem Absenden). Neues optionales Positionsfeld **mengeneinheit** (VE/Stueck): Sie können die Menge in Stück übergeben, HEGA rechnet anhand von anzahlartve selbst auf VE um (Kapitel 5.2 & 12.5). Einzelartikel-Endpunkt in die Referenz aufgenommen.

Neu in Version 2.0: neue Abschnitte 5.4 „**Technische Hinweise für die Anbindung**“ (Datums-/Zeitformat, Paginierung, 12-Monats-Fenster, Token- & Fehlerverhalten, Encoding,

Freischaltung pro Konto) und **5.5 „Test & Abnahme“**; präzisierte **Länderliste** (per API bestellbar) und Login-Beispiel. Vorgängerfassung: 26.09.2025.

1. Wegweiser — welches Kapitel für wen?

Dieses Handbuch deckt unterschiedliche Rollen ab. Die folgende Übersicht hilft Ihnen, gezielt die für Sie relevanten Kapitel zu finden — Sie müssen nicht alles lesen:

Ihre Rolle	Empfohlene Kapitel
Fachbereich / kaufmännisch — Sie wollen verstehen, <i>was</i> die Anbindung leistet und <i>was Sie davon haben</i>	Kapitel 2 (Überblick), 8 (Rückstand), 10 (Storno & Support), 11 (Glossar)
IT / Entwicklung — Sie binden die Schnittstelle technisch an	Kapitel 4 (FTP), 5 (API, inkl. 5.4 Technische Hinweise & 5.5 Test & Abnahme), 6 (Status & Belege abrufen)
SAP-Anwender — Sie denken in PO / ASN / Rechnung	Anhang 12 (Begriffe & Abläufe HEGA ↔ SAP)
Prozessverantwortliche — Sie gestalten den Geschäftsprozess	Anhang 13 (Gestaltungsmöglichkeiten & Spielräume)

Kapitel 2 beschreibt die Anbindung **ohne technische Details** — als gemeinsame Grundlage für beide Seiten. Begriffe wie *Lieferschein*, *Tracking* oder *Endpunkt* sind im **Glossar** erklärt.

2. Die Anbindung im Überblick *(für den Fachbereich)*

2.1 Worum geht es?

Über die HEGA-Schnittstelle kann **Ihr System** (z. B. Ihr Online-Shop oder Warenwirtschaft) direkt mit HEGA „sprechen“: Bestellungen Ihrer Endkunden werden automatisch an HEGA übermittelt, HEGA liefert die Ware **direkt an Ihren Endkunden** (Dropshipping) und meldet Ihnen zurück, was mit der Bestellung passiert ist. Es muss niemand mehr Bestellungen per Hand abtippen.

Es gibt **zwei technische Wege** für diese Verbindung — Ihre IT wählt einen aus: - **FTP** — einfacher Dateiaustausch (wie ein automatisches Postfach für Dateien). - **API** — direkte, sofortige Verbindung von Programm zu Programm. Der modernere Weg, über den Sie auch alle Rückmeldungen (Status, Lieferschein, Sendungsverfolgung) bekommen.

2.2 Was passiert, wenn eine Bestellung an HEGA geht?

Jede Bestellung durchläuft bei HEGA **sechs Stationen**. Sie können jederzeit nachsehen, an welcher Station Ihre Bestellung gerade steht:

#	Station	Was das bedeutet
1	Eingegangen	Die Bestellung ist bei HEGA angekommen und geprüft.
2	Im Auftragspool	Sie wurde ins HEGA-System übernommen und wartet aufs Lager.
3	In Bearbeitung	Das Lager stellt die Ware zusammen (Kommissionierung).
4	Gepackt	Die Ware ist geprüft und verpackt, wartet auf den Paketdienst.

#	Station	Was das bedeutet
5	Versendet	Das Paket ist unterwegs — jetzt gibt es eine Sendungsnummer.
6	Berechnet	Die Rechnung wurde erstellt.

2.3 Was bekommen Sie zurück?

Sobald Ihre IT den Rückkanal nutzt, stehen Ihnen jederzeit zur Verfügung:

- der **Bestellstatus** (eine der sechs Stationen oben),
- der **Lieferschein** als PDF oder als Datenliste,
- die **Sendungsverfolgung** (Sendungsnummer + fertiger Link), sobald versendet wurde,
- die **Rechnung** als PDF oder Datenliste,
- offene **Rückstände** (was noch nachgeliefert wird).

Damit können Sie Ihre eigenen Endkunden informieren, ohne bei HEGA nachfragen zu müssen.

2.4 Womit können Sie wann rechnen?

Schritt	Üblicher Rahmen
Bestellung wird ins Lager übernommen	binnen Minuten (Abruf alle ~15 Min.)
Kommissionierung & Versand	i. d. R. selber oder nächster Werktag
Rechnung	werktags am Abend (Mo–Fr)

Schritt	Üblicher Rahmen
Zustellung beim Empfänger	1–2 Werktage nach Versand

2.5 Was Sie entscheiden sollten

1. **Welcher Weg — FTP oder API?** (Empfehlung: API, weil Sie darüber auch alle Rückmeldungen erhalten. Vergleich in Kapitel 7.)
2. **Wollen Sie den Rückkanal nutzen?** Also Status & Sendungsverfolgung automatisch abrufen — oder reicht Ihnen vorerst „Bestellung rein, Lieferschein per Mail“? Das bestimmt, wie viel Ihre IT anbinden muss.
3. **Rückstände erwünscht?** (Teil-Lieferung mit automatischer Nachlieferung — siehe Kapitel 6.)

3. Überblick & Adressen (für die IT)

Adressen:

System	Adresse
API	https://api.hega.net
Web-Portal (Reservierungen, Storno, Auswertungen)	https://portal.hega.net

Kontakt: FTP & Stornierungen support@ftphega.de · Frachtschein-Logo & API-Freischaltung EDV@hega.net

Unterstützte Versandländer — Versand via **DPD** (Standard) oder **DHL**, Ländercodes nach **ISO 3166-1 alpha-2**. Per API bestellbar sind aktuell diese 26 Länder:

DE Deutschland	BE Belgien	LU Luxemburg	NL Niederlande
AT Österreich	DK Dänemark	CZ Tschechien	IT Italien
FR Frankreich	MC Monaco	HU Ungarn	PL Polen
SI Slowenien	SK Slowakei	PT Portugal	SE Schweden
ES Spanien	FI Finnland	IE Irland	EE Estland
LT Litauen	LV Lettland	BG Bulgarien	HR Kroatien
RO Rumänien	GR Griechenland		

Es werden ausschließlich die oben genannten Länder beliefert. Andere Ziele — u. a. die **Schweiz und Großbritannien** — gehören **bewusst nicht** zum Versandgebiet.

Lieferbenachrichtigung: Pro Bestellung **E-Mail oder Mobilnummer** des Empfängers angeben (E-Mail bevorzugt) — der Logistiker informiert dann aktiv über den Zustelltermin.

DPD-Hinweis: Fehlen E-Mail **und** Telefon, kann DPD einen zweiten Zustellversuch berechnen.

DHL-Packstation: *Straße* = Packstation, *Hausnummer* = Packstationsnummer, *Nachname* = 8-stellige DHL-Postnummer des Empfängers, übrige Felder normal.

4. Die FTP-Integration

Jede Bestellung wird als **separate Textdatei** (*.txt, ANSI-codiert) hochgeladen, benannt nach `IhreKundennummer_IhreBestellnummer.txt`.

