

# HEPLA GmbH

## Händler-Handbuch

# Teil D: API & Konfigurator-iframe

Artikeldaten · Lagerbestand · Vorschau-Rendering · iframe-Konfigurator

Stand: 1. Juni 2026

Diese Anleitung beschreibt, wie Sie als Händler die HEPLA-API und den iframe-Konfigurator in Ihre eigene Website oder Ihren Online-Shop einbinden. Sie ist Teil eines Pakets aus dieser PDF und zwei Demo-Dateien (`logify_remote.py` und `iframe_demo.html`), die als funktionsfähige Vorlagen dienen.

## 1 Überblick

Die HEPLA-Schnittstelle besteht aus zwei voneinander unabhängigen Teilen, die Sie einzeln oder kombiniert einsetzen können.

Komponente	Wofür?	Zugriff über
Daten-API	Artikeldaten, Preise, Lagerbestand, Bild- und Clip-URLs.	HTTPS-Aufrufe mit API-Token.
Render-API	3D-Vorschau-Clips (MP4) mit individuell platzierten Aufdrucken.	HTTPS-Aufruf mit Bild-Upload (multipart/form-data).
Konfigurator-iframe	Interaktive Konfigurations-Oberfläche zum Einbinden auf der eigenen Produktseite. Endkunde gestaltet den Aufdruck selbst.	iframe + postMessage-Nachrichten.

### 1.1 Mitgelieferte Demo-Dateien

Datei	Zweck
<code>logify_remote.py</code>	Lauffähiges Python-Skript zum Aufruf der Render-API. Erzeugt aus Artikel + Bilddatei einen 3D-Vorschau-Clip. Eigenständige Datei, lässt sich direkt nutzen oder als Vorlage für eine Integration verwenden.
<code>corners.py</code>	Hilfsskript zur Bestimmung der Eckpunkte (corners) einer Druckfläche aus einer Druckmasken-PNG. Liefert die 8 Promille-Werte, die <code>logify_remote.py</code> bzw. die Render-API als Eckpunkte der Logo-Platzierung erwartet.
<code>iframe_demo.html</code>	Eigenständige HTML-Datei mit einem voll

funktionsfähigen iframe-Einbau, inklusive postMessage-Empfänger. Öffnen Sie die Datei im Browser, tragen Sie Ihren Token ein, fertig.

*i* Alle drei Demo-Dateien sind eigenständig und ohne Installation lauffähig. `logify_remote.py` benötigt nur Python 3.7+ und das Paket "requests". `corners.py` benötigt Python 3.7+, Pillow und NumPy. `iframe_demo.html` benötigt nichts außer einen Browser.

## 1.2 Voraussetzungen

- API-Token: erhalten Sie auf Anfrage von HEPLA. Der Token ist persönlich und wird wie ein Passwort behandelt.
- HTTPS-Verbindung zu `www.hepla.de`.
- Für die Render-API: Bilddateien (PNG oder JPG) Ihrer Logos.
- Für den iframe: eine Webseite, auf der Sie einen iframe einbetten können, und idealerweise eine kleine JavaScript-Anpassung zum Empfang der postMessage-Antworten.

## 1.3 Basis-URL

Alle Endpunkte sind erreichbar unter:

```
https://www.hepla.de
```

Beispiel für einen vollständigen API-Aufruf:

```
https://www.hepla.de/api/v1/artikeldata/?token=IHR_TOKEN&artnr=7262-0
```

## 2 Authentifizierung

Alle API-Aufrufe werden über einen persönlichen Token authentifiziert. Der Token wird auf zwei Arten akzeptiert; wählen Sie die für Ihre Integration passendere Variante:

Variante	Beispiel
Query-Parameter (einfach, gut zum Testen im Browser)	?token=IHR_TOKEN
HTTP-Header (sauberer, empfohlen für Produktivnutzung)	X-API-Token: IHR_TOKEN

⚠ Token-Hygiene: Behandeln Sie den Token wie ein Passwort. Geben Sie ihn nicht in öffentlichen Repositories oder Frontend-Code preis. Bei Verlust melden Sie sich bei HEPLA – der Token kann ungültig gemacht und ein neuer ausgestellt werden.

## 3 Daten-API

Die Daten-API liefert Artikeldaten, Preise und Lagerbestände als JSON bzw. CSV. Alle Endpunkte sind GET-Aufrufe.

