



NoyesStorage Betriebsanleitung

Build-Stand 3e11ee9a7

Noyes Robotics GmbH

Copyright © 2025 Noyes Robotics GmbH

Inhaltsverzeichnis

1. Home

1.1 Inhalt und Ziel

1.2 Erforderliche und ergänzende Dokumentation

1.3 Darstellung von Informationen

1.4 Sicherheitshinweise

1.5 Symbole

2. Produkt & Technik

2.1 Warum NoyesStorage?

2.2 Konfigurationen

2.3 Komponenten

2.4 Durchsatz

2.5 Sicherheitshinweise

2.6 Weitere Systemspezifikationen

3. IT & Cybersicherheit

3.1 Übersicht

3.2 Netzwerk & Internetanbindung

3.3 Zugriff & Sicherheitskonzept

3.4 Go-Live Checkliste

4. Bedienung & Integration

4.1 Bedienung über App

4.2 Integration über API

5. Support & Service

5.1 Fehlerbehebung

5.2 Reinigung, Inspektion, Wartung und Reparatur

5.3 Transport und Lagerung

5.4 Demontage und Entsorgung

6. Changelog

6.1 v4.38 (783fdef39) ab 10.03.2026

6.2 v4.37 (d2edd42e7) ab 24.02.2026

6.3 v4.36 (8c7bdc6f7) ab 11.02.2026

6.4 v4.35 (4a61c0c9e) ab 27.01.2026

6.5 v4.34 (8bfbf673c) ab 12.01.2026

6.6 v4.33 (44da13c9d) ab 19.12.2025

6.7 v4.32 (f0f86cc27) ab 01.12.2025

6.8 v4.31 (0d4cd4c3f) ab 17.11.2025

6.9 v4.30 (04b4ec69b) ab 03.11.2025

6.10 v4.29 (5a42740fa) ab 20.10.2025

6.11 v4.28 (00a0b2853) ab 06.10.2025

6.12 v4.27 (da4af2fb5) ab 19.09.2025

6.13 v4.26 (8ed961f77) ab 05.09.2025

6.14 v4.25 (b0e02f1e7) ab 12.08.2025

6.15 v4.24 (7173867ca) ab 25.07.2025

6.16 v4.23 (cfb64c84f) ab 16.07.2025

6.17 v4.22 (94520107f) ab 10.07.2025

6.18 v4.21 (9208edcc1) ab 12.06.2025

6.19 v4.20 (00337a922) ab 15.05.2025

6.20 v4.19 (d338a651c) ab 29.04.2025

6.21 v4.18 (2f2ae4c89) ab 03.04.2025

6.22 v4.16 (eeb87589d) ab 06.03.2025

6.23 v4.15 (e77b9e6e4) ab 20.02.2025

6.24 v4.14 (2d929340f) ab 23.01.2025

6.25 v4.13 (7fb944a28) ab 21.11.2024

7. About

1. Home

1.1 Inhalt und Ziel

Dieses Dokument gilt für das Produkt **NoyesStorage**, einem automatisierten Kleinteilelager. Zweck dieses Dokuments ist es:

- eine detaillierte technische Spezifikation für den NoyesStorage bereitzustellen und
- wichtige Hinweise zur Verfügung zu stellen, wie das System sicher und bestimmungsgemäß zu bedienen ist.

Lesen Sie diese Dokumentation vollständig und insbesondere das Kapitel [Sicherheitshinweise](#) und den Abschnitt Kapitel [Allgemeine Sicherheitshinweise](#), bevor Sie mit dem Produkt arbeiten.

Dieses Dokument ist so strukturiert, dass es die wichtigsten Informationen über den NoyesStorage in übersichtlicher Form bereitstellt:

- **NoyesStorage** enthält detaillierte technische Spezifikationen, einschließlich Umwelt-, Leistungs-, mechanischer, elektrischer, Software- und Sicherheitsaspekte des Systems. Dies ist notwendig, um sicherzustellen, dass das System die Betriebsstandards und gesetzlichen Anforderungen erfüllt.
- **Sicherheitshinweise** erklärt, wie das System sicher und bestimmungsgemäß verwendet wird, und worauf es bei der Bedienung hinsichtlich Sicherheit zu achten gilt.
- **Bedienung des NoyesStorage** erläutert die Verwendung des Systems, z. B. zum Ein- und Auslagern von Ware oder dem Management von Stammdaten. Dieser Abschnitt hilft, zu beurteilen, ob das System den spezifischen betrieblichen Anforderungen entspricht.
- **Integration des NoyesStorage** erklärt, wie NoyesStorage mit bestehenden Systemen wie ERP oder WMS verbunden werden kann.
- **Fehlerbehebung** behandelt die Analyse und Lösung von Fehlern. Zunächst werden die Statusanzeigen von NoyesStorage und NoyesBot erläutert, gefolgt von Anweisungen zur Behebung einfacher Fehler.
- **Reinigung, Inspektion, Wartung und Reparatur** beschreibt die empfohlenen Intervalle für Reinigung, Wartung und Inspektion des NoyesStorage. Diese Maßnahmen tragen dazu bei, die Lebensdauer des Systems zu verlängern und die Ausfallrate gering zu halten.
- **Transport und Lagerung** liefert Informationen zu Transport und Lagerung des NoyesStorage, um Transport- und Lagerschäden zu vermeiden.
- **Demontage und Entsorgung** erklärt, wie der NoyesStorage am Ende seiner Lebensdauer sicher demontiert und fachgerecht entsorgt wird.

1.2 Erforderliche und ergänzende Dokumentation

Nehmen Sie das Produkt erst dann in Betrieb, wenn Sie die Betriebsanleitung erhalten, verstanden und befolgt haben.

Dieses Handbuch soll zusammen mit den folgenden Referenzdokumenten verwendet werden:

- Service Level Agreement (SLA)
- [Allgemeine Geschäftsbedingungen \(AGB\)](#)
- NoyesStorage Spezifikation
- CE-Konformitätserklärung (PDF) ([Deutsch](#), [Englisch](#))

1.3 Darstellung von Informationen

Damit Sie mithilfe dieser Dokumentation schnell und sicher mit Ihrem Produkt arbeiten können, werden einheitliche Sicherheitshinweise, Symbole, Begriffe und Abkürzungen verwendet. Zum besseren Verständnis werden diese in den folgenden Abschnitten erläutert.

1.4 Sicherheitshinweise

In dieser Dokumentation stehen Sicherheitshinweise vor einer Handlungsabfolge, bei der die Gefahr von Personen- oder Sachschäden besteht. Die beschriebenen Maßnahmen zur Gefahrenabwehr sind zu beachten.

Sicherheitshinweise sind wie folgt aufgebaut:

WARNUNG

Art und Quelle der Gefahr

Konsequenzen bei Nichtbeachtung:

- Wie vermeidet man 1
- Wie vermeidet man 2

- **Warnzeichen:** macht auf die Gefahr aufmerksam
- **Signalwort:** gibt die Schwere der Gefahr an
- **Art und Quelle der Gefahr:** benennt die Art und Quelle der Gefahr
- **Konsequenzen:** beschreibt die Folgen des Ignorierens der Gefahr
- **Verhinderung:** gibt an, wie die Gefahr zu vermeiden ist

Die folgenden Sicherheitshinweise finden in dieser Dokumentation Anwendung (Gefahrenklassen nach ANSI Z535.6-2006):

GEFAHR

Diese Kombination aus Symbol und Signalwort weist auf eine gefährliche Situation hin, die, wenn sie nicht vermieden wird, zum Tod oder zu schweren Verletzungen führen *wird*.

WARNUNG

Diese Kombination aus Symbol und Signalwort weist auf eine gefährliche Situation hin, die, wenn sie nicht vermieden wird, zum Tod oder zu schweren Verletzungen führen *kann*.

VORSICHT

Diese Kombination aus Symbol und Signalwort weist auf eine gefährliche Situation hin, die, wenn sie nicht vermieden wird, zu leichten oder mittelschweren Verletzungen führen kann.

HINWEIS

Diese Kombination aus Symbol und Signalwort weist auf eine möglicherweise gefährliche Situation hin, die zu Sach- oder Umweltschäden führen kann, wenn sie nicht vermieden wird.

1.5 Symbole

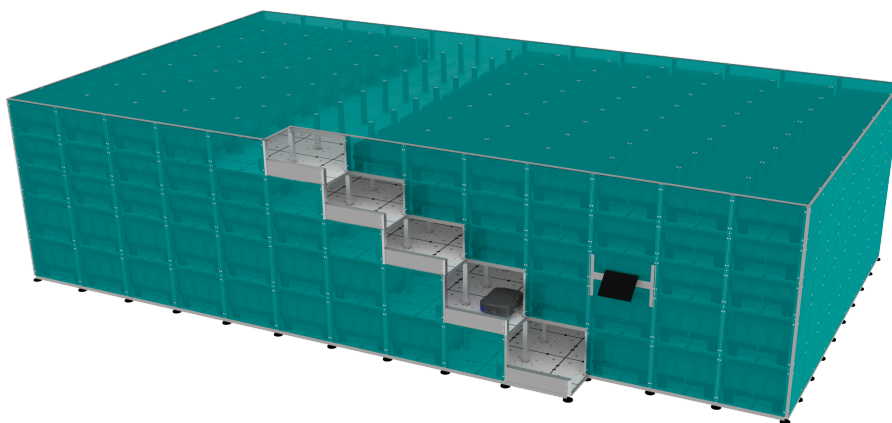
Information

Wenn diese Hinweise nicht beachtet werden, kann das Produkt nicht optimal eingesetzt und betrieben werden.

2. Produkt & Technik

2.1 Warum NoyesStorage?

NoyesStorage ist ein modular aufgebautes, automatisiertes Kleinteilelager (engl. Automated Storage and Retrieval System, ASRS), das eine effizientere Flächennutzung ermöglicht und zugleich eine schnellere Kommissionierung unterstützt. Der folgende Abschnitt liefert eine Übersicht und beschreibt die Anwendungsfälle, in denen NoyesStorage eingesetzt wird, sowie die zentralen Herausforderungen, die dadurch adressiert werden. Danach werden die verschiedenen verfügbaren **Konfigurationen** spezifiziert, die zur Bestellung angeboten werden. Abschließend erfolgt eine detaillierte Beschreibung des **Durchsatzes** sowie **Weiterer Systemspezifikationen**).



NoyesStorage bietet flexible Automatisierung mit einer schnellen Amortisation, besonders in platzbeschränkten Umgebungen. Es verwandelt Lagerprozesse – die traditionell durch wiederholte, fehleranfällige manuelle Aufgaben geprägt sind – in effiziente, digitalisierte Prozesse. Derzeit wird NoyesStorage sowohl für Kommissionier- als auch für Lageraufgaben genutzt.

Für die Lagerung steigern das zum Patent angemeldete Design und die mehrfachtiefe Einlagerung die Lagerdichte im Vergleich zu herkömmlichen Regalsystemen erheblich. Dies führt zu einer effizienteren Raumnutzung und verbessert die Gesamtlagerkapazität um bis zu 71% (beim NoyesStorage M im Vergleich zu einem konventionellen Lager mit Fachbodenregalen).