HEGA GmbH · Postfach 890230 · 30519 Hannover

4.1 Aufbau der Bestelldatei

```
//Header_dropship
10011_10.10.2012_10001
HEGA GmbH*Am Mittelfelde 45*30519 Hannover
```

```
//Endkunde
Firma;Anrede;Vorname;Name;Zusatz;STR;HausNr.;PLZ;ORT;EMAIL;Mobil;DE
```

```
//Details
6618476      1,00    1,01
//Ende
```

Zeile	Inhalt	Erläuterung
1	//Header_dropship	feste Ausgabe
2	10011_10.10.2012_10001	interne Auftragsnummer ** _** Datum (TT.MM.JJJJ) ** _** HEGA-Kundennummer; Trenner = Unterstrich
3	HEGA GmbH*Am Mittelfelde 45*30519 Hannover	feste HEGA-Anschrift
4	(Leerzeile)	
5	//Endkunde	feste Ausgabe
6	Adressfelder Endkunde	siehe 4.2
7	(Leerzeile)	
8	//Details	feste Ausgabe
9	Artikelzeilen	siehe 4.3
10	//Ende	feste Ausgabe

4.2 Adressfelder Endkunde (Semikolon-getrennt)

#	Feld	Typ / Länge	#	Feld	Typ / Länge
1	Firma	alphanum., 40	7	Hausnummer	alphanum., 10
2	Anrede	alphanum., 40	8	Postleitzahl	alphanum., 10
3	Vorname	alphanum., 40	9	Ort	alphanum., 40
4	Nachname	alphanum., 40	10	E-Mail	alphanum., 50
5	Zusatz	alphanum., 40	11	Mobilnummer	alphanum., 25
6	Straße	alphanum., 30	12	Länderkennzeichen	ISO 3166-1 A2

4.3 Artikelzeilen

Ein Artikeldatensatz hat **insgesamt 40 Zeichen**: Artikelnummer (20, linksbündig) · Menge (10, Dezimal mit **Komma**, rechtsbündig) · Preis (10, Dezimal mit **Komma**, rechtsbündig), mit Leerzeichen auf 40 aufgefüllt.

Wichtig — Verpackungseinheit (VE): Ist die $VE > 1$, übergeben Sie die Menge **in VE, nicht in Stück**. Beispiel: Endkunde bestellt 12×800 g, der Artikel wird als 6×800 g VE geführt → Bestellmenge = **2**. Ungültige Mengen können wir nicht verarbeiten.

Eine VE ist eine **physische Verpackungseinheit, die wir nicht auftrennen** — wir liefern und berechnen ausschließlich ganze Gebinde. **Lose Einzelstücke (z. B. 1 von 6) sind nicht bestellbar**; eine Stückmenge muss daher stets ein Vielfaches der VE sein.

4.4 Abruf & Rückmeldungen

- **Abruf** der hochgeladenen Bestellungen: ca. alle **15 Minuten** (werktags sowie sonntags).
- **Import-Report** je Durchlauf: IMPORT_REPORT_{yyyyMMddHHmmss}.txt.
- **Sendungsdaten** nach Versandabschluss:
IMPORT_DISPATCHINFO_REPORT_{yyyyMMddHHmmss}_{KUNR}.txt (CSV, Semikolon) mit Kundennr., Lieferscheinr., Auftragsnr., Adresse, Artikel, EAN, **gelieferter Menge** und **Sendungsnummern**.
- **Versender an der Paketnummer:** Präfix 00 = DHL, 09 = DPD.
- **Paketzeilen:** Artikelnr. 9999992 (19 % MwSt.) bzw. 9999992_1 (7 % MwSt.); deren Menge = Anzahl Pakete.

HEGA GmbH · Postfach 890230 · 30519 Hannover

```

Übertragungsprotokoll: Kunde:{KUNR} {Kundenname} | Pfad:ABACUS_ORDERS/{KUNR}/
Übertrag gestartet: 01.02.2021 09:35:57
*****
Kopiere:{DATEINAME_1}.txt
*****
//Header_dropship
{BESTELLNUMMER}_29.01.2021_{KUNR}
{Absender Name}*{Strasse Hausnummer}*{PLZ} {ORT}
//Endkunde
Firma;Anrede;Vorname;Name;Zusatz;Straße;93;Postleitzahl;Ort;Mail@Adresse.de;Mobilnummer;Länderkennzeichen
//Details
7000234          10      1,25
12623            10      31,50
9103142          10      5,95
27660            10      13,92
27689            10      7,26
//Ende
*****
Übertrag gestartet: 01.02.2021 09:35:58
*****
Kopiere:{DATEINAME_2}.txt
*****
//Header_dropship
{BESTELLNUMMER}_29.01.2021_{KUNR}
{Absender Name}*{Strasse Hausnummer}*{PLZ} {ORT}
//Endkunde
Firma;Anrede;Vorname;Name;Zusatz;Straße;93;Postleitzahl;Ort;Mail@Adresse.de;Mobilnummer;Länderkennzeichen
//Details
7000234          10      1,25
12623            10      31,50
9103142          10      5,95
27660             2      13,92
27689             1      7,26
//Ende
*****
Übertrag gestartet: 01.02.2021 09:35:58
*****
Kopiere:{DATEINAME_3}.txt
*****
//Header_dropship
{BESTELLNUMMER}_29.01.2021_{KUNR}
{Absender Name}*{Strasse Hausnummer}*{PLZ} {ORT}
//Endkunde
Firma;Anrede;Vorname;Name;Zusatz;Straße;93;Postleitzahl;Ort;Mail@Adresse.de;Mobilnummer;Länderkennzeichen
//Details
9103142          10      5,95
27660             2      13,92
27689             1      7,26
//Ende
*****

```

Beispiel einer Dispatchinfo-Datei

HEGA GmbH · Tiernahrung, Feldsaaten, Rasen

Eichelkampstraße 24 · 30519 Hannover · Telefon 05 11/8 60 43-0 · Fax 05 11/8 60 43-43 oder 87 61 74

Sitz Hannover · Registergericht Hannover HRB 60484 · Geschäftsführer: Uwe Bernd Heine; Tobias Heine, Goran Rose

Steuer-Nr.: 25/220/03873 · Ust-IdNr.: DE225081057 · Bankverbindung: Deutsche Bank Hannover · BLZ 250 700 70 · Kto.-Nr. 030 507 800

Fehlerfall: Nicht verarbeitbare Dateien werden umbenannt in `IMPORT_NICHT_ERFOLGT_FEHLER_IM_INHALT_{dateiname}.txt` (dann **kein** Import-Report; bei Mengenfehlern zusätzlich E-Mail).

4.5 Tägliche Stammdaten per FTP

Artikeldaten — IMPORT_DatenFTPHEGAART.TXT (PIPE-getrennt, ISO-8859-1):

NUMMER|EAN|INTERNETTEXT1|INTERNETTEXT2|INTERNETVBE|INTERNETVERBRGROESSE|
INTERNETWARENGRUPPE|ANZAHLARTVE|LISTENVK|EMPFOHLENERVK|GESAMTBESTAND|
INTERNETHERSTELLER|INTERNETHAUPTGRUPPE|GEWICHT|INTERNETBILDURL|MWST|MARKER
|
LETZTEAENDERUNG|AUSLAUFARTIKEL|BESCHREIBUNG

1. Einmal täglich importieren.
2. Bild-URLs **herunterladen** und vom eigenen Server ausliefern — nicht hotlinken.
3. **AUSLAUFARTIKEL = X** → Artikel im Shop deaktivieren.
4. **Delta prüfen:** neue Artikel kommen hinzu, andere werden ausgelistet (ggf. ohne Markierung).
5. Spalte **Bestand** hier nicht verarbeiten — Bestände kommen separat.

Warenverfügbarkeit — IMPORT_DatenWarenbestand.txt (PIPE-getrennt):

Artikelnr|EAN|Bezeichnung1|Bezeichnung2|BestandStatus|Verfügbar am — Status 1 = verfügbar, 0 = nicht verfügbar, 2 = begrenzt verfügbar.

5. Die API-Integration

Aktuelle, fortlaufend erweiterte REST-API (OpenAPI 3.0). Empfohlen für Echtzeit-Abläufe und für den **vollständigen Rückkanal** (Status, Belege, Sendungsverfolgung).