### 3.1 Endpunkt-Übersicht

Endpunkt	Zweck	Format
----------	-------	--------

/api/v1/artikeldaten/	Stammdaten und Preise eines Artikels (Live-Datenbank).	JSON
/api/v1/artikeldaten/vorschau/	Wie /artikel-daten/, aber mit Preisen aus der Test-Datenbank (Jahreswechsel-Vorbereitung).	JSON
/api/v1/lagerbestand/	Aktueller Lagerbestand. Wahlweise als JSON oder CSV.	JSON oder CSV
/api/v1/render/	3D-Vorschau-Clip mit Aufdruck generieren (POST).	multipart/form-data → JSON-Antwort

## 3.2 Artikeldaten

Liefert Stammdaten, Bezeichnung, Bild- und Clip-URLs und – falls Sie authentifiziert sind und der Artikel für Sie freigegeben ist – auch die Preisstaffeln in Ihrer Konditionsgruppe.

GET [https://www.hepla.de/api/v1/artikeldaten/?token=IHR\\_TOKEN&artnr=7262-0](https://www.hepla.de/api/v1/artikeldaten/?token=IHR_TOKEN&artnr=7262-0)

Parameter:

Parameter	Pflicht	Beschreibung
token	ja	API-Token (alternativ als Header X-Api-Token).
artnr	ja	Artikelnummer, z.B. 7262-0.
sprache	nein	Sprache der Texte: de (Standard), en, fr, es, nl, it.

Antwort (Auszug):

```
{
  "ok": true,
  "artnr": "7262-0",
  "bezeichnung": "Kugelschreiber Lyno",
  "beschreibung": "...",
  "bild_url": "https://www.hepla.de/media/Artikelbilder/7262-0.jpg",
  "clip_url": "https://www.hepla.de/media/Artikelclips/7262-0.mp4",
  "preise": [
    {"ab_menge": 100, "preis": 0.89},
    {"ab_menge": 250, "preis": 0.79},
    {"ab_menge": 500, "preis": 0.71}
  ],
  "druckflaechen": [
    {"code": "T1", "breite_mm": 35, "hoehe_mm": 7, "max_farben": 1},
    {"code": "D1", "breite_mm": 35, "hoehe_mm": 7, "max_farben": null}
  ]
}
```

**i** Bild- und Clip-URLs folgen dem Schema /media/Artikelbilder/<artnr>.jpg und /media/Artikelclips/<artnr>.mp4. Sie können diese URLs direkt auf Ihrer Webseite einbinden – ein zusätzlicher Download über die API ist nicht nötig.

### 3.3 Lagerbestand

Aktueller Lagerbestand zu allen für Sie sichtbaren Artikeln. Auf Wunsch auch als CSV-Download:

```
GET https://www.hepla.de/api/v1/lagerbestand/?token=IHR_TOKEN
```

```
GET https://www.hepla.de/api/v1/lagerbestand/?token=IHR_TOKEN&format=csv
```

Optionale Filter:

Parameter	Beschreibung
artnr	Nur einen bestimmten Artikel (sonst alle).
format	csv = CSV-Download mit Spalten-Header. Sonst JSON.

### 3.4 Jahreswechsel: Vorschau auf neue Preise

Zum Jahresende stellt HEPLA in einer Test-Datenbank die Preise fürs kommende Jahr bereit, bevor sie offiziell in Kraft treten. Sie können diese Preise vorab abrufen, um Ihre Kalkulation, Preislisten oder Datenfeeds vorzubereiten:

```
GET https://www.hepla.de/api/v1/artikel/daten/vorschau/?token=IHR_TOKEN&artnr=7262-0
```

Antwortformat identisch zum Live-Endpunkt; die Preise sind jedoch die neuen Jahrespreise. Zwischen den Jahren weichen die Werte häufig nicht oder nur geringfügig ab; geprüfte Veränderungen erfahren Sie über die reguläre Preisliste.

## 4 Render-API: 3D-Vorschau-Clips

Die Render-API erzeugt 3D-Vorschau-Clips (MP4), die Ihren Artikel mit einem konkret platzierten Logo zeigen – ideal für Produktdetailseiten, Kundenangebote oder Mailings.