Bei der Kommissionierung überzeugt NoyesStorage in Anwendungen wie dem E-Commerce oder der Produktionsvorbereitung, bei denen sortenreine Artikel entnommen werden, um Körbe mit mehreren unterschiedlichen Artikeln zusammenzustellen. Diese Körbe werden anschließend für den Versand verpackt oder in der Produktion verwendet. Durch die Automatisierung dieses Prozesses steigert NoyesStorage die Effizienz der Kommissionierung, indem zeitaufwändiges Suchen entfällt und Laufwege für Picker minimiert werden. So sind je nach spezifischem Anwendungsfall bis zu 300 Picks pro Stunde mit einem Picker möglich (**Durchsatz**).

In beiden Szenarien bietet die Digitalisierung des Systems volle Transparenz und minimiert Fehler – ohne zusätzliche Kosten. Fehler werden reduziert, indem stets der richtige Artikel präsentiert wird, wodurch manuelles Suchen entfällt und eine gezielte Führung zur Auswahl des korrekten Artikels erfolgt. Das System verfolgt den Standort jedes Artikels, und Roboter liefern diese zum Kommissionierbalkon. Alle Interaktionen mit dem NoyesStorage werden entsprechend erfasst, wodurch sich ein Maß an Transparenz ergibt, das in traditionellen, manuellen Lagerprozessen nicht wirtschaftlich wäre.

NoyesStorage ist für die Lagerung von Kleinteilen konzipiert und kann entsprechend Artikel mit Abmaßen bis zu 600 x 400 x 320 mm und einem maximalen Gewicht von 40 kg aufnehmen, die typischerweise in Kisten gelagert werden. Im Gegensatz zu anderen ASRS-Systemen ist NoyesStorage nicht auf spezielle Kisten angewiesen, sondern unterstützt eine Vielzahl von Boxentypen, wodurch hohe Kosten für die Neuanschaffung von Boxen vermieden werden. Um die Lagerung

weiter zu optimieren, unterstützt die Software auch kleinere Boxen sowie Boxeneinsätze. Neben Standardboxen werden auch andere Ladungsträger wie Kartonagen und Fässer unterstützt, was den Bedarf an Umpackprozessen im Wareneingang verringert.

NoyesStorage ist standardmäßig als eigenständiges System konzipiert und bietet vollständige Lager- und Abruffunktionen ohne Integrationsbedarf. Diese sofort einsetzbare Lösung wird vollständig über eine App unterstützt, die Zugriff auf verschiedene Prozesse und Funktionen bietet (siehe [Betrieb](#)). Für Anwender, die zusätzliche Konnektivität benötigen, bietet NoyesStorage auch die Möglichkeit zur Integration mit anderen Systemen wie Enterprise Resource Planning (ERP) oder Lagerverwaltungssystemen (WMS). Die intuitive Programmierschnittstelle (API) vereinfacht den Integrationsprozess, und die [Integrationsanleitung](#) enthält eine Schnellstart-Anleitung, die eine prototypische Integration in Stunden statt Tagen ermöglicht.

NoyesStorage wird maßgeblich durch Software definiert, die kontinuierlich weiterentwickelt wird. Das System wird mit der neuesten Softwareversion ausgeliefert, und gemäß dem zusätzlich abzuschließenden SLA werden regelmäßige Over-the-Air-Updates eingespielt. Diese Updates umfassen Leistungsverbesserungen (z. B. zur Steigerung des Durchsatzes), neue Funktionen (z. B. für ein effizienteres Management und Nutzung des Systems) sowie Sicherheitsupdates und Fehlerbehebungen. Dieser Ansatz stellt sicher, dass das System im Laufe seines Lebenszyklus leistungsfähiger und effizienter wird und langfristig Wert und Anpassungsfähigkeit bietet.

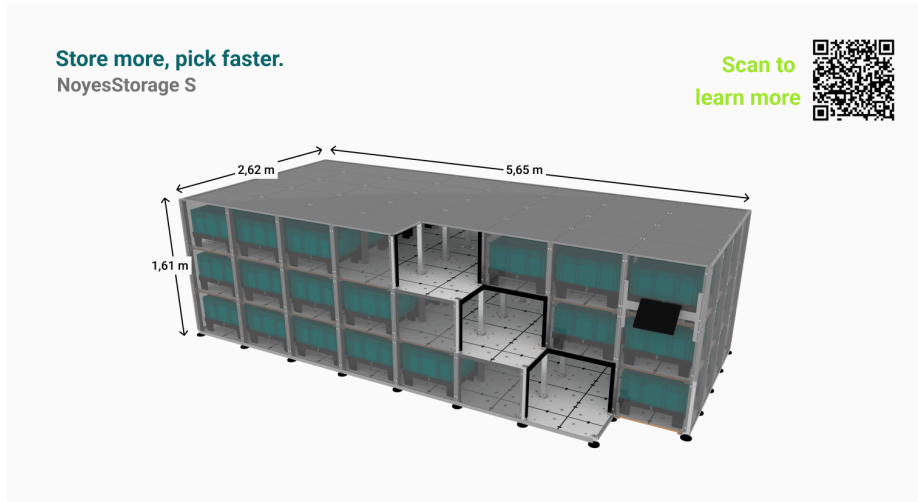
Die Anforderungen und Platzverhältnisse in der Logistik variieren stark zwischen Lagern. Um diesen unterschiedlichen Gegebenheiten gerecht zu werden, orientiert sich NoyesStorage an einem modularen Ansatz. Ähnlich wie sich Küchen durch die Kombination verschiedener Komponenten an spezifische Anforderungen anpassen lassen, funktioniert auch NoyesStorage. Dieser besteht aus modularen Komponenten, die zu einer Systemkonfiguration kombiniert werden können, die den speziellen Bedingungen und Bedürfnissen jedes Anwendungsfalls entspricht. NoyesStorage wird in drei Standardkonfigurationen angeboten, wobei bei Bedarf maßgeschneiderte Lösungen konfiguriert werden können ([Konfigurationen](#)).

Diese Modularität unterstützt auch die zukünftige Skalierbarkeit. Beispielsweise kann das System in Situationen, in denen Durchsatz- und Lageranforderungen anfangs unklar sind, im Laufe der Zeit problemlos erweitert werden, um den steigenden Bedarf zu decken.

2.2 Konfigurationen

Die folgenden Standardkonfigurationen des NoyesStorage sind verfügbar. Weitere Konfigurationen können bei Bedarf angepasst werden.

NoyesStorage S



Specification

System

Width	5,65 m	# Modules	8
Depth	2,62 m	# Modules	5
Height	1,19 - 1,61 m (depending on chosen Level height)	# Modules	3
Footprint	14,8 m ²		
Installation time	3 Days		

Storage Capacity

Large Boxes (600x400)	84 (up to 40 kg each, evenly distributed)
Med. Boxes (300x400)	168 (up to 20 kg each, evenly distributed)
Small Boxes (300x200)	336 (up to 10 kg each, evenly distributed)
Mini Boxes (150x200)	672 (up to 5 kg each, evenly distributed)
Box Height	<ul style="list-style-type: none"> • 320 mm for high levels • 170 mm for low levels
Maximum Payload	up to 40 kg evenly distributed
Maximum Dimensions	<ul style="list-style-type: none"> • on high levels: 600x400x320 • on low levels: 600x400x170
Module Utilization	73%
Constraints	No dangerous goods according to ADR

* Boxes are not included in this offer. However, suitable boxes for your NoyesStorage system can be provided separately by Noyes Robotics GmbH upon request

Operation Conditions

Temperature	3-30 °C
Humidity	20-90% (non-condensing)
Usage	Indoor

NoyesStorage Component List

Modules	114	Carriers	84
Picking balconies	3	NoyesBots	3-6
Layers high	0-3	Layers low	0-3
Charging Stations	3-6	Commander	1
Platform (with steps)	0	Elevator	0

Throughput Performance

	2 NoyesBots (per Level)	1 NoyesBot (per Level)
System Perf.	449 Picks per hour	228 Picks per hour
Process Perf. (PP)	293 Picks per hour	210 Picks per hour
PP, but two pickers	354 Picks per hour	216 Picks per hour
PP, but 30 s	120 Picks per hour	118 Picks per hour
Confirmation Time	120 Picks per hour	118 Picks per hour
PP, but equally distributed products	212 Picks per hour	130 Picks per hour

* Throughput performance is estimated through simulations. Each scenario handles 100 requests with five different SKUs each. Process Scenario: single picker, 10 second pick time, pareto product distribution, medium boxes. Simulated results may vary. For detailed information, contact sales@noyes-tech.com

Power Supply and Connectivity

Power	Schuko-plug (CEE 7/4), 220-240 V (50 / 60 Hz), 16 A
Energy Consumption	Max. 1710 W
Internet	<ul style="list-style-type: none"> • 50 Mbit/s download, 10 Mbit/s upload • via 2.5 GbE LAN

Integration

API	REST-API, no networking required, quickstart templates available to setup integration in hours instead of days.
-----	---

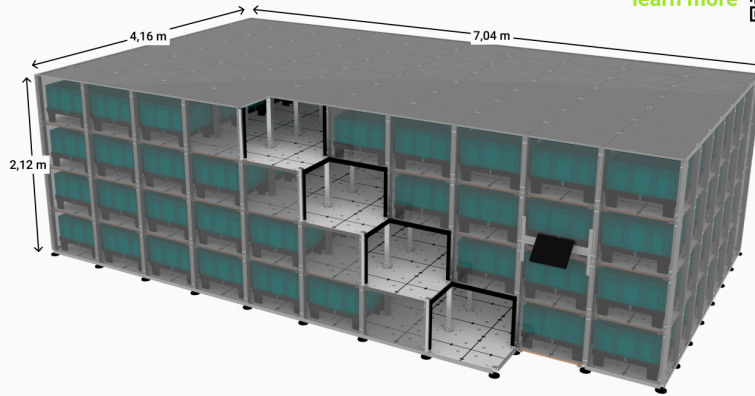
* All product information is subject to change without prior notice.

* The rights of all brand names, product names and trademarks belong to their respective owners.



Store more, pick faster.
NoyesStorage M

Scan to
learn more

Specification

System

Width	7,04 m	# Modules	10
Depth	4,16 m	# Modules	8
Height	1,56 - 2,12 m <small>(depending on chosen level height)</small>	# Modules	4
Footprint	29,3 m ²		
Installation time	5 Days		

Storage Capacity

Large Boxes (600x400)	242 (up to 40 kg each, evenly distributed)
Med. Boxes (300x400)	484 (up to 20 kg each, evenly distributed)
Small Boxes (300x200)	968 (up to 10 kg each, evenly distributed)
Mini Boxes (150x200)	1936 (up to 5 kg each, evenly distributed)
Box Height	<ul style="list-style-type: none"> • 320 mm for high levels • 170 mm for low levels
Maximum Payload	up to 40 kg evenly distributed
Maximum Dimensions	<ul style="list-style-type: none"> • on high levels: 600x400x320 • on low levels: 600x400x170
Module Utilization	79%
Constraints	No dangerous goods according to ADR

* Boxes are not included in this offer. However, suitable boxes for your NoyesStorage system can be provided separately by Noyes Robotics GmbH upon request

Operation Conditions

Temperature	3-30 °C
Humidity	20-90% (non-condensing)
Usage	Indoor

NoyesStorage Component List

Modules	306	Carriers	242
Picking balconies	4	NoyesBots	4-8
Layers high	0-4	Layers low	0-4
Charging Stations	4-8	Commander	1
Platform (with steps)	1	Elevator	0

Throughput Performance

	2 NoyesBots (per Level)	1 NoyesBot (per Level)
System Perf.	433 Picks per hour	240 Picks per hour
Process Perf. (PP)	313 Picks per hour	225 Picks per hour
PP, but two pickers	360 Picks per hour	228 Picks per hour
PP, but 30 s Confirmation Time	120 Picks per hour	119 Picks per hour
PP, but equally distributed products	210 Picks per hour	114 Picks per hour