5.1 Zugang & Authentifizierung

- **Basis-URL:** Test <https://test.api.hega.net> · Live <https://api.hega.net> (gleiche Endpunkte & Datenformate, nur andere Basis-URL — siehe 5.5)
- **Login:** POST /api/Account/login mit Body { "apimail": "...", "password": "...", "kdnmr": "12345" } → accessToken (JWT, **30 Min.** gültig), refreshToken, expiresAt, kundenNummer
- **Token erneuern:** POST /api/Account/refresh-token · **Logout:** POST /api/Account/logout — die Erneuerung setzt einen **noch gültigen** Access-Token im Authorization-Header voraus; nach dessen Ablauf ist ein neuer Login nötig
- Alle weiteren Aufrufe mit Header Authorization: Bearer <accessToken>
- JSON für Requests/Responses; bei einigen Dropshipping-Endpunkten Encoding ISO-8859-1
- Zum Testen Postman, beginnend mit /api/Account/login

API-Versionierung — **wichtig: Lieferschein und Rechnung** laufen **mit** Version im Pfad (/api/v1/...). Alle übrigen Dropshipping-Endpunkte **ohne** v1 (/api/DSOrder/..., /api/POD/..., /api/DSArtikelBestand/...). Genau auf das v1 achten — sonst 404.

5.2 Bestellung anlegen

Empfohlener Ablauf je Bestellung. Sie müssen unsere Artikelstammdaten **nicht** dauerhaft in Ihrem System speichern. Es gibt zwei gleichwertige Wege — wählen Sie nach Bestellvolumen:

Variante A — mit eigener Stammdatenhaltung (schlanker bei vielen Bestellungen):

1. Einmal täglich GET /api/DSArtikelDaten/light (bzw. .../lastchange/{date} für Änderungen) abrufen und Artikelnummer, EAN und anzahlartve in Ihrem System vorhalten.

2. Beim Bestellen je Position aus Ihrem Cache die VE-Menge bestimmen.
3. Bestellung senden (Request unten).

Variante B — ohne eigene Speicherung, gezielter Abruf je Bestellung (wenn Sie unsere Daten bewusst *nicht* vorhalten wollen):

1. **Login** (einmal je Lauf): POST /api/Account/login → Token (30 Min. gültig).
2. **Je Position** des Auftrags: GET /api/DSArtikelDaten/{artikelnummer} abrufen und anzahlartve lesen (HTTP 404 = für Sie nicht bestellbar → Position klären).
3. **Menge bestimmen:** entweder selbst auf VE umrechnen, **oder** die Stückmenge übergeben und je Position mengeneinheit: "Stueck" setzen — dann rechnet HEGA selbst um (Details unten).
4. **Bestellung senden:** POST /api/DSOrder/OrderAdd/ mit allen Positionen des Auftrags.
5. **Mehrere Aufträge:** Schritte 2–4 je Auftrag der Reihe nach abarbeiten; der Login aus Schritt 1 gilt für alle Aufträge des Laufs.

Variante B kostet pro Position einen zusätzlichen GET, hält Ihr System aber frei von HEGA-Stammdaten. Beide Varianten liefern dasselbe Ergebnis — Mengen gehen in **VE** an HEGA (bzw. in Stück mit mengeneinheit: "Stueck").

POST /api/DSOrder/OrderAdd/

Pflichtfelder: bestellnummer (Ihre Referenz, 1–20 Zeichen), vorname, nachname, straße, hausnummer, postleitzahl, ort, laenderkennzeichen (ISO-A2), versender ("DHL" / "DPD"), dsorderzeileto (Array aus artnr + menge in VE). Optional: firma, zusatz, emailadresse, mobilnummer, useCustomerAddress (liefert an die hinterlegte Kundenadresse).

Wichtig bei versender: "DPD" — Kontaktdaten des Empfängers sind Pflicht. DPD benachrichtigt den Empfänger selbst über Zustelltag, Zeitfenster, Umleitung und Ablageort (Predict). Ohne Kontaktinfo lehnt DPD das Versandetikett ab und die Sendung bleibt bei uns liegen. Senden Sie deshalb bei DPD **entweder** eine gültige emailadresse **oder** eine

mobilnummer (internationales Format, z. B. +4915112345678) — eines von beiden genügt. Fehlt beides, weisen wir die Bestellung sofort mit HTTP 400 und errorCode: "MISSING_CONTACT_DATA" zurück; sie wird **nicht** angelegt. Bei versender: "DHL" bleiben beide Felder optional.

Bei useCustomerAddress: true verwenden wir die E-Mail aus Ihrem Kundenstamm; eine mitgeschickte mobilnummer wird zusätzlich übernommen.

Optional je Position: mengeneinheit ("VE" oder "Stueck"). Standard ist "VE" — dann ist menge wie bisher die Verkaufseinheiten-Menge. Mit "Stueck" geben Sie menge in Einzelstücken an, und HEGA rechnet selbst auf VE um ($\text{menge} / \text{anzahlartve}$); ist die Stückmenge kein Vielfaches der VE, antworten wir mit INVALID_QUANTITY. Details und Beispiel in Kapitel 4.3 bzw. 12.5.

Erfolg (HTTP 200): Bestätigungstext mit Ihrer Bestellnummer. Identifikation in allen Folgeabfragen über **Ihre Bestellnummer**. Häufige Fehlercodes (errorCode): ARTICLE_NOT_FOUND, ARTICLE_INACTIVE, DUPLICATE_ORDER (gleiche Bestellnummer < 24 h), API_NOT_ENABLED, INVALID_ADDRESS, INVALID_QUANTITY, MISSING_CONTACT_DATA (bei versender: "DPD" fehlt eine gültige emailadresse bzw. mobilnummer).

Vorab-Prüfung (Dry-Run) & Sammelfehler. Mit POST /api/DSOrder/OrderValidate/ (gleicher Request wie OrderAdd) prüfen Sie eine Bestellung – Adresse, Artikel, Mengen/VE – **ohne** sie anzulegen. Die Antwort liefert valid (true/false) und unter errors **alle** gefundenen Probleme auf einmal (je mit Positionsnummer). Ideal zum Testen Ihres Mappings – funktioniert bewusst auch **vor** der finalen Freischaltung. Schlägt eine echte OrderAdd an Positionen fehl, enthält die HTTP-400-Antwort zusätzlich dasselbe errors-Array (alle fehlerhaften Positionen); errorCode und details auf oberster Ebene bleiben der **erste** Fehler – bestehende Integrationen laufen unverändert weiter. Der B2B-Pfad /api/Order bietet OrderValidate und das errors-Sammelfeld analog.

Code-Beispiele (Login + Bestellung). Anmeldung und eine Bestellung über POST /api/DSOrder/OrderAdd/. Soll an die **zur Kundennummer hinterlegte Adresse** geliefert

werden (statt einer mitgeschickten Empfängeradresse), genügt "useCustomerAddress": true
— die Adressfelder entfallen dann.

cURL

1) Login -> accessToken

```
TOKEN=$(curl -s -X POST "https://test.api.hega.net/api/Account/login?api-version=1.0" \
-H "Content-Type: application/json" \
-d '{"apimail":"filiale@example.com","password":"****","kdnr":"1234567"}' \
| sed -n 's/.*"accessToken":\"([^\"]*)\".*/\1/p')
```

2) Bestellung

```
curl -X POST "https://test.api.hega.net/api/DSOrder/OrderAdd?api-version=1.0" \
-H "Authorization: Bearer $TOKEN" -H "Content-Type: application/json" \
-d '{"bestellnummer":"BESTELL-2026-0001",
  "vorname":"Maria","nachname":"Schmidt",
  "straße":"Musterweg","hausnummer":"42a",
  "postleitzahl":"50667","ort":"Köln",
  "laenderkennzeichen":"DE","versender":"DHL",
  "dsorderzeileto":{"artnr":"12345","menge":2}}'
```