### 4.1 Demo-Skript logify\_remote.py

Die einfachste Art, die Render-API zu nutzen, ist das mitgelieferte Demo-Skript logify\_remote.py. Es nimmt Ihren Token, eine Artikelnummer, eine Bilddatei und schreibt das gerenderte MP4 in das Zielverzeichnis Ihrer Wahl.

Voraussetzung: Python 3.7 oder neuer und das Paket requests:

```
pip install requests
```

Einfachster Aufruf:

```
python logify_remote.py 7262-0 /tmp/out D1=logo.png \  
--token=IHR_TOKEN
```

Ergebnis: in /tmp/out/ liegt anschließend 7262-0.mp4.

Mit Eckpunkten (Logo nicht über die ganze Druckfläche, sondern an einer konkreten Position):

```
python logify_remote.py 7262-0 /tmp/out \  
D1[120,85,890,85,890,930,120,930]=logo.png \  
--token=IHR_TOKEN
```

Mit Farbreduktion (für Sieb-/Tampondruck):

```
python logify_remote.py 7262-0 /tmp/out \  
  S1_3[120,85,890,85,890,930,120,930]=logo.png \  
  --token=IHR_TOKEN
```

Mehrere Druckflächen in einem Aufruf:

```
python logify_remote.py 7254-3 /tmp/out \  
  D1[656,52,656,936,350,936,350,52]=logo.png \  
  D2[67,572,950,572,950,866,67,866]=logo.png \  
  --token=IHR_TOKEN
```

Die Datei logify\_remote.py enthält oben in einem ausführlichen Doc-String alle weiteren Optionen und Beispiele. Sie können das Skript auch als Basis für eine eigene Integration in Ihrer Sprache nutzen – die HTTP-Aufrufe sind dort kompakt und nachvollziehbar implementiert.

## 4.2 Eckpunkte ([corners]) ermitteln

Die Eckpunkte beschreiben, wo genau auf der Druckfläche das Logo sitzen soll. Sie werden als acht Promille-Werte (0 bis 1000) angegeben, relativ zur Druckmasken-Skizze des Artikels. Die Reihenfolge ist:

```
[ul_x, ul_y, ur_x, ur_y, lr_x, lr_y, ll_x, ll_y]  
ul = oben links  
ur = oben rechts  
lr = unten rechts  
ll = unten links
```

Wenn Sie keine Eckpunkte angeben, füllt das Logo die gesamte Druckfläche aus – das ist der einfachste und häufigste Fall. Eigene Eckpunkte brauchen Sie nur, wenn das Logo bewusst kleiner sein soll als die volle Druckfläche oder eine spezielle Position braucht.

### 4.2.1 Hilfsskript corners.py

Das mitgelieferte Skript corners.py liest eine HEPLA-Druckmasken-PNG und gibt die acht Eckpunkte der bedruckbaren Fläche im richtigen Format aus. Die Druckmasken-PNGs zu Ihren Artikeln stehen unter <https://www.hepla.de/media/Artikelclips/<artnr>#<druckcode>.png> zum Download bereit (z.B. 7262-0#D1.png für Artikel 7262-0, Druckcode D1).

△ URL-Hinweis: das "#" im Dateinamen ist im URL-Pfad ein reserviertes Zeichen (Fragment-Trenner) und muss als %23 maskiert werden, sonst schneidet curl, wget oder der Browser die URL am # ab. Beispiel-Download mit curl:

```
curl -O "https://www.hepla.de/media/Artikelclips/7262-0%23D1.png"
```

Die heruntergeladene Datei heißt dann lokal 7262-0#D1.png – das # im Dateinamen auf dem Dateisystem ist unproblematisch.

Voraussetzungen: Python 3.7 oder neuer und die Pakete Pillow und numpy:

```
pip install Pillow numpy
```

Einfachster Aufruf:

```
$ python corners.py 7262-0#D1.png  
120,85,890,85,890,930,120,930
```

Verschiedene Ausgabeformate (per --format):

Format	Ausgabe	Verwendung
csv (Standard)	120,85,890,85,890,930,120,930	CSV-Listen, eigene Skripte.
bracket	[120,85,890,85,890,930,120,930]	Direkt einsetzbar in logify_remote.py-Aufrufen.
json	[120, 85, 890, 85, 890, 930, 120, 930]	JSON-Felder in eigenen REST-Aufrufen.
api	"ecken": [120, 85, 890, 85, 890, 930, 120, 930]	Snippet zum direkten Einsetzen in den params-JSON-Aufruf an die Render-API.