* Throughput performance is estimated through simulations. Each scenario handles 100 requests with five different SKUs each. Process Scenario: single picker, 10 second pick time, pareto product distribution, medium boxes. Simulated results may vary. For detailed information, contact sales@noyes-tech.com

Power Supply and Connectivity

Power	Schuko-plug (CEE 7/4), 220-240 V (50 / 60 Hz), 16 A
Energy Consumption	Max. 1710 W
Internet	<ul style="list-style-type: none"> • 50 Mbit/s download, 10 Mbit/s upload • via 2.5 GbE LAN

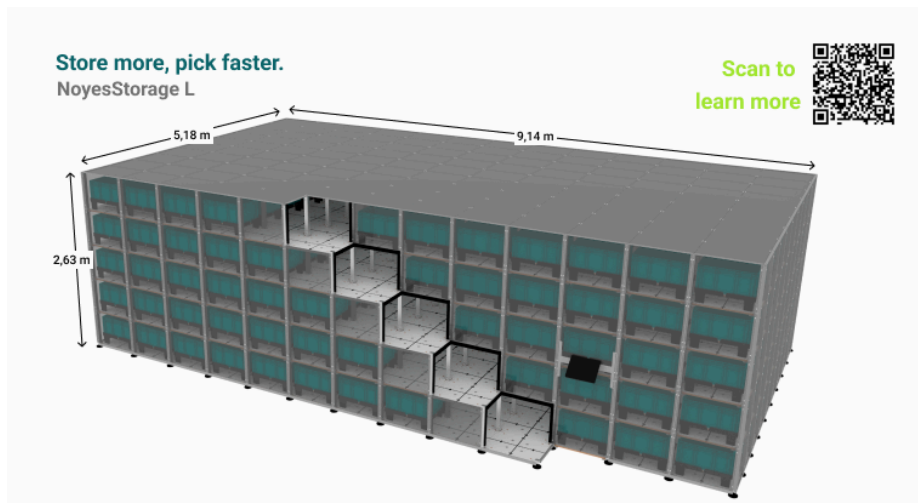
Integration

API	REST-API, no networking required, quickstart templates available to setup integration in hours instead of days.
-----	---

* All product information is subject to change without prior notice.
* The rights of all brand names, product names and trademarks belong to their respective owners.



NoyesStorage L



Specification

System

Width	9,14 m	# Modules	13
Depth	5,18 m	# Modules	10
Height	1,92 - 2,63 m (depending on chosen Level height)	# Modules	5
Footprint	47,3 m ²		
Installation time	10 Days		

Storage Capacity

Large Boxes (600x400)	525 (up to 40 kg each, evenly distributed)
Med. Boxes (300x400)	1050 (up to 20 kg each, evenly distributed)
Small Boxes (300x200)	2100 (up to 10 kg each, evenly distributed)
Mini Boxes (150x200)	4200 (up to 5 kg each, evenly distributed)
Box Height	<ul style="list-style-type: none"> • 320 mm for high levels • 170 mm for low levels
Maximum Payload	up to 40 kg evenly distributed
Maximum Dimensions	<ul style="list-style-type: none"> • on high levels: 600x400x320 • on low levels: 600x400x170
Module Utilization	84%
Constraints	No dangerous goods according to ADR

* Boxes are not included in this offer. However, suitable boxes for your NoyesStorage system can be provided separately by Noyes Robotics GmbH upon request

Operation Conditions

Temperature	3-30 °C
Humidity	20-90% (non-condensing)
Usage	Indoor

NoyesStorage Component List

Modules	625	Carriers	525
Picking balconies	5	NoyesBots	5-10
Layers high	0-5	Layers low	0-5
Charging Stations	5-10	Commander	1
Platform (with steps)	1	Elevator	0

Throughput Performance

	2 NoyesBots (per Level)	1 NoyesBot (per Level)
System Perf.	381 Picks per hour	227 Picks per hour
Process Perf. (PP)	315 Picks per hour	216 Picks per hour
PP, but two pickers	325 Picks per hour	225 Picks per hour
PP, but 30 s Confirmation Time	118 Picks per hour	120 Picks per hour
PP, but equally distributed products	190 Picks per hour	107 Picks per hour

* Throughput performance is estimated through simulations. Each scenario handles 100 requests with five different SKUs each. Process Scenario: single picker, 10 second pick time, pareto product distribution, medium boxes. Simulated results may vary. For detailed information, contact sales@noyes-tech.com

Power Supply and Connectivity

Power	Schuko-plug (CEE 7/4), 220-240 V (50 / 60 Hz), 16 A
Energy Consumption	Max. 1710 W
Internet	<ul style="list-style-type: none"> • 50 Mbit/s download, 10 Mbit/s upload • via 2.5 GbE LAN

Integration

API	REST-API, no networking required, quickstart templates available to setup integration in hours instead of days.
-----	---

* All product information is subject to change without prior notice.

* The rights of all brand names, product names and trademarks belong to their respective owners.

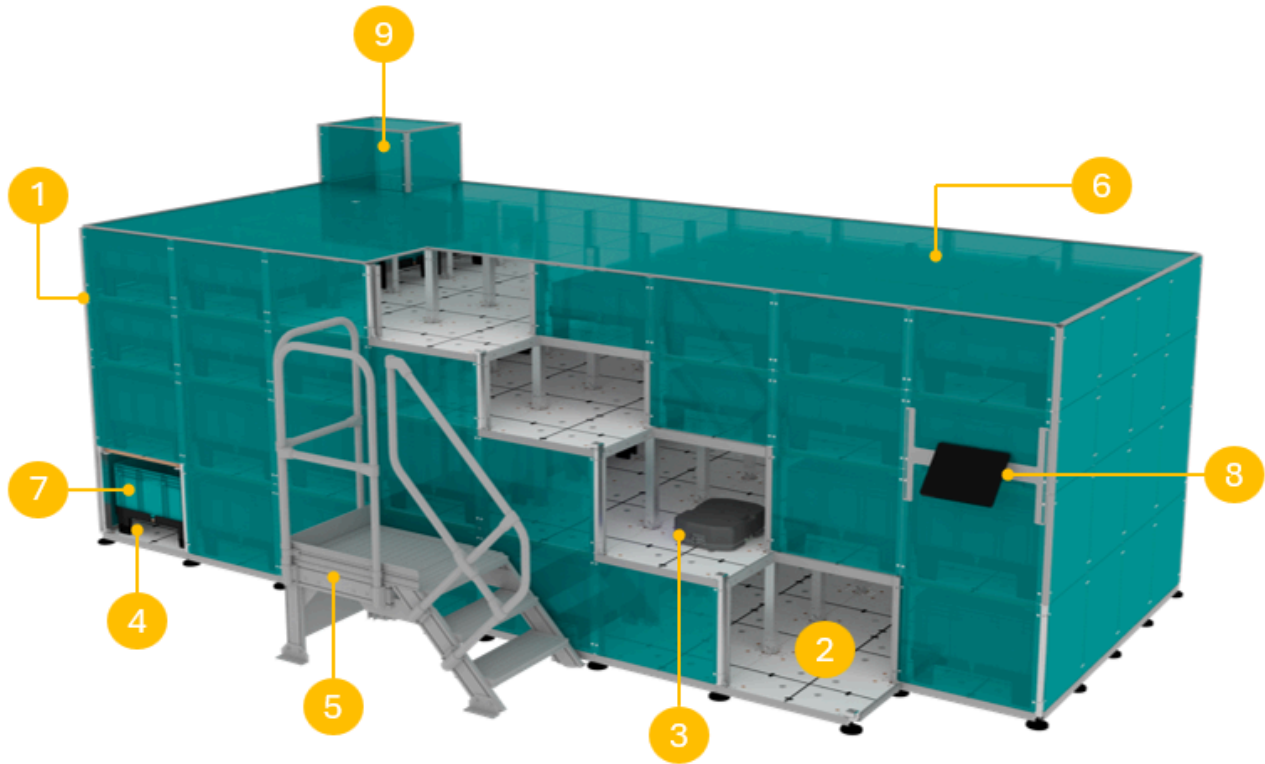


Individuelle Konfigurationen

NoyesStorage kann weiter an individuelle Anforderungen und Gegebenheiten angepasst werden. Für weitere Details wenden Sie sich bitte an sales@noyes-tech.com.

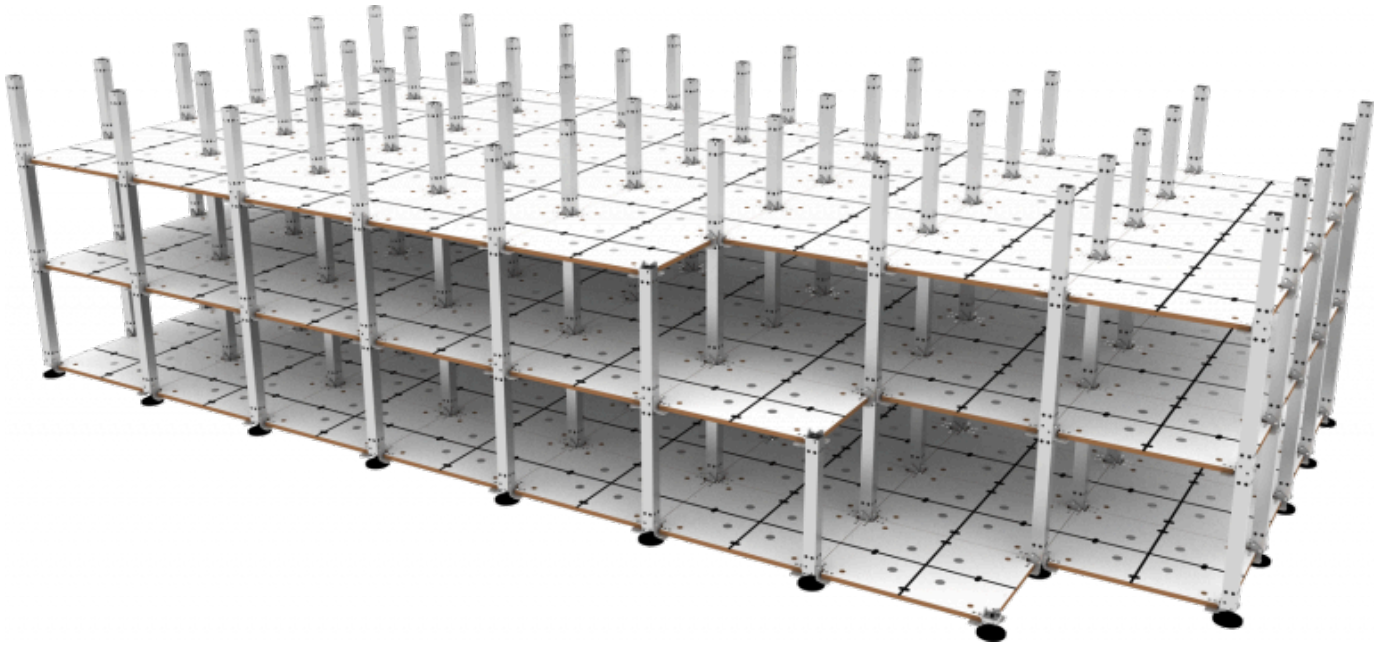
2.3 Komponenten

NoyesStorage erreicht seine Flexibilität durch mehrere Komponenten: ein Regal aus einer modularen Grundplatte, den NoyesBot, die Ladungsträger (auch Carrier genannt) und eine intuitive Benutzeroberfläche. Diese Elemente zusammen bilden den NoyesStorage, der sich an unterschiedliche Durchsatzanforderungen, Lagerbedarfe und Platzverhältnisse anpasst. Die wichtigsten Komponenten – **Regal** (1), **Balkon** (2), **NoyesBot** (3), **Carrier** (4), **Plattform** (5), **Abdeckung** (6), **Box** (7), **Bedienoberfläche** (8) und **Commander** (9) – werden im Folgenden detailliert.