C# (.NET, HttpClient)

```
var http = new HttpClient { BaseAddress = new Uri("https://test.api.hega.net") };

var login = await http.PostAsJsonAsync("/api/Account/login?api-version=1.0",
    new { apimail = "filiale@example.com", password = "****", kdnr = "1234567" });
var token = (await login.Content.ReadFromJsonAsync<Dictionary<string, object>>())!["accessToken"].ToString();

http.DefaultRequestHeaders.Authorization = new("Bearer", token);
var order = new {
    bestellnummer = "BESTELL-2026-0001",
    vorname = "Maria", nachname = "Schmidt",
    straÙe = "Musterweg", hausnummer = "42a",
    postleitzahl = "50667", ort = "Köln",
    laenderkennzeichen = "DE", versender = "DHL",
    dsorderzeileto = new[] { new { artnr = "12345", menge = 2 } }
};
```

```
var resp = await http.PostAsJsonAsync("/api/DSOrder/OrderAdd/?api-version=1.0", order);
Console.WriteLine(await resp.Content.ReadAsStringAsync());
```

Python (requests)

```
import requests
```

```
base = "https://test.api.hega.net"
```

```
token = requests.post(f'{base}/api/Account/login', params={"api-version": "1.0"},
                      json={"apimail": "filiale@example.com", "password": "****", "kdnr": "1234567"}
                      ).json()["accessToken"]
```

```
order = {
    "bestellnummer": "BESTELL-2026-0001",
    "vorname": "Maria", "nachname": "Schmidt",
    "straße": "Musterweg", "hausnummer": "42a",
    "postleitzahl": "50667", "ort": "Köln",
    "laenderkennzeichen": "DE", "versender": "DHL",
    "dsorderzeileto": [{"artnr": "12345", "menge": 2}],
}
r = requests.post(f'{base}/api/DSOrder/OrderAdd/', params={"api-version": "1.0"},
                  headers={"Authorization": f'Bearer {token}'}, json=order)
```

```
print(r.status_code, r.text)
```

Java (11+, java.net.http)

```
var http = HttpClient.newHttpClient();
```

```
var login = "{\n  \"apimail\": \"filiale@example.com\",\n  \"password\": \"****\",\n  \"kdnr\": \"1234567\"\n}";
```

```
var lr = http.send(HttpRequest.newBuilder()
    .uri(URI.create("https://test.api.hega.net/api/Account/login?api-version=1.0"))
    .header("Content-Type", "application/json")
    .POST(HttpRequest.BodyPublishers.ofString(login)).build(),
    HttpResponse.BodyHandlers.ofString());
```

```
String token = /* accessToken aus lr.body() parsen, z. B. mit Jackson */ "";
```

```
var body = "{\n  \"bestellnummer\": \"BESTELL-2026-0001\",\n  \"vorname\": \"Maria\",\n  \"nachname\": \"Schmidt\",\n  \"straße\": \"Musterweg\",\n  \"hausnummer\": \"42a\",\n  \"postleitzahl\": \"50667\",\n  \"ort\": \"Köln\",\n  \"laenderkennzeichen\": \"DE\",\n  \"versender\": \"DHL\",\n  \"dsorderzeileto\": [{\n    \"artnr\": \"12345\",\n    \"menge\": 2\n  }]\n}";
```

```
var resp = http.send(HttpRequest.newBuilder()
    .uri(URI.create("https://test.api.hega.net/api/DOrder/OrderAdd/?api-version=1.0"))
    .header("Authorization", "Bearer " + token).header("Content-Type", "application/json")
    .POST(HttpRequest.BodyPublishers.ofString(body)).build(),
    HttpResponse.BodyHandlers.ofString());
```

```
System.out.println(resp.body());
```

PHP (8.x / 7.4, nur ext-curl — kein Composer nötig)

```
<?php
```

```
$basis = 'https://test.api.hega.net';
```

```
function hega(string $url, array $daten, ?string $token = null): array {
    $header = ['Content-Type: application/json'];
    if ($token !== null) { $header[] = 'Authorization: Bearer ' . $token; }
    $ch = curl_init($url);
    curl_setopt_array($ch, [
        CURLOPT_POST => true,
        CURLOPT_POSTFIELDS => json_encode($daten), // Umlaute werden korrekt als \uXXXX kodiert
        CURLOPT_HTTPHEADER => $header,
        CURLOPT_RETURNTRANSFER => true,
    ]);
    $antwort = curl_exec($ch);
    $status = curl_getinfo($ch, CURLINFO_RESPONSE_CODE);
    curl_close($ch);
    return [$status, json_decode($antwort, true)];
}
```

// 1) Login -> accessToken

```
[, $login] = hega("$basis/api/Account/login?api-version=1.0", [
    'apimail' => 'filiale@example.com', 'password' => '***', 'kdnnr' => '1234567',
]);
$token = $login['accessToken'];
```

// 2) Bestellung

```
[$status, $ergebnis] = hega("$basis/api/DOrder/OrderAdd/?api-version=1.0", [
    'bestellnummer' => 'BESTELL-2026-0001',
    'vorname' => 'Maria', 'nachname' => 'Schmidt',
    'straße' => 'Musterweg', 'hausnummer' => '42a',
```

```
'postleitzahl' => '50667', 'ort' => 'Köln',
'laenderkennzeichen' => 'DE', 'versender' => 'DHL',
'dsorderzeileto' => [['artnr' => '12345', 'menge' => 2]],
], $token);

echo $status, ' ', is_string($ergebnis) ? $ergebnis : json_encode($ergebnis), PHP_EOL;
JavaScript (Node.js 18+, eingebautes fetch — kein npm install nötig)

const basis = 'https://test.api.hega.net';

async function hega(url, daten, token) {
  const kopf = { 'Content-Type': 'application/json' };
  if (token) { kopf.Authorization = 'Bearer ' + token; }
  const antwort = await fetch(url, {
    method: 'POST',
    headers: kopf,
    body: JSON.stringify(daten), // echtes UTF-8 - "straße" geht unverändert durch
  });
  const text = await antwort.text();
  try { return [antwort.status, JSON.parse(text)]; }
  catch { return [antwort.status, text]; }
}

// 1) Login -> accessToken
const [, login] = await hega(`${basis}/api/Account/login?api-version=1.0`, {
  apimail: 'filiale@example.com', password: '***', kdnnr: '1234567',
});
const token = login.accessToken;

// 2) Bestellung
const [status, ergebnis] = await hega(`${basis}/api/DSOrder/OrderAdd/?api-version=1.0`, {
  bestellnummer: 'BESTELL-2026-0001',
  vorname: 'Maria', nachname: 'Schmidt',
  'straße': 'Musterweg', hausnummer: '42a',
  postleitzahl: '50667', ort: 'Köln',
  laenderkennzeichen: 'DE', versender: 'DHL',
  dsorderzeileto: [{ artnr: '12345', menge: 2 }],
}, token);
```

`console.log(status, ergebnis);`

Vollständige Referenzclients zum Download — PHP, Python und Node.js:

Sprache	Download	Voraussetzungen
PHP	https://api.hega.net/HEGA-PHP-Client.php	PHP 7.4+, nur ext-curl/ext-json — kein Composer
Python	https://api.hega.net/hega_api_client.py	Python 3.8+, nur Standardbibliothek — kein pip install
Node.js	https://api.hega.net/hega-api-client.cjs	Node.js 18+, nur eingebautes fetch — kein npm install
TypeScript	https://api.hega.net/hega-api-client.d.cts	Typdatei zum Node.js-Client — einfach danebenlegen, keine Konfiguration nötig

(Auf der Test-Umgebung dieselben Dateinamen unter <https://test.api.hega.net/>.)