Beispiel für die direkte Verkettung mit logify\_remote.py – die Eckpunkte werden zur Laufzeit ermittelt und in den Aufruf eingesetzt:

```
python logify_remote.py 7262-0 /tmp/out \  
D1$(python corners.py 7262-0#D1.png --format=bracket)=logo.png \  
--token=IHR_TOKEN
```

Mit Pixelgrößen-Diagnose (Bildgröße + Bounding-Box in Pixeln gehen auf stderr, das Hauptergebnis bleibt sauber auf stdout):

```
$ python corners.py 7262-0#D1.png --mit-info  
Bildgröße: 1000 × 1000 px  
Bounding: x = 120...890, y = 85...930  
120,85,890,85,890,930,120,930
```

#### 4.2.2 Eigene Position innerhalb der Druckfläche

Sie können von der durch corners.py ermittelten Bounding-Box ausgehen und das Logo innerhalb davon verkleinern oder verschieben. Beispiel: Logo zentriert auf einer Druckfläche von 35 mm × 7 mm, mit Rand von 10 % links/rechts und 5 % oben/unten:

```
D1[100,50,900,50,900,950,100,950]=logo.png
```

### 4.3 Direkter HTTP-Aufruf (ohne Demo-Skript)

Wenn Sie die Render-API direkt aus einer eigenen Anwendung aufrufen möchten, statt logify\_remote.py zu verwenden:

```
POST https://www.hepla.de/api/v1/render/?token=IHR_TOKEN  
Content-Type: multipart/form-data
```

Felder:

```
params          JSON mit den Render-Parametern  
aufdruck_<CODE> Bilddatei je Druckfläche (PNG oder JPG)
```

JSON-Format von "params":

```
{
  "artikelnummer": "7262-0",
  "flaechen": [
    {
      "code": "D1",
      "datei": "aufdruck_D1",
      "ecken": [120,85,890,85,890,930,120,930],
      "farbreduktion": 3
    }
  ],
  "masken": false
}
```

Antwort (JSON, Base64-kodierte Dateien):

```
{
  "ok": true,
  "mp4": "<Base64-kodiertes MP4>",
  "masken": { "D1": "<Base64-PNG>" }, // nur wenn masken=true
  "log": "...",
  "warnungen": []
}
```

***i** Die typische Rechenzeit liegt bei 5-10 Sekunden pro Clip. Setzen Sie in Ihrer Implementation ein großzügiges Timeout (z.B. 120 Sekunden), um Lastspitzen abzufedern.*

## 5 iframe-Konfigurator

Der Konfigurator-iframe ist eine fertige Web-Oberfläche, die der Endkunde direkt auf Ihrer Webseite nutzt: Logo hochladen, auf der Druckfläche positionieren, Farben anpassen, Vorschau prüfen. Am Ende übergibt der iframe einen Bestellcode an Ihre Seite, mit dem Sie die Konfiguration später wieder abrufen können.

### 5.1 Demo-Datei `iframe_demo.html`

Die mitgelieferte `iframe_demo.html` ist eine eigenständige HTML-Datei mit allem, was Sie zum Start brauchen:

- iframe-Einbettung mit den richtigen URL-Parametern.
- JavaScript-Empfänger für die `postMessage`-Antworten.
- Beispielanzeige des Bestellcodes und der Vorschau-URL nach Übernahme.

Schnellstart:

- `iframe_demo.html` in einem Editor öffnen.
- Ihren API-Token (`HAENDLER_TOKEN`) am Anfang des Skripts eintragen.
- Datei im Browser öffnen.
- Konfigurieren, „Konfiguration übernehmen“ - der Bestellcode wird unter dem iframe angezeigt.

Die Datei dient als Vorlage für Ihre eigene Integration; sie ist bewusst klein gehalten und kommentiert.