REGAL, ABDECKUNGEN UND PLATTFORM

Das modulare Regalsystem ist das Gerüst des NoyesStorage. Dadurch ist der NoyesStorage anpassungsfähig und passt in jede Gebäudeform. Das Regalsystem ist eine Konstruktion aus über Aluminiumsäulen miteinander verbundene Regalböden (auch genannt Module) aus feuerbeständige nach DIN EN 13501-1 (Klasse C-s2,d0) Holzfaserplatten. Um die gelagerten Artikel vor Verschmutzung und Eindringen von außen zu schützen, ist der NoyesStorage mit transparenten Abdeckungen versehen.



Die Roboter bewegen sich ebenegebunden auf den Modulen und folgen dabei virtuellen Spuren, die durch schwarze Markierungen definiert sind. Die auf den Grundplatten verbauten passiven RFID-Tags speichern verschlüsselte Informationen für die Roboter, um diese z.B. bei der Positionierung zu unterstützen.

Die Module sind durch leichte und robuste Säulen verbunden, die aus gepresstem und CNC-bearbeitetem Aluminium gefertigt werden. Diese Säulen bestimmen die Höhe der einzelnen Ebenen und damit die maximale Höhe der Boxen, die im System gelagert werden können. Mithilfe von Nivelierfüßen können Bodenunebenheiten ausgeglichen werden.

Abgeleitet aus den in der Logistik verwendeten Standardboxen sind Ebenen mit zwei unterschiedlichen Höhen verfügbar. Standardmäßig bieten niedrige Ebenen Platz für Boxen bis zu einer Höhe von 170 mm, während hohe Ebenen Boxen mit einer maximalen Höhe von 320 mm aufnehmen können. Die Zusammensetzung der Ebenen definiert schlussendlich die Gesamthöhe des NoyesStorage. Der Zugriff auf gelagerte Artikel ab der vierten Ebene wird dem Bedienpersonal durch eine Plattform mit Stufen ermöglicht.

Für spezielle Anforderungen bietet der NoyesStorage die Möglichkeit, die Ebenenhöhen individuell anzupassen. Diese flexible Konfiguration ermöglicht die Lagerung von Boxen mit Höhen zwischen 150 mm und 700 mm, wobei die genaue Auslegung auf Ihre spezifischen Bedürfnisse zugeschnitten werden kann.

Die modulare Bauweise des NoyesStorage ermöglicht extrem schnelle Montagezeiten zwischen 3 und 10 Tagen (abhängig von der Konfiguration). Zusätzlich kann das System in Zukunft erweitert werden, um einen höheren Lagerbedarf abbilden zu können.

NOYESBOT

Der NoyesBot spielt eine entscheidende Rolle im NoyesStorage-System und übernimmt die Bewegung der Waren innerhalb des Systems. Er ist speziell darauf ausgelegt, schwere Lasten präzise zu heben und zu transportieren, um reibungslose und effiziente Abläufe zu gewährleisten.



Das präzise Steuersystem des NoyesBot sorgt für eine genaue Navigation. Redundante Lokalisierungstechnologien, die absolute und relative Positionierung kombinieren, erhöhen die Zuverlässigkeit, minimieren das Risiko von Systemunterbrechungen und verbessern die Betriebszeit. Die präzise Steuerung ermöglicht hohe Beschleunigungen, selbst bei der maximalen Traglast von 40 kg, wenn diese gleichmäßig verteilt ist. Der NoyesBot unterstützt zudem ungleichmäßige Lasten bis zu einem Unterschied von 20 kg.

Dank seines energieeffizienten Designs kann der NoyesBot bis zu 6 Stunden ohne Aufladung arbeiten. Mit flexibler Batteriekapazität kann die Laufzeit je nach Betriebsanforderungen auf bis zu 24 Stunden erweitert werden (auf Anfrage). Die sichere LiFePO4-Batterietechnologie ermöglicht einen vollautomatischen Ladevorgang, wodurch manuelle Eingriffe überflüssig werden. Weitere Details zur Batterietechnologie finden Sie im Abschnitt [LiFePO4 Batterien](#).

Der NoyesBot kommuniziert drahtlos über 2,4 GHz Wi-Fi und ermöglicht eine nahtlose Zusammenarbeit mit der Flotte, um den Durchsatz zu optimieren. Eine zentrale Steuereinheit koordiniert die Bewegungen und sorgt für eine effiziente Aufgabenverteilung im gesamten System.

Mit einem Fokus auf Langlebigkeit minimiert der NoyesBot die Anzahl beweglicher Teile, was den Verschleiß reduziert. Diese Designentscheidung unterstützt eine lange Lebensdauer und verringert den Wartungsaufwand, wodurch er zu einem zuverlässigen Bestandteil für den Dauerbetrieb wird.

Tabelle: Detaillierte Angaben zu Gewicht, Abmessungen, Geschwindigkeit, Beschleunigung und anderen Parametern des NoyesBots.

Spezifikation	Wert
Gewicht (inkl. vier Batterien, exkl. Beladung)	20 kg
Maximale Zuladung	40 kg (gleichverteilt, zzgl. 5 kg für Box und Carrier)
Fahrdauer von Modul zu Modul	2,5 s
Maximale Geschwindigkeit	0,8 m/s (ca. 2,88 km/h)
Maximale Beschleunigung	3 m/s ²
Abmaße (W × L × H)	341 mm × 451 mm × 104 mm
Betriebstemperatur	2°C to 35°C
Batterietechnologie	LiFePO ₄
Ladezyklen Batterie	300 – 500 Ladezyklen
Betriebszeit	6 Stunden (bis zu 24 Stunden möglich auf Anfrage)
Ladezeit (0-100%)	1,4 Stunden
Antrieb	Bürstenlose Gleichstrommotoren, 24V

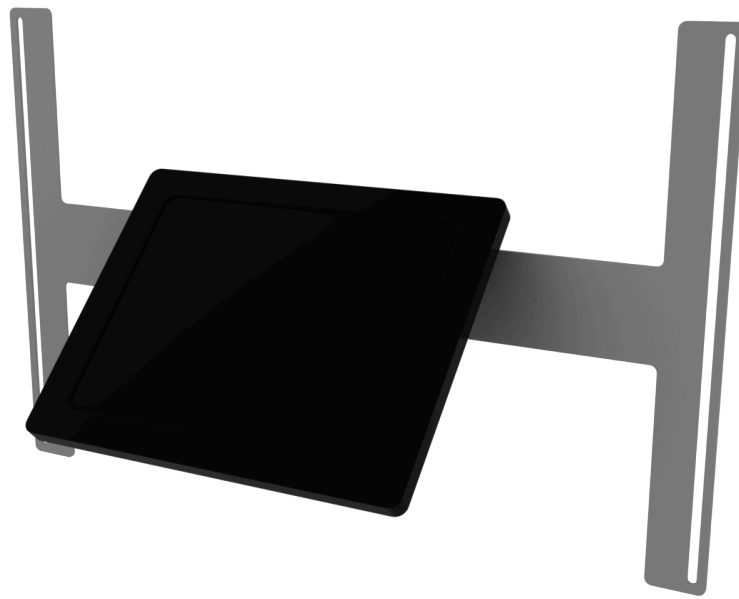
BALKON

Die Kommissionierstation ist der Punkt der Interaktion zwischen dem Bediener und der im NoyesStorage gelagerten Ware. Sie besteht aus mindestens einem Balkon auf jeder Ebene, von dem aus der Bediener auf die ausgelagerten oder einzulagernde Waren zugreifen kann, und der Bedienoberfläche am NoyesStorage. Um Artikel aus dem Lager zu entnehmen, werden sie an die Kommissionierstation geliefert. Die Balkone sind für effiziente Kommissionier- und Nachschubvorgänge konzipiert. Die Ladungsträger werden von Robotern zu diesen Balkonen transportiert.

Ein inherent sicheres Systemdesign gewährleistet die Sicherheit der Bediener ohne zusätzliche Sensorik oder anderer Maßnahmen. Sobald der Roboter einen Träger platziert hat, wird der Bediener durch die Bedienoberfläche zur Interaktion aufgefordert. Die Bediener arbeiten in der Regel parallel auf mehreren Balkonen auf verschiedenen Ebenen. Um die Balkone einwandfrei zu identifizieren sind diese nummeriert. Die Balkone auf höheren Ebenen werden über eine von Noyes bereitgestellte Plattform über Stufen erreicht.

BEDIENBERFLÄCHE

Die Bedienung des NoyesStorage erfolgt über eine moderne Bedienoberfläche. Hierfür steht eine App auf einem 12,9 Zoll großen Tablet zur Verfügung und ermöglicht alle Funktionen zum Betrieb des NoyesStorage. Dabei wird Fokus auf intuitive und einfache Handhabung gelegt, um den Einstieg und das Training zu erleichtern.

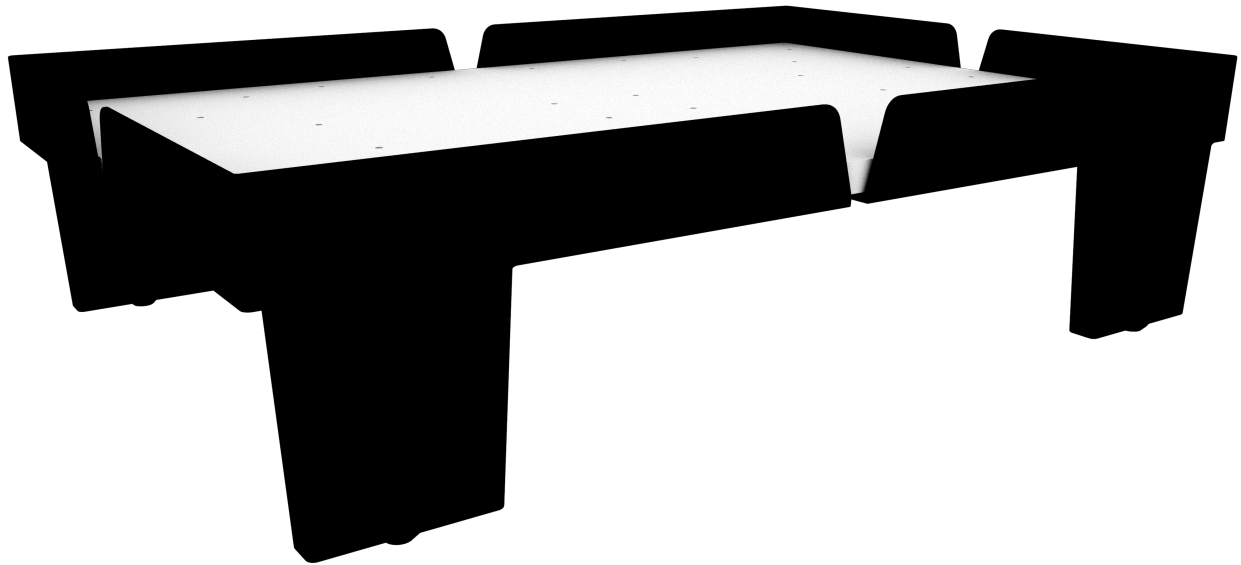


Neben den Kernprozessen wie dem Ein- und Auslagern wird auch das Inventarmanagement abgebildet, das zum Beispiel Einsicht über Bestände gibt und die Anlage neuer SKU-Stammdaten ermöglicht. Weiter sind technische Wartungsfunktionen wie das On- und Offboarding von Robotern (z.B. zur Steigerung des Durchsatzes) oder die Systemsteuerung zum Pausieren oder Stoppen des Systems umgesetzt.