Die Snippets oben zeigen nur den Kern. Die Download-Dateien sind lauffähige Clients und erledigen zusätzlich das, was eine Produktiv-Anbindung braucht: automatische Token-Erneuerung und Neuansmeldung, Auswertung **beider** Fehlerschemas (siehe 5.4), Wiederholung bei HTTP 429, die sichere OrderAdd-Variante nach Timeout (Statusabfrage statt Blind-Wiederholung) sowie das korrekte Durchblättern der Artikeldaten. Alle drei sind inhaltlich identisch aufgebaut, bewusst ohne Fremdbibliotheken, damit sie auch in abgeschotteten Umgebungen sofort laufen. Sie dürfen die Dateien frei anpassen — sie sind als Startpunkt gedacht, nicht als fertige Integration.

Hinweis für PHP: `json_encode()` maskiert Umlaute standardmäßig als `\uXXXX` — das ist gültiges JSON und wird von uns korrekt gelesen; das Feld `straße` funktioniert damit ohne Zusatzflag. Voraussetzung ist, dass Ihre Strings **UTF-8** sind. Lesen Sie Ihre Daten aus einer

ISO-8859-1-Quelle (z. B. ältere CSV-Exporte), vorher mit `mb_convert_encoding(...)` umwandeln — sonst liefert `json_encode()` false und der Request geht leer raus.

Hinweis für Node.js: Alle Aufrufe sind **asynchron** — ohne `await` erhalten Sie ein Promise statt des Ergebnisses. Das Beispiel oben nutzt `await` auf oberster Ebene; das setzt eine `.mjs`-Datei bzw. `"type": "module"` in Ihrer `package.json` voraus. Andernfalls legen Sie den Code in eine `async function`. Der Referenzclient trägt bewusst die Endung `.cjs`, damit er in **beiden** Welten unverändert läuft: `require('./hega-api-client.cjs')` in klassischen CommonJS-Projekten und `import { HegaApiClient } from './hega-api-client.cjs'` in Projekten mit `"type": "module"`.

5.3 Endpunkt-Referenz

Bestellung & Status

Zweck	Endpunkt
Bestellung anlegen	POST <code>/api/DSOrder/OrderAdd/</code>
Bestellung vorab prüfen (Dry-Run, ohne Anlage)	POST <code>/api/DSOrder/OrderValidate/</code>
Status einer Bestellung	GET <code>/api/DSOrder/Customer/{bestellnummer}</code>
Status-Sammelabfrage (Zeitraum, max. 5 Tage)	POST <code>/api/DSOrder/Status/</code>
Bestellinfo (LS-Nr + Bestellnr)	GET <code>/api/DSOrderInfo/{hega_lsnr},{bestellnr}</code>

Belege & Sendungsverfolgung (Rückkanal)

Zweck	Endpunkt
Lieferschein-Detail	GET <code>/api/v1/DeliveryNote/{lsnr}</code>
Lieferschein PDF	GET <code>/api/v1/DeliveryNote/{lsnr}/pdf (?type=freight = Frachtschein)</code>

Zweck	Endpunkt
Lieferschein CSV	GET /api/v1/DeliveryNote/{lsnr}/csv
Lieferschein MDE (EAN + Stück)	GET /api/v1/DeliveryNote/{lsnr}/mde
Rechnung-Detail	GET /api/v1/Invoice/{renr}
Rechnung PDF / CSV	GET /api/v1/Invoice/{renr}/pdf · .../csv
Rechnungs-Sammelexport / Offene Posten	GET /api/v1/Invoice/export/csv · .../overdue
Zustellnachweis (POD)	GET /api/POD/Customer/{bestellnummer} · GET /api/POD/{lsnr}

Rückstände / Reservierungen

Zweck	Endpunkt
Offene Rückstände ab Datum	GET /api/DSReservierungen/fromdate/{date}
Rückstände zu einem Lieferschein	GET /api/DSReservierungen/{lsnr}
Stornierte/abgearbeitete Rückstände (max. 5 Tage)	POST /api/DSReservierungen/StatusCanceled/

Artikel, Bestand & Bilder

Zweck	Endpunkt
Warenbestand-Änderungen ab Datum	GET /api/DSArtikelBestand/lastchange/{date}
Warenbestand Einzelartikel	GET /api/DSArtikelBestand/{nummer}

Zweck	Endpunkt
Artikeldaten (kompakt) / Änderungen ab Datum	GET /api/DSArtikelDaten/light · .../lastchange/{date}
Artikeldaten Einzelartikel (inkl. anzahlartve für VE-Umrechnung)	GET /api/DSArtikelDaten/{nummer} · kompakt .../light/{nummer}
Nicht mehr bestellbare Artikel ab Datum	GET /api/DSArtikelDaten/ArticleNoLongerAvailableForOrder/{date}
Produktbild (JPEG) / Bild-Info / geänderte Bilder	GET /api/Bilder/{EAN} · .../Info/{EAN} · .../Liste/{AenderungAb}
Sammelinfos ab Datum (max. 90 Tage)	GET /api/DS_SAMMELINFO/ab/{date}

Für reine Dropshipping-Abläufe ist durchgängig die **DSOrder**-Familie maßgeblich.

5.4 Technische Hinweise für die Anbindung

Praxisrelevante Punkte, die typische Integrationsfehler vermeiden:

- **Freischaltung pro Konto:** Einzelne Funktionen (Bestellungen, Warenbestand, Artikeldaten) müssen für Ihre Kundennummer freigeschaltet sein. Ein **HTTP 403** bedeutet i. d. R. nicht „Token ungültig“, sondern „Funktion nicht freigeschaltet“ → bitte bei HEGA anfragen.
- **Datum & Uhrzeit:** Datums-Parameter ({date}) im Format **yyyy-MM-dd** (z. B. 2026-06-02). Zeitstempel in Antworten sind **MEZ ohne Zeitzone-Suffix** (kein Z / +01:00) — bei Verarbeitung aus anderen Zeitzone berücksichtigen.
- **Zeitfenster:** Lieferschein- und Rechnungs-Endpunkte liefern nur Belege der **letzten 12 Monate** (ältere → HTTP 404, auch bei direkter Nummer). Der Status Invoiced ist

im Status-Endpunkt nur ca. **3 Monate** sichtbar — danach über die Rechnungs-Endpunkte abrufen.

- **Paginierung:** Listen-Endpunkte (DeliveryNote, Invoice) paginieren über `pageIndex` (ab 1) und `pageSize` (Default 10); die Antwort enthält `totalPages`, `hasNextPage` usw.
- **Paginierung der Artikeldaten:** GET `/api/DSArtikelDaten, .../light` und `.../lastchange/{date}` akzeptieren optional `page` (ab 1) und `pageSize` (Default 500, max 5000); ohne diese Parameter wird unverändert das komplette Sortiment geliefert. Gesamtzahl im Response-Header `X-Total-Count` (nur Seite 1). **Wichtig:** Durch die kundenspezifische Freischaltung kann eine Seite **weniger als `pageSize`** Artikel enthalten — nicht bei einer „kurzen“ Seite abbrechen, sondern `page` hochzählen, bis die Summe der gelieferten Artikel `X-Total-Count` erreicht.
- **Feld-Regeln:** Menge je Bestellzeile **1–10.000**. versender und laenderkennzeichen sind **case-sensitive** Enum-Werte ("DHL" / "DPD", Land in Großbuchstaben).
- **DPD braucht Empfänger-Kontaktdaten:** Bei versender: "DPD" ist eine gültige emailadresse **oder** eine mobilnummer Pflicht — DPD benachrichtigt den Empfänger selbst und lehnt Sendungen ohne Kontaktinfo ab. Fehlt beides: HTTP 400 mit `MISSING_CONTACT_DATA`. OrderValidate (Dry-Run) meldet denselben Fehler vorab.
- **Token-Handling:** Access-Token 30 Min. gültig, **keine Kulanzzzeit** — Refresh proaktiv ~1–2 Min. vor Ablauf; nach vollständigem Ablauf ist ein neuer Login nötig. **Achtung — nicht auf expiresAt als Ablauf des Access-Tokens steuern:** dieses Feld gehört zum *Refresh*-Token und liegt später. Maßgeblich für den Access-Token ist der Claim **exp** im JWT selbst (bzw. schlicht „Login-Zeitpunkt + 30 Min.“). Wer nach expiresAt erneuert, läuft zwischenzeitlich in HTTP 401. Ebenfalls möglich und unkritisch: Die Erneuerung schlägt fehl, weil die Sitzung serverseitig nicht mehr bekannt ist (z. B. nach einer Wartung) — dann einfach neu anmelden. Planen Sie diesen Rückfall von Anfang an ein.