## 5.2 iframe-URL

Der iframe wird mit folgender URL eingebunden:

```
https://www.hepla.de/konfigurator/?token=IHR_TOKEN&artikel=7262-0
```

URL-Parameter:

Parameter	Pflicht	Beschreibung
token	ja	Ihr API-Token.
artikel	ja	Artikelnummer, z.B. 7262-0.
sprache	nein	Anzeigesprache: de (Standard), en, fr, es, nl, it.
ref	nein	Frei wählbare Händler-Referenz (z.B. Ihre interne Bestellnummer). Wird in der späteren postMessage-Antwort unverändert zurückgereicht.
code	nein	Vorhandener Bestellcode zum Wiedereinlesen einer früheren Konfiguration. Der iframe lädt die gespeicherte Konfiguration beim Start und stellt sie zum Bearbeiten bereit.
artikelname	nein	Ihr Anzeigenname für den Artikel, wenn Sie nicht den Standard-Namen verwenden möchten.
logos	nein	Komma-getrennte Liste von URLs zu Logos Ihrer Kunden, die beim Start in der Galerie vorgeladen werden sollen.
theme	nein	Farbschema. Standard verwendet die HEPLA-Optik; sprechen Sie uns für eine händlerspezifische Variante an.

## 5.3 iframe einbinden

Minimaler HTML-Code zum Einbetten:

```
<iframe  
  src="https://www.hepla.de/konfigurator/?token=IHR_TOKEN&artikel=7262-0"  
  width="100%" height="800"  
  style="border:0">  
</iframe>
```

**i** *Empfohlene Mindestgröße: 800 px Höhe, mindestens 600 px Breite. Bei kleinerem Viewport wird die Druckmasken-Vorschau und das Werkzeug-Panel zu eng dargestellt. Auf Mobilgeräten passt sich der Konfigurator automatisch responsiv an.*

### 5.3.1 Download der 3D-Vorschau ermöglichen

Standardmäßig kann der Endkunde die berechnete 3D-Vorschau innerhalb des iframes ansehen, aber nicht als Videodatei herunterladen. Falls Sie Ihren Kunden das Herunterladen erlauben möchten, müssen Sie ZWEI Dinge tun:

- Im iframe-Tag das Attribut `sandbox` setzen und dort `allow-downloads` aufnehmen (zusätzlich zu den anderen benötigten Flags). Ohne diesen Eintrag blockiert der Browser den Download.
- In der iframe-URL den Parameter `&downloads=1` ergänzen. Erst dann zeigt der Konfigurator den Download-Knopf an. Ohne diesen Parameter erscheint kein Knopf, damit der Endkunde keine Funktion sieht, die in seinem Kontext nicht aktiv ist.

Komplettes Beispiel mit aktiviertem Download:

```
<iframe
  src="https://www.hepla.de/konfigurator/?token=IHR_TOKEN&artikel=7262-
0&downloads=1"
  width="100%" height="800"
  style="border:0"
  sandbox="allow-scripts allow-same-origin allow-forms allow-popups allow-
downloads">
</iframe>
```

***i** Der Download wird vom HEPLA-Server mit dem HTTP-Header "Content-Disposition: attachment" ausgeliefert. Der Browser speichert die Datei direkt als 3D-Vorschau\_<Artikelnummer>.mp4, ohne dass der Endkunde sie erst im Video-Player anschauen muss.*

## 5.4 Kommunikation per postMessage

Der iframe sendet zwei Arten von Nachrichten an die einbettende Seite. Sie können in Ihrem JavaScript darauf reagieren, um z.B. den Bestellcode in Ihren Warenkorb zu übernehmen.

### 5.4.1 konfigurator\_ready

Wird einmal beim vollständigen Laden des Konfigurators gesendet:

```
{
  type:          'konfigurator_ready',
  artikelnummer: '7262-0'
}
```

Sie können diese Nachricht z.B. nutzen, um einen Ladeindikator auszublenden.

### 5.4.2 konfigurator\_result

Wird gesendet, wenn der Endkunde im iframe auf „Konfiguration übernehmen“ klickt:

```
{
  type:          'konfigurator_result',
  code:          '4A7B2X',           // Bestellcode
  gueltig_bis:   '2026-06-13T14:32:00+02:00', // ISO 8601
  artikelnummer: '7262-0',
}
```

```

haendler_ref:      '...', // wie URL-Param 'ref'
druckflaechen:    [ /* siehe unten */ ],
vorschau_clip_url: 'https://www.hepla.de/.../...mp4' // optional
}

```

Das druckflaechen-Array enthält pro konfigurierter Fläche einen Eintrag mit:

Feld	Beschreibung
code	Druckcode (z.B. "D1").
farben	Anzahl genutzter Farben (bei Sieb-/Tampondruck).
nachreichen	true, wenn der Kunde angegeben hat, das Bildmotiv per E-Mail nachzureichen statt es jetzt hochzuladen.
ecken	Eckpunkte [ul_x, ul_y, ur_x, ur_y, lr_x, lr_y, ll_x, ll_y] in Promille (0-1000).
logo_name	Dateiname des gewählten Logos.