Auf die App hat lediglich zugewiesenes Personal Zugriff, denn für die Nutzung der NoyesStorage App ist ein authentifizierter Nutzer notwendig. Eine Anmeldung erfolgt direkt über die Bedienoberfläche per Benutzername und Passwort. Eine Abmeldung des Users nach Vollendung der Nutzung wird empfohlen, um unbefugte Verwendung zu verhindern.

CARRIER

Carrier fungieren als standardisierter Träger zwischen dem NoyesBot und der gelagerten Waren innerhalb des NoyesStorage. Als Verbindungspunkt für Kundenware, die typischerweise in Kartons verpackt ist, ermöglichen die Carrier dem NoyesBot einen nahtlosen Transport der Artikel zum Balkon. Der Carrier bietet auf einer Grundfläche von 600 x 400 mm die Möglichkeit eine Vielzahl von Verpackungstypen aufzunehmen – von Kartonagen über standardisierte Ladungsträger (z. B. KLT- oder Euroboxen) bis hin zu handelsüblichen Getränkeboxen. Dies ermöglicht eine automatisierte Lagerung ohne zusätzliches Umpacken.



Die Carrier bestehen aus Kunststofffüßen und einer feuerfesten Holzfaserverplatte nach DIN EN 13501-1 (Klasse C-s2,d0). Sie sind so konzipiert, dass sie während des Transports effizient gestapelt werden können. Jeder Träger ist mit einem integrierten, passiven RFID-Tag zur eindeutigen Identifizierung ausgestattet. Die Bodenplatten im Regal bieten definierte Positionen für die Warenträger, so dass diese zentriert bleiben. Dadurch stellen die Träger sicher, dass die Kisten ordnungsgemäß gelagert und sicher im NoyesStorage positioniert sind.

BOX

Auf einem Ladungsträger können eine oder mehrere Boxen gestellt werden, die die zu lagernde Ware beinhalten. Es können bis zu 8 Boxen oder Abteile auf einem Ladungsträger separat genutzt werden, um eine hohe Dichte für Artikel mit geringem Volumen zu ermöglichen. Hierdurch ist eine flexible Konfiguration auf einem Träger möglich. Zum Beispiel: 1 große Box (600 mm x 400 mm), 2 mittlere Boxen (400 mm x 300 mm) oder 1 mittlere und 2 kleine Boxen (300 mm x 200 mm). Abhängig von den zu lagernden Artikeldimensionen kann es erforderlich sein, hohe Boxen (>170 mm) zu verwenden. Diese Boxen können lediglich in hohen Ebenen gelagert werden.

Die folgenden Box-Standards wurden für die Verwendung im NoyesStorage getestet und freigegeben. Noyes Robotics GmbH empfiehlt die Verwendung von Boxen, die diesen Standards entsprechen:

Empfohlene Boxen für den NoyesStorage

Box Name	Größe der Boxen (Länge x Tiefe x Höhe in mm)
R-KLT 6429/RL-KLT 6280	600 x 400 x 280
R-KLT 6415/RL-KLT 6147	600 x 400 x 147
R-KLT 4329/RL-KLT 4280	400 x 300 x 280
R-KLT 4315/RL-KLT 4147	400 x 300 x 147
R-KLT 3215/RL-KLT 3147	300 x 200 x 147
EURO-Box	600 x 400 x 320
EURO-Box	600 x 400 x 170
EURO-Box	400 x 300 x 320
EURO-Box	400 x 300 x 170
EURO-Box	300 x 200 x 170
EURO Box	150 x 200 x 170

Darüber hinaus ist es möglich, kundenspezifische Boxen (oder Kartons) zu verwenden. Bei anderen als den in der obigen Tabelle empfohlenen Boxen ist der Kunde für die Stabilität auf dem Träger und die Kompatibilität im NoyesStorage verantwortlich. Konkret ist darauf zu achten, dass:

- die Box eine Größe von 600 x 400 x 320 mm bei hoher Ebene bzw. 600 x 400 x 170 mm bei niedriger Ebene nicht überschreitet,
- die Box(en) den ganzen Ladungsträger ausfüllen, um ein Verrutschen zu vermeiden (andernfalls sind Anti-Rutsch-Matten zwischen Box(en) und Ladungsträger vorgeschrieben),
- die Box eine maximale Wandschräge von 2° nicht überschreitet (um ein Anecken der Box im NoyesStorage zu vermeiden),
- die Höhe der Box nicht größer ist als deren Breite oder Länge (um ein Umkippen durch einen hohen Masseschwerpunkt zu verhindern) und
- nicht mehrere Boxen übereinander gestapelt werden (um ein Umkippen durch einen hohen Masseschwerpunkt zu verhindern).

Bei Bedarf können Boxen für den NoyesStorage von Noyes Robotics GmbH angeboten werden.

COMMANDER

Der Commander ist die zentrale Steuereinheit des NoyesStorage. Er beherbergt die Hauptrecheneinheit (ein IPC), verwaltet die Kommunikation mit den Robotern und steuert die Stromversorgung der Ladestationen.

Technische Anforderungen - Stromversorgung: Schuko-Steckdose (CEE 7/3) - Spannung: 230 V AC - Frequenz: 50 Hz - Stromstärke: 16 A - Netzwerk: Stabile Internetverbindung - Download: 50 MBit/s - Upload: 10 MBit/s

Sicherheitsaspekte und Stromversorgung Aus Sicherheits- und Betriebsgründen ist der Commander bewusst ohne Ein-/Ausschalter konzipiert. Diese Entscheidung basiert auf folgenden Überlegungen:

- Ladestationen: Der Commander versorgt die Ladestationen mit Strom. Ein ungewolltes Abschalten würde dazu führen, dass sich die Roboter über Nacht entladen und einen erheblichen Mehraufwand im Service verursachen, da diese dann manuell aus dem System entnommen und extern geladen werden müssten.
- Roboterbetrieb: Die Roboter sind batteriebetrieben und bleiben auch bei ausgeschaltetem Commander im Betrieb. Ein Ausschalten des Commanders würde daher keine Änderung des Gefahrenzustands im Servicefall bewirken.
- Systemüberwachung: Die kontinuierliche Überwachung des Systems ermöglicht Noyes einen 24/7-Zugang, um den Status fortlaufend zu überwachen, mögliche Fehler frühzeitig zu erkennen und einen optimalen Service anzubieten.
- Sicherheit: Die Ladespannung von 24 VDC wird als Sicherheitskleinspannung eingestuft und stellt keine Gefahr für den Nutzer dar.

Bei Bedarf kann das Commander-Modul über sein steckbares Kaltgeräte-Kabel vom Stromnetz getrennt werden. Diese Lösung entspricht der Norm EN ISO 60204-1, Abschnitt 5.3.2 e, die eine Stecker/Buchsen-Kombination als "Versorgungstrennvorrichtung" zulässt. Für den IPC (Industrial PC) steht bei Bedarf ein "RESET"-Knopf zur Verfügung, falls ein Neustart erforderlich sein sollte.

2.4 Durchsatz

EINFLUSSFAKTOREN AUF DEN DURCHSATZ

Der NoyesStorage unterstützt über 300 Picks pro Stunde und Kommissionierer. Der erwartete Durchsatz hängt stark von verschiedenen Faktoren ab, darunter die Größe und das Layout des NoyesStorage sowie prozessbezogene Faktoren wie die Handlingszeit, die SKU-Verteilung, die Abwicklungsstrategie (vorgeplant oder auf Abruf), Boxengrößen und weitere Aspekte.

Der folgende Abschnitt bietet einen Überblick über die Schlüsselfaktoren, die den Durchsatz beeinflussen. Das Verständnis dieser Faktoren ist entscheidend, um den erwarteten Durchsatz, der mit dem NoyesStorage in einem bestimmten Logistikprozess erreicht werden kann, realistisch einzuschätzen. Für jede Konfiguration bietet Noyes einen Überblick über den zu erwartenden Durchsatz in Abhängigkeit von den folgenden Einflussfaktoren:

- **Systemgröße:**

NoyesStorage wird in drei Standardkonfigurationen (S, M und L) sowie in kundenspezifischen Konfigurationen auf Anfrage angeboten. Jede Konfiguration unterstützt eine unterschiedliche Anzahl von Balkonen und Systemgrößen. Tiefe Konfigurationen (lange „Autobahn“) sind in der Regel etwas schneller, während breite Konfigurationen (kurze „Autobahn“) im Allgemeinen eine etwas höhere Dichte bieten. Mehr Balkone ermöglichen eine parallele Kommissionierung. Größere Systeme neigen jedoch zu einer höheren durchschnittlichen Entnahmetiefe, was die Entnahmezeit pro Kommissionierung erhöht.

- **Handlingszeit:**

Die Handlingszeit ist die Zeitspanne, die ein Carrier auf dem Balkon verbleibt, bis der Kommissionierer den Pick bestätigt. Während dieser Zeit kann der NoyesStorage bereits einen zweiten Carrier für denselben Balkon vorbereiten. Um den Durchsatz zu maximieren, können Kunden die Anzahl der Picks pro Stunde deutlich erhöhen, indem sie die Handlingszeit pro Pick minimieren. Wir bewerten mehrere Konfigurationen für die Handlingszeiten: 0 s, 10 s, 30 s und 60 s. Die 0-Sekunden-Konfiguration stellt die theoretische Maximalleistung des NoyesStorage dar und schließt menschliche Faktoren aus. Die meisten Kunden arbeiten mit Handlingszeiten von etwa 10 s.

- **SKU-Verteilung:**

Es werden zwei Standard-SKU-Verteilungen definiert: Pareto-Verteilung und Gleichverteilung. In einer gleichverteilten Konfiguration wird angenommen, dass jede SKU mit der gleichen Häufigkeit angefragt wird. Die meisten Kunden weisen jedoch eine Pareto-Verteilung auf, bei der ein kleiner Prozentsatz der Artikel häufig angefragt wird (z. B. entfallen 80% der Anfragen auf 20% der SKUs). Diese Verteilung beeinflusst den Durchsatz, da der NoyesStorage die häufig abgefragten SKUs so positioniert, dass die durchschnittliche Entnahmetiefe minimiert wird.

- **Boxengrößen:**

Ähnlich wie die SKU-Verteilung können auch die Boxengrößen die durchschnittliche Entnahmetiefe beeinflussen. In Umgebungen mit ungleichmäßiger Verteilung führen kleinere Boxen in der Regel zu einer geringeren Entnahmetiefe und somit zu einem höheren Durchsatz.

- **Abwicklungsstrategie:**

Der NoyesStorage kann mehrere Anfragen gleichzeitig bearbeiten. Eine vorgeplante Abwicklungsstrategie, bei der Anfragen im Voraus bekannt sind, sorgt dafür, dass die Roboter kontinuierlich aktiv sind, und unterstützt so einen hohen Durchsatz. Eine Abwicklung auf Abruf führt dagegen häufig zu einer Unterauslastung der Roboter und damit zu einem geringeren Durchsatz.