- **Fehlerauswertung — es gibt zwei Antwortformate.** Die Bestell-Endpunkte (/api/DSOrder und /api/Order) liefern ein strukturiertes Fehlerobjekt errorCode + message (+ details, bei Positionsfehlern zusätzlich errors[]). **Vorgelagert** ist jedoch die formale Feldprüfung: Verstöße gegen Pflichtfeld- und Längenregeln (fehlender vorname, unbekannter Enum-Wert ...) beantworten wir im Standardformat **ohne** errorCode — { "title": "One or more validation errors occurred.", "status": 400, "errors": { "<Feld>": [...] } }. Ihre Fehlerbehandlung muss **beide** Formen abfangen; sonst sehen Sie bei einem Mapping-Fehler nur einen nackten HTTP 400. Alle übrigen Endpunkte (Lieferschein, Rechnung, Bestand ...) geben einfache Texte zurück → dort den **HTTP-Statuscode** auswerten.
- **Fehlertexte sind bewusst allgemein gehalten.** Bei einem serverseitigen Problem antworten wir mit einem knappen Text wie „Fehler beim Abrufen der Rechnungen.“ — ohne technische Details. Das ist Absicht: technische Meldungen könnten Rückschlüsse auf unsere Systeme zulassen. Die vollständige Diagnose steht in unserem Server-Protokoll. Wenn Sie einen solchen Fehler reproduzierbar sehen, melden Sie ihn bitte mit **Zeitpunkt (inkl. Uhrzeit), Ihrer Kundennummer und dem aufgerufenen Endpunkt** an it@hega.net — damit finden wir den passenden Eintrag sofort.
- **Datei-Encoding:** Lieferschein-/Rechnungs-CSV und MDE sind **ISO-8859-1**; einzige Ausnahme ist der Rechnungs-Sammelexport (/api/v1/Invoice/export/csv) — dieser ist **UTF-8**.
- **Beim Shop-Import beachten:** beschreibung kann **HTML** enthalten (Zeilenumbrüche als
). empfohlenervk ist **Brutto inkl. MwSt.**, listenvk ist Netto. gesamtbestand in den Artikeldaten ist immer 0 — echte Bestände nur über DSArtikelBestand.
- **Wiederholung nach Timeout:** Das Anlegen einer Bestellung ist nicht vollständig idempotent. Bei Timeout / 500 bitte zuerst per GET /api/DSOrder/Customer/{bestellnummer} prüfen, ob die Bestellung doch angekommen ist, bevor Sie erneut senden — sonst drohen Doubletten.

5.5 Test & Abnahme

Binden Sie die Schnittstelle **zuerst gegen die Test-Umgebung** an und schalten erst nach erfolgreicher Abnahme auf Live um.

- **Testumgebung:** <https://test.api.hega.net> — gleiche Endpunkte und Datenformate wie Live, nur andere Basis-URL. Nach der Abnahme einfach auf <https://api.hega.net> umstellen.
- **Testbestellungen kennzeichnen:** Verwenden Sie ein erkennbares Präfix in der bestellnummer (z. B. TEST-...).

Schritt 1 — Artikeldaten abrufen (Pflicht vor dem Bestellen): Bevor Sie (Test-)Bestellungen senden, rufen Sie unsere **aktuellen Artikeldaten** ab und übernehmen sie in Ihr System:

- GET /api/DSArtikelDaten/light (bzw. .../lastchange/{date} für Änderungen).
- Nur Artikel, die in diesen Daten **existieren und aktiv** sind, sind bestellbar — andernfalls ARTICLE_NOT_FOUND bzw. ARTICLE_INACTIVE.
- Aktualisieren Sie die Artikeldaten **regelmäßig (täglich)**, damit neue/ausgelistete Artikel und Preise stimmen.

Schritt 2 — Verpackungseinheit (VE) beachten (häufigste Abnahme-Hürde): Aus den Artikeldaten ergibt sich die **VE** je Artikel. Mengen werden **in VE** übergeben, nicht in Stück (siehe Kapitel 4.3).

- Beispiel: Artikel wird als 6 × 800 g geführt, Endkunde bestellt 12 × 800 g → übergebene Menge = **2 (VE)**, nicht 12 (Stück).
- Prüfen Sie genau diesen Punkt in der Abnahme — falsche VE-Mengen sind die häufigste Fehlerquelle.

Abnahme-Checkliste:

1. Login gegen test.api.hega.net erfolgreich (Token erhalten).
2. Artikeldaten abrufbar und im eigenen System übernommen.
3. Testbestellung (TEST-...) wird akzeptiert (HTTP 200).
4. Status der Testbestellung abrufbar; ggf. Lieferschein/Tracking im Rückkanal sichtbar.
5. VE-Mengen korrekt (Stichprobe).
6. **Go-Live:** Basis-URL auf https://api.hega.net umstellen.

6. Nach der Bestellung — Status & Belege abrufen

Die sechs Stationen aus Kapitel 2.2 — hier mit den konkreten Feldern und dem empfohlenen Abruf.

6.1 Status & Felder

GET /api/DSOrder/Customer/{bestellnummer} gibt den aktuellen Stand zurück (bei Teillieferung mehrere Einträge):

Station	ordersStatus
Eingegangen	Arrived (1)
Im Auftragspool	InTheOrderPool (2)
In Bearbeitung	InProgress (3)
Gepackt	Packed (4)
Versendet	Dispatched (5)

Station	ordersStatus
Berechnet	Invoiced (6)

Die internen Nummern lsnr (Lieferschein) und renr (Rechnung) füllen sich erst, sobald die jeweilige Station erreicht ist (vorher 0). **Ab Dispatched** enthält jede Position unter dsorderzeileresultdetails die Felder trackingid (Sendungsnummer) und url (Tracking-Link). Pro Position zeigt menge (geliefert) vs. orderMenge (bestellt) etwaige Abweichungen.

6.2 Empfohlene Abruf-Strategie (Polling)

Die API arbeitet nach dem **Abruf-Prinzip** — Sie fragen ab, wenn Sie Informationen brauchen (automatische Benachrichtigungen/Webhooks sind in Planung, aber noch nicht verfügbar):

1. **1× täglich:** versendete Bestellungen über POST /api/DOrder/Status/ (Status Dispatched) → alle Sendungsnummern + Links auf einmal.
2. **1× täglich:** Rückstände über GET /api/DSReservierungen/fromdate/{date}.
3. **Bei Bedarf:** Bestandsänderungen über GET /api/DSArtikelBestand/lastchange/{date}.

Für die Sendungsverfolgung gegenüber dem Endkunden genügt **ein Abruf pro Tag**.

7. FTP oder API — was wann?

Aspekt	FTP	API
Nutzung	Stapelbetrieb (Dateien)	Echtzeit (HTTP)
Authentifizierung	(S)FTP-Zugangsdaten	JWT (Bearer-Token)

Aspekt	FTP	API
Datenformat	TXT/CSV (ANSI/ISO-8859-1)	JSON
Stärke	große Mengen, geplant	schnell, granular, vollständiger Rückkanal
Verzögerung	~15 Min.	sofort
Belege/Tracking/POD	Dispatchinfo-Datei	Lieferschein-/Rechnungs-PDF, POD, Tracking

Empfehlung: Bestellungen per FTP **oder** API einreichen; für **Status, Belege und Sendungsverfolgung** ist die **API** der vollständigere und aktuellere Weg (Kapitel 6).