Beispiel-Empfänger in JavaScript:

```

window.addEventListener('message', (event) => {
  if (event.origin !== 'https://www.hepla.de') return;
  const data = event.data;
  if (data.type === 'konfigurator_ready') {
    console.log('Konfigurator bereit.');

```

⚠ Sicherheitsempfehlung: Prüfen Sie in Ihrem postMessage-Empfänger immer event.origin === "https://www.hepla.de", bevor Sie die Daten verwenden. So stellen Sie sicher, dass die Nachricht tatsächlich vom Konfigurator stammt und nicht von einer fremden Seite.

## 5.5 Bestellcode und Gültigkeit

Der Bestellcode ist eine 6-stellige alphanumerische Kennung, die HEPLA die vom Endkunden gestaltete Konfiguration eindeutig zuordnet. Geben Sie diesen Code zusammen mit Ihrer Bestellung an den HEPLA-Innendienst weiter; wir können dann auf die Konfiguration zugreifen, ohne dass Sie die Druckdateien selbst weiterleiten müssen.

- Format: 6 Großbuchstaben und Ziffern (Beispiel: 4A7B2X).
- Gültigkeit: 30 Tage ab Übernahme der Konfiguration. Das genaue Ablaufdatum steht im Feld gueltig\_bis.
- Wiederbearbeitung: solange der Code gültig ist, können Sie ihn als URL-Parameter ?code=... erneut an den iframe übergeben. Der Endkunde sieht dann seine vorherige Konfiguration und kann sie anpassen.

## 6 Bestellablauf

Empfohlener Ablauf einer Bestellung über die Schnittstelle:

Schritt	Bei Ihnen	Auf HEPLA-Seite
1	Sie binden den iframe-Konfigurator in Ihre Produktseite ein.	-
2	Endkunde gestaltet im iframe Logo, Platzierung, Farben, sieht 3D-Vorschau.	Speichert die Konfiguration intern.
3	Endkunde klickt „Konfiguration übernehmen“.	Erzeugt 6-stelligen Bestellcode, sendet postMessage zurück.
4	Sie speichern den Code in Ihrem Bestellsystem (zusammen mit Stückzahl, Lieferadresse und Ihrer eigenen Auftragsnummer).	-
5	Sie senden Ihre Bestellung an HEPLA - per E-Mail an den Innendienst, mit Bestellcode, Stückzahl und gegebenenfalls haendler_ref als Referenz.	-
6	-	Innendienst liest die Konfiguration anhand des Codes ein, produziert den Auftrag, sendet Auftragsbestätigung.

## 7 Fehler und Antwortformate

Im Erfolgsfall enthält die Antwort immer das Feld "ok": true. Bei Fehlern:

```
{ "ok": false, "fehler": "Beschreibungstext" }
```

Typische Fehler:

HTTP	Fehler	Bedeutung
401	Kein API-Token angegeben.	token-Parameter und X-API-Token-Header beide leer.
401	Ungültiger oder gesperrter API-Token.	Token nicht (mehr) gültig. Bei HEPLA melden.
400	Artikel nicht gefunden.	Artikelnummer prüfen.
400	Artikel nicht für API freigegeben.	Nicht jeder Artikel ist für den iframe-Konfigurator gedacht. Bei HEPLA nachfragen.
500	Render-Fehler.	Selten, meist durch ungültige Eckpunkte oder beschädigte Bilddatei. Demo-Skript liefert genauere Diagnose im Server-Log.

## 8 Support und Kontakt

Bei Fragen zur Integration, neuen Token-Anfragen, oder bei Problemen:

- HEPLA GmbH, [www.hepla.de](http://www.hepla.de)
- E-Mail: [it@hepla.de](mailto:it@hepla.de) (Token-Verwaltung und API-Fragen).
- E-Mail: [vertrieb@hepla.de](mailto:vertrieb@hepla.de) (Bestellungen, Artikel-Freigaben).

Bitte geben Sie bei API-Fragen wenn möglich an: Ihre Händler-Kennung, die Artikelnummer und gegebenenfalls den genauen Aufruf oder die Fehlermeldung – damit können wir schneller helfen.