SIMULATION DES SYSTEMDURCHSATZES

Der erwartete Durchsatz des NoyesStorage wird aus Simulationen abgeleitet, die die reale Systemleistung möglichst genau widerspiegeln. Diese Simulationen modellieren den gesamten Kommissionierprozess – inklusive der Bewegungen der Roboter, der Ladezyklen und der Handlingszeiten – um präzise Leistungsprognosen zu liefern.

Für jede Produktkonfiguration werden 100 Anfragen an den NoyesStorage übergeben, die jeweils 5 unterschiedliche Artikel zum Picken enthalten. Dies ergibt insgesamt 500 Picks. Der Test beginnt mit der ersten Anfrage und endet, sobald der letzte Pick abgeschlossen ist. Anschließend wird der Durchsatz berechnet, indem die Gesamtanzahl der Picks durch die gemessene Simulationszeit und die Anzahl der Kommissionierer geteilt wird. Diese simulierten Ergebnisse sind für jede [Produktkonfiguration](#) dokumentiert.

Die folgende Tabelle gibt einen Überblick über die simulierten Parameter – inklusive des Basisszenarios und zusätzlicher Parameter, die für jede Konfiguration evaluiert wurden.

Parameter	Beschreibung	Basisszenario	Weitere Parameter
Anzahl der Kommissionierer	Anzahl der Kommissionierer, die aktiv Ware picken.	2	1
Handlingszeit	Zeit des Kommissionierers zwischen zwei Picks am NoyesStorage.	10 s	0 s und 30 s
SKU-Verteilung	Häufigkeit der Artikel in den Kundenordern.	Pareto-Verteilung	Gleichverteilung
Kistengröße	Durchschnittliche Größe der Kisten im System.	Medium	Groß und klein

2.5 Sicherheitshinweise

Der NoyesStorage wurde nach den allgemein anerkannten Regeln der Technik sowie den geltenden Vorschriften konstruiert und gefertigt. Dennoch besteht die Gefahr von Personen- und Sachschäden, wenn Sie dieses Kapitel und die Sicherheitshinweise nicht beachten.

- Lesen Sie diese Dokumentation sorgfältig und vollständig, bevor Sie mit dem NoyesStorage arbeiten.
- Bewahren Sie die Dokumentation so auf, dass sie für alle Benutzer jederzeit zugänglich ist.

2.5.1 Sachgemäße Verwendung

Der in diesem Handbuch beschriebene NoyesStorage ist so konstruiert und gebaut, dass es bei bestimmungsgemäßen Gebrauch sicher funktioniert.

Der NoyesStorage darf nur in der vom Hersteller angegebenen Weise ordnungsgemäß betrieben werden. Der bestimmungsgemäße Gebrauch des NoyesStorage ist die sichere Auslagerung von Gütern mithilfe von Robotern bis hin zur Kommissionierung an Picking Stations (Balkone), bei der ein Picker/Refiller intuitiv durch den Prozess geführt wird. Der NoyesStorage darf nur an den dafür vorgesehenen Stellen betrieben werden. Die Aufstellorte werden durch den Betreiber bestimmt. Zur bestimmungsgemäßen Verwendung gehört auch die Beachtung aller Hinweise in dieser Anleitung. Jede über die bestimmungsgemäße Verwendung hinausgehende oder andersartige Benutzung gilt als nicht bestimmungsgemäß. Das NoyesStorage ist nur in dem Zustand, in dem es ausgeliefert wurde, von den zuständigen Prüfstellen geprüft worden und darf daher nur in dem Zustand, in dem es ausgeliefert wurde, bestimmungsgemäß verwendet werden. Es ist verboten, konstruktive, mechanische oder elektrische Änderungen ohne erneute Sicherheitsprüfung vorzunehmen. Ein Beispiel hierfür ist die Demontage und Montage.

Jede Art von Veränderung am System muss z. B. von Noyes Robotics GmbH vorgenommen werden:

- Anbauen von Modulen
- Änderung der Verkabelung
- Modifikation von Modulen/Änderung des Layouts
- Demontage von Komponenten, mit Ausnahme von Abdeckungen im pausierten System im Falle einer Störung. Anweisungen zur (De-)Montage finden Sie in der Installationsanleitung.

Sie verwenden den NoyesStorage korrekt, wenn:

- er an den dafür vorgesehenen Stellen verwendet wird
- die [Inspektions- und Wartungsvorschriften](#) eingehalten werden
- eine Risikobeurteilung durch den Betreiber vor der „Betriebsbereiten Übergabe (BBÜ)“ durchgeführt wurde
- die [Weiteren Systemspezifikationen](#) eingehalten werden
- er in technisch einwandfreiem Zustand betrieben wird

Zur bestimmungsgemäßen Verwendung gehört auch, dass Sie diese Dokumentation und insbesondere die [Produkt- und technologieabhängigen Sicherheitshinweise](#) vollständig gelesen und verstanden haben.

Eine andere Verwendung als die hier beschriebene, sachgemäße Verwendung entspricht nicht der bestimmungsgemäßen Verwendung und ist daher nicht zulässig. Noyes Robotics GmbH übernimmt keine Verantwortung für Schäden, die durch unsachgemäßen Gebrauch entstehen. Die Risiken einer nicht bestimmungsgemäßen Verwendung liegen ausschließlich beim Betreiber.

Beachten Sie auch: - nicht ins Innere des NoyesStorage zu gehen oder zu klettern - den NoyesStorage nicht zu beklettern oder betreten - Unbefugten keinen Zugang zum NoyesStorage zu gewähren - den NoyesStorage nicht ohne Abdeckungen zu betreiben - nichts in den NoyesStorage zu stellen, außer auf den Ladungsträger - keine Gegenstände auf dem NoyesStorage abzustellen - nichts einzulagern, was selbst ein Risiko darstellen könnte, wie gefährliche Güter, instabile oder explosive Chemikalien, leicht brennbare und radioaktive Materialien - keine Gegenstände in Boxen zu stellen, die die Höhe der Boxen überschreiten - keine Tiere oder Menschen im NoyesStorage zu lagern



HINWEIS

Lediglich der Maintainer darf nach erfolgreich absolviertem Training zur Fehlerbehebung – nach entsprechenden sicherheitsgerichteten Vorkehrungen – in den NoyesStorage klettern.

2.5.2 Allgemeine Sicherheitsinformationen

- Beachten Sie die geltenden Vorschriften zur Unfallvermeidung und zur Arbeitssicherheit.
- Beachten Sie die Sicherheitsvorschriften und -bestimmungen des Landes, in dem der NoyesStorage verwendet/angewendet wird.
- Benutzen Sie den NoyesStorage nur in technisch einwandfreiem Zustand.
- Beachten Sie alle Hinweise zum Aufbau des NoyesStorage.
- Personen, die den NoyesStorage montieren, bedienen, demontieren oder warten, dürfen nicht unter dem Einfluss von Alkohol, anderen Drogen oder Medikamenten stehen, die ihr Reaktionsvermögen beeinflussen.
- Verwenden Sie nur von Noyes Robotics GmbH freigegebene Zubehör- und Ersatzteile, um Personengefährdungen durch ungeeignete Ersatzteile auszuschließen.
- Beachten Sie die in der Produktdokumentation angegebenen technischen Daten und Umgebungsbedingungen.
- Werden ungeeignete Komponenten in sicherheitsrelevante Anwendungen eingebaut oder eingesetzt oder wird die sicherheitsrelevante Funktion zweckentfremdet genutzt, können unbeabsichtigte Betriebszustände auftreten, die Personen- und/oder Sachschäden verursachen können. Stellen Sie sicher, dass insbesondere sicherheitsrelevante Anlagenteile nicht verändert oder beeinträchtigt werden.
- Beachten Sie alle Warn- und Gefahrenhinweise in der Betriebsanleitung.
- Sie dürfen das Produkt erst dann in Betrieb nehmen, wenn festgestellt wurde, dass der NoyesStorage den länderspezifischen Vorschriften, Sicherheitsregeln und Normen für die Anwendung entspricht.

2.5.3 Produkt- und technologieabhängige Sicherheitshinweise

Dieser Abschnitt enthält Sicherheitshinweise, die für den gesamten NoyesStorage gelten



GEFAHR

Verletzungsgefahr durch sich bewegende Teile!

Beim Hineinklettern in den NoyesStorage oder der Demontage von Abdeckungen besteht Verletzungsgefahr durch bewegende Teile.

- Unbefugte dürfen den NoyesStorage zu keiner Zeit betreten.
- Abdeckungen an den Seiten und der Decke des Systems dürfen von Unbefugten nicht demontiert werden.
- Der NoyesStorage ist in einen Ruhezustand zu bringen, bevor er von geschultem Personal betreten wird.



WARNUNG

Verletzungsgefahr durch sich bewegende Teile!

Bei der Bedienung besteht Verletzungsgefahr durch sich bewegende Teile.

- Unbefugten ist der Zugang zum NoyesStorage nicht gestattet.
- Zugangskontrolle des NoyesStorage durch den Store Manager.
- Nur Benutzer mit gültigem Passwort können den NoyesStorage bedienen.

 **WARNUNG**

Verletzungsgefahr und Sachschäden durch Kontakt mit gefährlichen Stoffen!

Bei Arbeiten am NoyesStorage besteht die Gefahr des Kontakts mit gefährlichen Stoffen.

- Keine Lagerung gefährlicher Güter gemäß der UN-Liste gefährlicher Güter.

 **WARNUNG**

Verletzungsgefahr durch Stolpern!

Wenn Kabel nicht in den Kabelkanälen gelegt werden und lose im Arbeitsbereich liegen, besteht Stolper- und Verletzungsgefahr.

- Kabel am Boden des Arbeitsbereiches sind nicht gestattet.
- Kabel am Boden auf Arbeitswegen sind zu vermeiden oder mit Kabelbrücken abzudecken.
- Alle Sicherheits- und Warnhinweise sind zu beachten.

 **WARNUNG**

Verletzungsgefahr durch Rauch und Hitze durch Feuer!

Wenn Flüssigkeiten oder Fremdkörper mit elektrischen Geräten in Berührung kommen, besteht Verletzungsgefahr, wenn Rauch oder Feuer ausbricht.

- Gegenstände sind nicht auf dem NoyesStorage abzustellen.
- Im Bereich der Picking Stations ist nichts abzustellen.
- Flaschen im NoyesStorage sind vor Beschädigungen zu sichern, um das Auslaufen von Flüssigkeit zu vermeiden.

 **WARNUNG**

Verletzungsgefahr durch herabfallende Teile des NoyesStorage!

Wenn Abdeckungen an den Außenwänden des NoyesStorage geöffnet sind, können herabfallende Teile schwere Verletzungen verursachen.

- Der NoyesStorage ist nicht ohne Abdeckungen zu betreiben.
- Alle Sicherheits- und Warnhinweise sind zu beachten.

 **WARNUNG**

Verletzungsgefahr durch sich bewegende Teile bei Störung des laufenden Betriebs!

Während des Betriebs besteht Verletzungsgefahr durch sich bewegende Teile – sowohl horizontale Bewegung des Roboters und Ladungsträgers, als auch vertikale Bewegung bei Anheben/Abstellen des Ladungsträgers.

- Der Benutzer darf nicht nach bewegenden Teilen im NoyesStorage greifen.
- Der Benutzer muss abwarten, bis ein Roboter den Ladungsträger abgestellt hat, bevor er sich dessen annähert.