8. Rückstandsfunktion (optional)

Optionales Feature — standardmäßig nicht aktiv. Die Rückstandsfunktion wird nur auf ausdrücklichen Wunsch für Ihre Kundennummer freigeschaltet (Kontakt: EDV@hega.net). Ohne Aktivierung werden nicht lieferbare Mengen **nicht** automatisch nachgeliefert.

Ist sie aktiviert, gilt für Mengen, die nicht vollständig lieferbar sind:

- Nicht verfügbare Artikel werden **in Rückstand** genommen.
- Die Bestellung wird mit allen verfügbaren Artikeln **trotzdem ausgeliefert**.
- Bei Wareneingang lösen wir den Rückstand **automatisch** aus und liefern nach.
- Dabei entstehen **doppelte Versandkosten** — bitte in Ihrer Logik berücksichtigen.

Erkennbar an gelieferter vs. bestellter Menge (API: *menge* vs. *orderMenge*; FTP: Feld *Menge* in der Dispatchinfo).

9. Frachtschein — eigenes Logo & Fußzeile

Auf Wunsch hinterlegen wir zu Ihrer HEGA-Kundennummer ein **Firmenlogo** und einen individuellen **Fußzeilentext** auf dem Frachtschein: Logo jpg/jpeg (Ausdruck S/W), Fußzeile bei Schriftgröße 10 bis **750 Zeichen** (Times New Roman). Zusendung an EDV@hega.net.

HEGA GmbH · Postfach 890230 · 30519 Hannover



Frachtschein

Lieferscheinnr:
Lieferdatum:
Seite:

Frachtscheinlogo Test. Kunde: 10666

>Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eimod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eimod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eimod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. A9

Beispiel-Frachtschein mit Kundenlogo

10. Stornierung & Support

Eine Bestellung kann derzeit **nicht** über die API storniert werden.

- **Per E-Mail** an support@ftphega.de — Betreff z. B. *Reklamation / Frage / Warenverfügbarkeit*. Inhalt:
 Kundenummer:
 Auftragsnummer:
 HEGA Lieferscheinnummer:
 Ihr Anliegen ...
- **Alternativ im Web-Portal** <https://portal.hega.net> → Bereich *Reservierungen*:
 Reservierungen oder noch nicht im Prozess befindliche Bestellungen selbst stornieren.

11. Glossar

Begriff	Kurz erklärt
Dropshipping	HEGA liefert die bestellte Ware direkt an Ihren Endkunden — in Ihrem Namen.
FTP	Automatisches Dateipostfach: Dateien werden hoch- und heruntergeladen.
API	Direkte Programm-zu-Programm-Verbindung in Echtzeit.
Endpunkt	Eine einzelne „Adresse“ der API für eine bestimmte Aufgabe (z. B. Status abrufen).
Lieferschein	Beleg, der dem Paket beiliegt — listet die gelieferte Ware (ohne Preise).
Frachtschein	Versandbeleg des Paketdiensts mit aufgedruckter Sendungsnummer.

Begriff	Kurz erklärt
Sendungsnummer / Tracking	Nummer + Link, mit dem der Empfänger sein Paket verfolgen kann.
POD (Proof of Delivery)	Zustellnachweis: ob, wann und an wen zugestellt wurde.
Rückstand	Nicht sofort lieferbarer Artikel, der später nachgeliefert wird.
VE (Verpackungseinheit)	Gebinde, in dem ein Artikel gehandelt wird (z. B. 6 × 800 g). Wird nicht aufgetrennt — Lieferung nur in ganzen Gebinden, keine losen Einzelstücke.
Token (JWT)	Zeitlich begrenzter „Schlüssel“, mit dem sich Ihr System bei der API anmeldet.
Polling	Regelmäßiges aktives Nachfragen („Gibt es etwas Neues?“).
Status	Eine der sechs Stationen, die eine Bestellung durchläuft.

12. Anhang — Für SAP-Anwender

Wenn Sie aus **SAP** heraus an HEGA anbinden, ordnet diese Gegenüberstellung unsere Begriffe Ihrem gewohnten Denkraum zu. Ihre Bestellnummer (z. B. 45...) reichen wir in **jeder** Rückmeldung als bestellnummer wieder durch — sie ist Ihr durchgängiger Schlüssel.

12.1 Begriffe HEGA ↔ SAP

HEGA	SAP-Entsprechung
Bestellung anlegen (OrderAdd)	Bestellung / Purchase Order (ME21N) , Belegnr. = Ihr bestellnummer

HEGA	SAP-Entsprechung
HEGA-Artikelnummer / EAN	Material (MATNR) bzw. Lieferanten-Materialnr. / EAN/GTIN
Menge in VE	Bestellmengeneinheit / VPE
Status <i>Eingegangen / Im Auftragspool</i>	Auftragsbestätigung (Beleg angenommen, vgl. ORDRSP)
Lieferschein / Status <i>Versendet</i> + Tracking	Lieferavis / ASN (vgl. DESADV) + Versandinfo
Lieferschein als MDE/CSV (EAN + Stück)	Vorlage für den Wareneingang (MIGO) gegen die Bestellung
Rückstand / Reservierungen	Offene Bestellmenge / Restmenge der Bestellung
Rechnung (PDF/CSV) / Status <i>Berechnet</i>	Rechnung → Rechnungsprüfung (MIRO), vgl. INVOIC

12.2 Integration: REST/FTP statt IDoc

Bitte beachten Sie einen wichtigen Unterschied zur klassischen SAP-EDI-Welt:

HEGA stellt die Schnittstelle als **REST-API (JSON)** und **FTP (CSV/TXT)** bereit — **nicht** als IDoc. Es gibt also keine fertigen ORDERS/ORDRSP/DESADV/INVOIC-Nachrichten.

Die Übersetzung übernimmt **Ihre Middleware** (z. B. SAP PI/PO, SAP Integration Suite / Cloud Integration oder Ihr EDI-Dienstleister). Typische Zuordnung:

Ihr SAP-Vorgang	Aufruf bei HEGA
Bestellung ausgeben (ORDERS)	POST /api/DSOrder/OrderAdd/ <i>oder</i> FTP-Bestelldatei
Bestätigung/Status einlesen (→ ORDRSP)	GET /api/DSOrder/Customer/{bestellnummer}
Lieferavis/ASN erzeugen (→ DESADV)	Lieferschein + Tracking aus Status Dispatched / .../DeliveryNote/{lsnr}
Wareneingang buchen	Lieferschein-MDE/CSV (EAN + Stück)
Rechnung verbuchen (→ INVOIC)	.../Invoice/{renr} (PDF/CSV)

Vor dem Go-Live klären: Erwartet Ihr Haus IDoc-Nachrichten, oder kann Ihre Middleware direkt REST/FTP anbinden? Das bestimmt Ihren Integrationsaufwand maßgeblich. Sprechen Sie uns dazu gern an (EDV@hega.net).

12.3 Typischer Ablauf aus SAP-Sicht

Das Dropshipping-Szenario entspricht in SAP dem **Streckengeschäft / Third-Party** (Direktlieferung an Ihren Endkunden). Die sechs HEGA-Stationen ordnen sich Ihrem Prozess so zu:

HEGA-Station	Bei Ihnen in SAP	Nachricht / Beleg
Bestellung an HEGA	Bestellung anlegen & ausgeben (ME21N)	ORDERS (Bestellung)
Eingegangen / Im Auftragspool	Bestätigung einlesen	ORDRSP (Auftragsbestätigung)
In Bearbeitung / Gepackt	— (Bearbeitung beim Lieferanten)	—

HEGA-Station	Bei Ihnen in SAP	Nachricht / Beleg
Versendet	Lieferavis/ASN verarbeiten	DESADV (Lieferavis)
Wareneingang (oft statistisch)	Wareneingang buchen (MIGO) gegen die Bestellung	Materialbeleg
Berechnet	Rechnung prüfen & buchen (MIRO)	INVOIC (Rechnung)

Da die Ware direkt an Ihren Endkunden geht, ist der Wareneingang bei Ihnen je nach Prozess ein logischer/statistischer Schritt.

