Bestand und Bestellmenge

Standardmäßig ermittelt FINN.ghost den Bestand eines Artikels aus der SelectLine-Ansicht `SL_vGetFreierArtikelBestand`: **Bestand minus Reserviert**, bei Stücklisten die kleinste Stückzahl, die sich aus den Bestandteilen bauen lässt. Für alle Fälle, in denen das nicht passt, gibt es `customStock` — eine eigene SQL-Abfrage, die den Bestand liefert.

> Typische Gründe: Es sollen nur bestimmte Läger in den Shop gemeldet werden, ein Sicherheitsbestand soll abgezogen werden, oder ein Sperrlager soll unberücksichtigt bleiben.

customStock

Alle drei Shop-Anbindungen kennen die Einstellung. Der Schlüssel ist derselbe, Abschnitt und Platzhalter unterscheiden sich:

Anbindung	Schlüssel	Platzhalter
Shopware 6	`sw6.customStock`	`${article.Number}`
Shopify	`shopify.customStock`	`${nummer}`
FINN.webshop	`shop.customStock`	`${artikelnummer}`

> Der Platzhalter ist je Anbindung ein anderer. Wird der falsche verwendet, wird er nicht ersetzt — die Abfrage läuft dann mit dem Text `${nummer}` als Artikelnummer, findet nichts und liefert Bestand **0** für jeden Artikel.

Was die Abfrage liefern muss

Eine Spalte mit dem Namen **`Bestand`**. Ausgewertet wird die **erste Zeile** des Ergebnisses.

Das Beispiel meldet nur die Läger `01` und `02` und zieht die reservierte Menge ab:

```
{
  "sw6": {
    "customStock": "SELECT SUM(ISNULL(Bestand,0)-ISNULL(Reserviert,0)) AS Bestand FROM
dbo.SL_vGetFreierArtikelBestandMitLager WHERE Artikelnummer = '${article.Number}' AND Lager IN
```

```
('01','02')"
  }
}
```

> Die Ansicht `SL_vGetFreierArtikelBestandMitLager` ist für solche Abfragen der bequemste Einstieg: dieselben Werte wie in der Standardberechnung, nur zusätzlich je Lager. Für Standorte statt Läger gibt es `SL_vGetFreierArtikelBestandMitStandort`.

Spalte heißt nicht `Bestand`	Ergebnis wird als 0 gewertet
Abfrage liefert keine Zeile	Bestand 0
Ergebnis ist negativ	wird auf 0 angehoben
Mehrere Zeilen	nur die erste zählt

> Der Wert wird als ganze Zahl übertragen. Nachkommastellen fallen weg — bei Artikeln, die in Metern oder Kilogramm geführt werden, ist das zu bedenken.

Was die Abfrage ersetzt

Ist `customStock` gesetzt, entscheidet **allein** die Abfrage. Die Standardlogik läuft nicht mehr — und damit auch nichts, was daran hängt:

Läuft nicht mehr	Folge
Abzug von **Reserviert**	muss in der Abfrage selbst stehen
Auflösung von **Stücklisten**	Set- und Behälterartikel liefern den Wert der Abfrage, nicht die baubare Menge

> Die Stücklistenauflösung ist der Punkt, der am häufigsten übersehen wird. Ohne `customStock` meldet FINN.ghost für einen Set-Artikel die Menge, die sich aus den Bestandteilen bauen lässt. Mit `customStock` liefert die Abfrage den Wert — und der eigene Lagerbestand eines Set-Artikels ist in der Regel 0.

Welche Artikel überhaupt übertragen werden, ändert `customStock` **nicht**: Die Auswahl nach *Shopaktiv* und *Inaktiv* passiert davor, beim Zusammenstellen der Artikelliste. Ein nicht shopaktiver Artikel wird also weiterhin gar nicht erst gemeldet.

> Bei **Shopify** ist das anders: Dort gehört die Prüfung auf *Shopaktiv* und *Inaktiv* zur Bestandsermittlung selbst und wird von `shopify.customStock` mit übergangen — ebenso

`shopify.zeroStockField`. Beides gehört dann in die Abfrage.