 **WARNUNG**

Verletzungsgefahr, wenn Flaschen durch Druck zerbrechen und Teile der beschädigten Flaschen den Benutzer verletzen!

Wenn der Benutzer druckbeaufschlagte Flaschen ungesichert in den NoyesStorage einlagert, besteht die Gefahr, dass Flaschen zerbrechen und der Benutzer verletzt werden.

- Druckflaschen oder Flaschen, die durch Bewegung unter Druck geraten können, sind nicht im NoyesStorage zu lagern.

 **WARNUNG**

Verletzungsgefahr durch Kurzschluss!

Während Wartungsarbeiten an elektrischen Komponenten besteht Verletzungsgefahr durch Kurzschlussstrom, wenn das Wartungspersonal metallische Gegenstände an der Hand trägt.

- Trennen Sie alle Stromquellen, bevor Sie Wartungsarbeiten durchführen.
- Das Wartungspersonal darf keine metallischen Gegenstände an Händen oder Armen tragen.
- Alle Sicherheits- und Warnhinweise sind zu beachten.

 **WARNUNG**

Verletzungsgefahr durch herabfallende Teile von zerbrochenen Gegenständen, die auf den Benutzer geschleudert werden!

Wenn der Kunde andere als die von Noyes Robotics GmbH empfohlenen Boxen verwendet, kann es zu Verletzungen durch herausgeschleuderte Teile von herunterfallenden Gegenständen kommen.

- Die verwendeten Ladungsträger werden von Noyes Robotics GmbH bereitgestellt, verwenden Sie keine anderen.
- Boxen dürfen immer nur direkt auf dem Ladungsträger platziert werden.
- Es dürfen von Benutzern zu keinem Zeitpunkt andere Gegenstände als die bereitgestellten Ladungsträger auf die Bodenplatten des NoyesStorage platziert werden.

 **WARNUNG**

Verletzungsgefahr durch Herunterfallen vom NoyesStorage!

Klettert jemand am NoyesStorage herauf oder auf ihn drauf, können Personen abstürzen und sich verletzen.

- Es darf nicht am NoyesStorage hochgeklettert werden.
- Die Abdeckung des NoyesStorage darf nicht betreten werden.
- Die Balkone des NoyesStorage dürfen nach Inbetriebnahme nicht zum Erreichen von höheren Ebenen verwendet werden.
- Um höhere Ebenen des Storage zu erreichen, darf ausschließlich die mitgelieferte Treppe verwendet werden.

 **WARNUNG**

Verletzungsgefahr durch Absturz aus höher gelegenen Ebenen!

Wenn der Benutzer in einer der höher gelegenen Ebenen arbeitet, besteht die Gefahr des Absturzes aus höher gelegenen Ebenen.

- Geeignete Mittel verwenden, um höher gelegene Ebenen sicher zu erreichen.
- Eine geeignete Treppe zu Erreichung höherer Ebenen wird mitgeliefert und ist zu verwenden.
- Lagern von Langsamdrehern möglichst auf den oberen Ebenen.
- Alle Sicherheits- und Warnhinweise sind zu beachten.

 **WARNUNG**

Verletzungsgefahr für Kopf oder Oberkörper des Benutzers durch herausgeschleuderte Teile!

Wenn Gegenstände höher als die Oberkante der Box auf dem Ladungsträger gelagert werden, besteht Verletzungsgefahr durch herabfallende oder herausgeschleuderte Gegenstände.

- Gegenstände sind nicht höher als die Oberkante der Box auf dem Träger zu lagern.
- Alle Sicherheits- und Warnhinweise sind zu beachten.

 **WARNUNG**

Verletzungsgefahr durch Unachtsamkeit oder Bewusstlosigkeit!

Bei Arbeiten am NoyesStorage besteht Verletzungsgefahr, wenn der Benutzer bewusstlos wird oder unachtsam ist und stürzt.

- Benutzung der vorgesehenen Treppe, um höher gelegene Ebenen zu erreichen.
- Lagern von Langsamdrehern möglichst auf den oberen Ebenen.
- Sorgfältiges und konzentriertes Arbeiten.

 **WARNUNG**

Verletzungsgefahr durch fahrende Roboter im Storage!

Steigen befugte Personen in den NoyesStorage, um Fehler zu beheben, können fahrende Roboter diese anfahren, was zu Verletzungen und Quetschungen am Körper führen kann.

- Ausschließlich befugtes Personal darf in den NoyesStorage steigen.
- Das befugte Personal hat den Anweisungen zum Betreten des NoyesStorage zu folgen und damit einhergehende Vorkehrungen zu treffen.

VORSICHT

Verletzungsgefahr für Finger/Hände/Arme beim Quetschen!

Wenn der Benutzer seine Finger/Hände/Arme zwischen bewegliche Teile (NoyesBot/Ladungsträger/Box) und feste Teile des Lagers steckt, besteht Verletzungsgefahr.

- Niemals die Finger/Hände/Arme zwischen bewegliche Teile und feste Teile des NoyesStorage stecken oder halten.
- Der Benutzer muss abwarten, bis ein Roboter den Ladungsträger abgestellt hat, bevor er sich dessen annähert.
- Nachdem ein Roboter einen Ladungsträger und die darauf befindliche Box anhebt, müssen die Tätigkeiten abgebrochen werden und der Benutzer muss seine Finger/Hände/Arme aus dem Balkonbereich entfernen.
- Nachdem ein Roboter einen Ladungsträger und die darauf befindliche Box anhebt, darf nicht mehr nach gelagerten Teilen vom Benutzer gegriffen werden.

VORSICHT

Verletzungsgefahr durch fahrende Roboter!

Der Roboter kann über Hände/Finger bei Berührungen der Bodenplatten des NoyesStorage fahren, was zu Verletzungen der Benutzer führen kann.

- Benutzer sollen zu keiner Zeit, während dem Betrieb, die Bodenplatten des NoyesStorage berühren.
- Balkone des NoyesStorage sind stets freizuhalten.

VORSICHT

Verletzungsgefahr durch sich bewegende Roboter!

Während des Betriebs kann das Arbeitspersonal zwischen die Komponenten des NoyesStorage geraten und sich an den beweglichen Robotern (z. B. beim Verlassen des Balkons) verletzen.

- Das Betreten des NoyesStorage ist vom Benutzer zu unterlassen.
- Der Benutzer muss abwarten, bis ein Roboter den Ladungsträger abgestellt hat, bevor er sich dessen annähert.
- Der Benutzer darf nicht unter einen Ladungsträger greifen.
- Der Benutzer darf Roboter nicht berühren, ohne dedizierte Anweisung über die NoyesStorage App (beim Off-/Onboarding).
- Der NoyesStorage darf nur mit angebrachten Schutzvorrichtungen um die Balkonbereiche betrieben werden.

VORSICHT

Verletzungsgefahr durch bewegliche Teile am Balkon!

Während des Betriebs kann das Arbeitspersonal zwischen die NoyesStorage-Komponenten und einen beweglichen Ladungsträger geraten und sich verletzen.

- Legen oder halten Sie niemals Ihre Finger/Hände/Arme zwischen bewegliche Teile und feste Teile des NoyesStorage.

VORSICHT

Verlust und Beschädigung von persönlichen Teilen!

Geraten Kleidung/Accessoires in den NoyesStorage und kommen diese mit beweglichen Teilen in Kontakt, kann es dazu führen, dass diese in den NoyesStorage hereingezogen und beschädigt werden oder Teile des NoyesStorage beschädigt werden.

- Benutzer des NoyesStorage sollen keine losen Kleidungsstücke tragen und auf lose oder scharfkantige Accessoires verzichten.

VORSICHT

Verletzungsgefahr durch eingezogene Haare!

Geraten Haare des Benutzers in den NoyesStorage und kommen diese mit beweglichen Teilen in Kontakt, kann es dazu führen, dass diese in den NoyesStorage hereingezogen werden.

- Mit dem Betrieb des NoyesStorage ist sicherzustellen, dass Benutzer geschlossene oder kurze Haare während der Interaktion mit dem NoyesStorage haben.

VORSICHT

Schmerzen nach längerer Nutzung

Für Benutzer kann die Bedienung des untersten Levels des NoyesStorage nach mehreren Arbeitsstunden zu Schmerzen führen.

- Lagern von Langsamdrehern auf der untersten Ebene, um häufiges Bücken zu reduzieren.

HINWEIS

Endkunden können erkranken wegen des Verzehrs vorzeitig verdorbener Lebensmittel

Eingelagerte Lebensmittel können wegen zu langer Lagerung oder falscher Lagertemperatur verderben.

- Der Betreiber des NoyesStorage ist verantwortlich, Verderblichkeit und Temperatureinhaltungen von Lebensmitteln sicherzustellen und zu überwachen.

2.5.4 Pflichten des Betreibers

BETREIBER

Betreiber ist diejenige Person, die den NoyesStorage zu gewerblichen oder wirtschaftlichen Zwecken, Nutzung/Betrieb durch einen Dritten zulässt und die rechtliche Produktverantwortung für den Schutz des Bedienpersonals trägt.

WEITERE PFLICHTEN DES BETREIBERS

Der NoyesStorage wird gewerblich genutzt. Der Betreiber der NoyesStorage unterliegt somit den gesetzlichen Pflichten zur Arbeitssicherheit.

Neben den Sicherheitshinweisen in dieser Anleitung sind auch die am Standort des NoyesStorage geltenden Sicherheitshinweise, Unfallvermeidungsvorschriften und Arbeitssicherheitshinweise zu beachten.

Der Betreiber des NoyesStorage muss sein Personal regelmäßig zu den folgenden Themen schulen: - Beachtung und Anwendung der Betriebsanleitung sowie der gesetzlichen Vorschriften - Beachtung der Vorschriften der Berufsgenossenschaft - Bestimmungsgemäße Verwendung/Betrieb - Unsachgemäße Verwendung/Bedienung - Erstellung einer eigenen Gefährdungsbeurteilung - Regelmäßige Sicherheitsunterweisungen seiner Mitarbeiter - Beachtung der Anweisungen der Betriebsanweisung des Betreibers - Verhalten in Notfällen

2.5.5 Qualifikation und Rollen des Bediersonals

Der NoyesStorage darf nur von Personen bedient werden, die mindestens 16 Jahre alt sind, eingewiesen/geschult wurden und die aufgrund ihrer geistigen und körperlichen Verfassung für die Bedienung solcher Anlagen geeignet sind. Sie müssen außerdem in der Lage sein, die Folgen ihres Handelns zu überblicken.

Personen, die den NoyesStorage bedienen, müssen die Landessprache oder Englisch verstehen können.

Für den Betrieb des NoyesStorage werden drei Rollen unterschieden. Diese können von der gleichen Person ausgefüllt werden:

Picker

Der Picker ist eine geschulte Person, die den NoyesStorage bedient. Der Picker hat vor der Benutzung ein Training vom zuständigen Store Manager erhalten. Diese Rolle ist die einfachste und verfügt über die wenigsten Rechte.

Der Picker ist zuständig für den Kommissioniervorgang, das heißt die Entnahme bzw. Einlagerung von Artikeln am NoyesStorage und die entsprechende Bestätigung am Bedienpanel, sowie wenige Zusatzprozesse (siehe [Betrieb – Bedienung](#)). Der Picker hat keine Möglichkeiten, außerhalb des geführten Ein- und Auslagerungsprozesses Änderungen am Storage vorzunehmen.