12.4 Partnerrollen & Adressen

Wichtig für die Adresszuordnung: **Ware** geht an Ihren Endkunden, die **Rechnung** an Sie.

SAP-Partnerrolle	Wer	Bei HEGA
Lieferant (LF / Vendor)	HEGA	—
Auftraggeber & Rechnungsempfänger (AG / RE)	Sie	Ihre HEGA-Kundennummer
Warenempfänger (WE / Ship-to)	Ihr Endkunde	Adressfelder der Bestellung (vorname, nachname, straße, ...)

Mit useCustomerAddress = true liefern wir stattdessen an Ihre **eigene** hinterlegte Kundenadresse (statt an den Endkunden) — nützlich z. B. für Mustersendungen.

12.5 Mengeneinheit / VE

In SAP trägt die Bestellposition eine **Bestellmengeneinheit (BME)** mit Umrechnung zur **Basismengeneinheit**. HEGA erwartet die Menge standardmäßig in der **Verpackungseinheit (VE)** des Artikels — nicht in Stück (siehe Kapitel 4.3).

Sie haben zwei Wege, das abzubilden:

Variante A — Umrechnung in SAP (Standard). Pflegen Sie die **Mengeneinheiten-Umrechnung** im SAP-Materialstamm so, dass die an HEGA übergebene Menge der VE entspricht. Beispiel: Artikel wird als 6×800 g geführt, Endkunde bestellt 12×800 g → übergebene Menge = **2** (VE), nicht 12 (Stück).

Variante B — Umrechnung durch HEGA (mengeneinheit: "Stueck"). Wenn Sie die VE-Umrechnung **nicht** in SAP pflegen möchten, übergeben Sie die Menge in Einzelstücken und setzen je Position mengeneinheit auf "Stueck". HEGA rechnet dann selbst mit dem Feld anzahlartve des Artikels um:

```
"dsorderzeileto": [
  { "artnr": "69315", "menge": 48, "mengeneinheit": "Stueck" }
]
```

Bei anzahlartve = 24 werden daraus **2 VE**. Ist die Stückmenge **kein** Vielfaches der VE (z. B. 50 bei VE 24), weisen wir die Position mit INVALID_QUANTITY zurück — so fällt eine falsche Mengenangabe sofort auf, statt zu einer Fehllieferung zu führen. Lassen Sie mengeneinheit weg oder setzen es auf "VE", gilt unverändert Variante A.

Bitte beachten: mengeneinheit: "Stueck" ist **nur eine Eingabe-Hilfe** — HEGA rechnet die Stückzahl auf VE um. Eine VE ist eine **physische Verpackungseinheit, die nicht aufgetrennt wird**; geliefert und berechnet werden **immer ganze Gebinde. Lose Einzelstücke (z. B. 1 Stück bei VE 6) sind nicht bestellbar** — deshalb muss die Stückmenge ein Vielfaches der VE sein, sonst INVALID_QUANTITY. Beispiel: VE 6 → erlaubt sind 6, 12, 18 ..., nicht 1, 3 oder 4.

Empfehlung: Holen Sie anzahlartve einmalig über GET /api/DSArtikelDaten/{nummer} ab (oder als Liste über .../light) und entscheiden Sie pro Artikel bewusst für eine Variante. Mischen ist möglich — die Angabe wirkt je Position.

12.6 SAP-ORDERS05 → HEGA-JSON: Feld-Mapping

Bewährtes Mapping eines ORDERS05-IDocs auf den HEGA-Bestell-Request (POST /api/DSOrder/OrderAdd/):

HEGA-JSON	SAP ORDERS05 (IDoc)	Hinweis
bestellnummer	E1EDK01/BELNR	Kunden-Bestellnummer, als String senden.
dsorderzeileto[.].artnr	E1EDP01/E1EDP19/IDTNR (QUALF=003)	HEGA-Artikelfnummer.
dsorderzeileto[.].menge	E1EDP01/MENGE	Ganzzahl ; Standard = Menge in VE (s. 12.5).
Empfängeradresse (vorname, nachname, straße, hausnummer, postleitzahl, ort, laenderkennzeichen)	Partnersegment E1EDKA1 (QUALF=WE, Warenempfänger)	Alternativ "useCustomerAddress": true → Lieferung an die zur Kundennummer hinterlegte Adresse; die Adressfelder entfallen dann.

Häufige 400er beim Mapping: bestellnummer als Zahl statt String · menge mit Nachkommastellen (z. B. "1.000") statt Ganzzahl · ein Array zu viel um die Positionen (genau ein dsorderzeileto[.]).

13. Anhang — Gestaltungsmöglichkeiten für Ihren Prozess

*Dieser Anhang richtet sich an alle, die den Geschäftsprozess gestalten. Er schreibt nichts vor — er zeigt, welche **Spielräume** die Schnittstelle lässt, damit Sie den Ablauf nach Ihren Anforderungen entwerfen können.*

13.1 In Stufen einsteigen — Sie bestimmen das Tempo

Sie müssen nicht alles auf einmal anbinden. Ein **möglicher** Ausbaupfad — jede Stufe bringt für sich Nutzen:

1. **Bestellungen übertragen** (FTP oder API) — der Pflichtteil.
2. **Status & Sendungsverfolgung** abrufen und Ihre Endkunden informieren.
3. **Belege** (Lieferschein/Rechnung) automatisiert verarbeiten.
4. **Bestände & Artikeldaten** regelmäßig synchronisieren.

13.2 Wo Sie Spielraum haben

Keine dieser Zeilen ist eine Vorgabe — sie zeigen, **zwischen** welchen Möglichkeiten Sie wählen können:

Gestaltungsfeld	Mögliche Ausprägungen
Integrationsweg	FTP-Stapel · API-Echtzeit · Hybrid (Bestellung per FTP, Rückkanal per API)
Automatisierungsgrad	manuell ausgelöst ... teilautomatisch ... vollautomatisch
Endkunden-Info zum Versand	Sie informieren aus Status <i>Versendet</i> · oder die Carrier-Benachrichtigung (DPD/DHL) · oder beides
Beleg-Handling	PDF anhängen/archivieren · oder CSV/Daten automatisch verbuchen
Teillieferung / Rückstand	Rückstandsfunktion nutzen (Nachlieferung) · oder ohne (Restmenge selbst behandeln)

Gestaltungsfeld	Mögliche Ausprägungen
Bestandssteuerung	täglicher Vollabgleich · oder Delta („seit Datum“) · Auslaufartikel automatisch deaktivieren
Abruf-Rhythmus	so häufig, wie Ihr Prozess es braucht (für Tracking genügt meist 1×/Tag)

13.3 Rahmen, den Sie einplanen sollten

Damit Ihr Prozessdesign nicht ins Leere läuft — das sind heute die Grenzen der Schnittstelle:

- **Storno** läuft per E-Mail/Portal, nicht über die Schnittstelle.
- **Retouren** werden (noch) nicht über die Schnittstelle angemeldet.
- **Automatische Benachrichtigungen (Webhooks)** sind in Planung — heute gilt das Abruf-Prinzip.
- **Bestand** ist ein Status (verfügbar / begrenzt / nicht), keine exakte Stückzahl.

Das sind keine Hindernisse, sondern Rahmen: Planen Sie z. B. für Storno und Retoure einen schlanken manuellen Pfad ein.

Bei Fragen zur Anbindung: EDV@hega.net. Sicherheit: SFTP für FTP, HTTPS/Bearer-Token für die API; Tokens sicher verwahren, 401 per Token-Refresh behandeln.