Rückfall auf die Standardberechnung (nur Shopware 6)

Bei Shopware 6 gibt es einen Ausweg für einzelne Artikel: Liefert die Abfrage den Wert `-9999`, verwirft FINN.ghost das Ergebnis und rechnet für diesen Artikel wie ohne `customStock`.

```
SELECT CASE WHEN ... THEN 0 ELSE -9999 END AS Bestand FROM ART WHERE Artikelnummer =
'${article.Number}'
```

Damit lässt sich die Sonderlogik auf die Artikel beschränken, die sie brauchen. Alle anderen laufen weiter über die geprüfte Standardberechnung — inklusive Stücklistenauflösung.

Shopify und FINN.webshop kennen diesen Rückfall nicht. Dort gilt die Abfrage für jeden Artikel.

Besonderheit bei Shopify

Shopify führt Bestände je Standort. `shopify.customStock` liefert **einen** Wert, und dieser wird auf **alle** eingerichteten Standorte geschrieben — nicht aufgeteilt.

Bei mehreren Standorten summiert Shopify die Bestände. Ein Wert von 10 bei drei Standorten wird im Shop zu 30 verfügbaren Stück. Wer mit mehreren Standorten arbeitet, sollte `customStock` dort nicht einsetzen.

Zusätzlich greift die Abfrage nur, wenn die Bestandsführung für den Artikel aktiv ist. Ist sie abgeschaltet, überträgt Shopify keinen Bestand — dann läuft die Abfrage nicht.

Weitere Einstellungen zum Bestand

Shopware 6

Schlüssel	Typ	Wirkung
`sw6.ignoreReserved`	Schalter	rechnet mit **Bestand** statt Bestand minus Reserviert. Reservierungen aus offenen Aufträgen mindern den Shopbestand dann nicht.
`sw6.customPurchase`	Abfrage	eigene Abfrage für die **maximale Bestellmenge**. Platzhalter `${article}`, Spalte `Bestand`.
`sw6.maxPurchaseField`	Feldname	Artikelfeld mit der maximalen Bestellmenge je Artikel.

> Ist `sw6.maxPurchaseField` gesetzt und im Artikel gefüllt, gilt dieser Wert — `sw6.customPurchase` kommt dann für diesen Artikel nicht mehr zum Zug. Das Feld hat Vorrang.

> `sw6.ignoreReserved` wirkt nur, wenn `sw6.customStock` **nicht** gesetzt ist oder für den Artikel `-9999` liefert. Sonst rechnet ohnehin nur die eigene Abfrage.

Shopify

Schlüssel	Typ	Wirkung
`shopify.zeroStockField`	Feldname	Artikelfeld; ist es `True`, wird Bestand **0** gemeldet, obwohl Bestand vorhanden ist. Wirkt auch über den Variantenartikel und über Stücklisten: hat ein Bestandteil das Feld gesetzt, ist die ganze Stückliste 0.
`shopify.inventoryTrackDisableField`	Feldname	Artikelfeld; ist es `True`, wird die **Bestandsführung** für diesen Artikel in Shopify abgeschaltet. Der Artikel ist dann unbegrenzt bestellbar.

> `shopify.zeroStockField` ist der saubere Weg, einzelne Artikel im Shop auf nicht verfügbar zu setzen, ohne sie in der SelectLine anzufassen — und deutlich einfacher als eine eigene Abfrage.

Wenn der Bestand im Shop nicht stimmt

Beobachtung	Erste Prüfung
alle Artikel stehen auf 0	Platzhalter falsch geschrieben, oder Spalte heißt nicht `Bestand`
Stücklisten stehen zu niedrig	`customStock` löst Stücklisten nicht auf
bei Shopify: gesperrte Artikel bestellbar	`shopify.zeroStockField` wird von der Abfrage übergangen
Shopify zeigt ein Vielfaches	mehrere Standorte, der Wert wird je Standort geschrieben
Bestand ändert sich nicht mehr	Lauf steht mit SQL-Fehler im Protokoll

Die Abfrage lässt sich vor dem Eintragen im SQL Management Studio prüfen — dort den Platzhalter durch eine echte Artikelnummer ersetzen. Meldungen der Läufe stehen im Protokoll, siehe Log Informationen.

Nächster Schritt

Shopware 6

Created 2026-09-07 11:28:23 UTC by API
Updated 2026-09-07 11:28:54 UTC by API