Dos:

- **Grundschulung anwenden:** Nutzt das grundlegende Wissen aus der Grundschulung, um die NoyesStorage App sicher zu bedienen.
- **Kommissionieren:** Kommissioniert und lagert Artikel gemäß den Aufträgen in der NoyesStorage App ein.
- **Probleme melden:** Informiert den Store Manager oder Techniker sofort über auftretende Probleme oder Unklarheiten.

Don'ts:

- **Keine Änderungen:** Nimmt keine Änderungen an SKU-Stammdaten, Verteilungen oder Box-Zuweisungen vor.
- **Kein technischer Eingriff:** Versucht nicht, technische Probleme eigenständig zu beheben oder erweiterte UI-Funktionen zu nutzen.
- **Keine Schulung:** Überlässt die Schulung neuer Kommissionierer ausschließlich dem Store Manager.

Store Manager

Der Store Manager ist eine trainierte Person, die ein Training von der Noyes Robotics GmbH oder einem anderen Store Manager erhalten haben muss. Diese Rolle verfügt über alle Nutzungsrechte. Zusätzlich zu den Berechtigungen des Pickers hat der Store Manager erweiterte Rechte. Diese umfassen Prozesse zur [erweiterten Bedienung](#), sowie die Prozesse für [Wartung, Reparatur und Reinigung](#).

Darüber hinaus ist der Store Manager verantwortlich für das Training von Personen, die das System bedienen sollen, gemäß der Betriebsanleitung. Der Store Manager erhält eine spezielle Schulung durch Noyes Robotics GmbH und ist verpflichtet, alle Personen, die Zugang zum NoyesStorage haben, zu schulen.

Darüber hinaus ist der Store Manager verantwortlich für die Einhaltung der Bedienungsanleitung.



HINWEIS

Der Store Manager ist dafür verantwortlich, nur geschultem Personal den Zugang zum NoyesStorage zu gewähren.

Dos:

- **Systemwartung:** Führt Wartungsaufgaben wie das Neustarten des Systems und das An- oder Abmelden von Bots durch.
- **Picker schulen:** Schult neue Picker und befähigt sie, das System sicher und effizient zu nutzen.
- **Aufträge und Verteilungen:** Erstellt und verwaltet Aufträge sowie die Verteilung von SKUs und Boxen innerhalb des Systems.
- **Problemunterstützung:** Unterstützt Picker bei der Lösung einfacher technischer Probleme innerhalb der NoyesStorage App.

Don'ts:

- **Keine physischen Eingriffe:** Führt keine physischen Eingriffe am System durch; diese sind ausschließlich dem Maintainer vorbehalten.
- **Keine unautorisierten Änderungen:** Nimmt keine Änderungen vor, die über den autorisierten Aufgabenbereich hinausgehen.

Maintainer

Um Supportzeiten zu reduzieren, kann optional ein Training zum Maintainer stattfinden. Dieses Training bietet weitere Einblicke in den Prozess zur Fehlerbehebung. Es ist zwingend erforderlich, um Probleme im Inneren des Systems selbständig oder lediglich mit remote Support von Noyes Robotics GmbH zu beheben.

Dos:

- **Systemzugriff bei Problemen:** Betritt den Storage nur in technischen Notfällen, um physische Probleme zu beheben.
- **Abdeckungen öffnen:** Öffnet bei Bedarf die Abdeckungen des Storage, um Störstellen zu erreichen, und stellt sicher, dass die Abdeckungen nach Behebung des Problems wieder ordnungsgemäß angebracht werden.
- **Technische Fehler beheben:** Behebt physische und technische Probleme eigenständig oder gemäß den Anweisungen des Noyes Supports.
- **Sicherheitsrichtlinien befolgen:** Hält sich strikt an alle Sicherheitsregeln und Anweisungen, um Risiken zu minimieren.
- **Systemzugriff minimieren:** Betritt den NoyesStorage nur, wenn es absolut notwendig ist, und stets unter Einhaltung der Sicherheitsrichtlinien.

Don'ts:

- **Keine Änderungen an Ladestationen:** Nimmt niemals Änderungen an den Ladestationen des Systems vor oder entfernt sie.
 - **Kein Eingriff in die Stromversorgung:** Trennt oder öffnet niemals die Stromversorgung (PowerSupply des Storage).
-

2.5.6 Persönliche Schutzausrüstung

Die persönliche Schutzausrüstung wurde von der Noyes Robotics GmbH wie folgt festgelegt:

- Bei Verwendung zum Ein- und Auslagern ist keine Schutzausrüstung erforderlich.
- Zur Montage des NoyesStorage sowie bei Wartungsarbeiten müssen Handschuhe getragen werden.
- Beim On- und Offboarding von Robotern sind Sicherheitsschuhe zu tragen.
- Tragen Sie bei Wartung, Reparatur und Reinigung Handschuhe.

2.6 Weitere Systemspezifikationen

TEMPERATUR, FEUCHTIGKEIT UND LUFTBEDINGUNGEN

Der NoyesStorage kann in einem Temperaturbereich von 3 °C bis 30 °C, ohne schnelle Veränderung, betrieben und zwischen -25 °C und 45 °C gelagert oder transportiert werden. Die Luftfeuchtigkeit muss zwischen 20 und 90 % liegen und darf nicht kondensieren. Der NoyesBot ist IP30, Commander und übrige Systemkomponenten sind IP20 konform und entsprechend für den Betrieb in abgeschlossenen Räumen ausgelegt.

BETRIESHÖHE

Der Betrieb ist bis zu einer Höhe von 2000 m über N.N. zugelassen. Oberhalb von 2000 m ist eine gesonderte Prüfung und Freigabe erforderlich.

BODENBELASTBARKEIT

Notwendige zulässige Bodenbelastung in Abhängigkeit des durchschnittlichen Füllgewichts pro Carrier

Anzahl Ebenen	20 kg Füllgewicht	30 kg Füllgewicht	40 kg Füllgewicht
3 Ebenen	302 kg/m ²	392 kg/m ²	482 kg/m ²
4 Ebenen	401 kg/m ²	521 kg/m ²	641 kg/m ²
5 Ebenen	500 kg/m ²	650 kg/m ²	800 kg/m ²
6 Ebenen	599 kg/m ²	779 kg/m ²	959 kg/m ²
7 Ebenen	698 kg/m ²	908 kg/m ²	1118 kg/m ²
8 Ebenen	797 kg/m ²	1037 kg/m ²	1277 kg/m ²
9 Ebenen	896 kg/m ²	1166 kg/m ²	1436 kg/m ²
10 Ebenen	995 kg/m ²	1295 kg/m ²	1595 kg/m ²
11 Ebenen	1094 kg/m ²	1424 kg/m ²	1754 kg/m ²
12 Ebenen	1193 kg/m ²	1553 kg/m ²	1913 kg/m ²

BRANDLAST UND BRANDSCHUTZ

Zur Bestimmung der Brandlast des NoyesStorage sind die signifikanten Materialbestandteile aufgelistet. Die Feuerfestigkeitsklassen der oberen 98 % sind angegeben. Die exakten Werte variieren je nach Konfiguration des NoyesStorage.

Brandlast und Feuerfestigkeitsklassen für ein L-Storage:

Material	Masseanteil	Feuerfestigkeitsklasse
MDF Platten	59.6 %	DIN EN 13501-1 C-s2,d0
Aluminium	13.3 %	DIN EN 13501-1 A1
Polycarbonat Lastträger	11.9 %	UL94-5VA
Stahl	11.2 %	DIN EN 13501-1 A1
Polycarbonat Abdeckungen	2.3 %	DIN EN 13501-1 B-s2,d0
LiFePO ₄	0.4 %	
Elektr. Motoren	0.4 %	
Anderes	0.8 %	

GERÄUSCHEMISSION

Die niedrige Geräuschemission des NoyesStorage erfordert im Alleinbetrieb keine persönliche Schutzausrüstung. Der Kunde ist dafür verantwortlich, eine personengebundene Lärmmessung nach LärmVibrationsArbSchV durchzuführen, wenn das System in einer Umgebung mit erhöhten Fremdgeräuschen sowie Reflexionsschall installiert wird.

Pegelwerte nach LärmVibrationsArbSchV und EG-Maschinenrichtlinie

Alle Messungen wurden an der unter Höchstlast laufenden Maschine unter Einsatzbedingungen durchgeführt.

Definition des Pegelwertes	Messwert	Anforderung
Emissions-Schalldruckpegel L_{pA}	≤ 70 dB(A)	nach EG-Maschinenrichtlinie
Lärmexpositionspegel $L_{ex,8h}$	≤ 70 dB(A)	< 80 dB(A) nach LärmVibrationsArbSchV
Spitzenschallpegel $L_{pC,peak}$	≤ 80 dB(A)	< 135 dB(A) nach LärmVibrationsArbSchV

CONNECTIVITY

NoyesStorage benötigt eine stabile Internetverbindung über ein LAN-Kabel mit einer Mindestbandbreite von 50 Mbit/s im Download und 10 Mbit/s im Upload. Niedrigere Geschwindigkeiten können die Systemverfügbarkeit beeinträchtigen. Eine Internetverbindung ermöglicht zusätzliche Funktionen wie kontinuierliche Software-Updates, webbasierten Zugriff und verbesserte Reaktionszeiten.

SAFETY, COMPLIANCE AND AVAILABILITY

NoyesStorage ist CE-konform und ausschließlich für die Bedienung durch geschulte Personen ab 16 Jahren ausgelegt.

CE-Konformitätserklärungen stehen als Download bereit ([Deutsch](#), [Englisch](#))

Die NoyesBots werden von sicheren, UN 38.3-zertifizierten Lithium-Eisenphosphat-Batterien angetrieben, die auf Zuverlässigkeit ausgelegt und hergestellt sind. Die Roboter können nach UN 3481 (für Lithium-Eisenphosphat-Batterien in Geräten, >100Wh) versandt werden.

Die Lagerung gefährlicher Güter (nach der [ADR-Liste für gefährliche Güter](#)) ist unter gewissen Umständen möglich. Voraussetzung hierfür ist, dass der Kunde vorab eigenverantwortlich eine Gefährdungsbeurteilung gemäß ArbSchG/BetrSichV durchführt und insbesondere die Folgen eines möglichen Brandes berücksichtigt. Basierend auf den Ergebnissen dieser Beurteilung sind entsprechende Schutzmaßnahmen (z.B. Einsatz geprüfter Gefahrgutbehälter) vom Kunden eigenständig umzusetzen. Die Verantwortung für die Einhaltung und Wirksamkeit dieser Maßnahmen liegt vollständig beim Kunden.

NoyesStorage ist in Deutschland, der Schweiz und Österreich verfügbar.

LIFEPO4 BATTERIEN

Die im NoyesBot verwendeten Batterien sind Lithium-Eisenphosphat-Akkumulatoren ($LiFePO_4$). Es handelt sich um eine 8S-Konfiguration mit weniger als 100 Wh gespeicherter Energie. Sie steht für Sicherheit, Zuverlässigkeit und hohe Leistung.

Maximale Sicherheit

$LiFePO_4$ ist eine der sichersten verfügbaren Lithium-Batteriechemien:

- **Kein thermisches Durchgehen:** Anders als andere Lithium-Ionen-Typen ist $LiFePO_4$ hochstabil, selbst unter Belastung. Im Gegensatz zu Li-Ion (NMC, LCO, usw.) oder LiPo (Lithium-Polymer), die oft mit $LiFePO_4$ verwechselt werden.
- **Extrem geringes Brandrisiko:** Es widersteht Überhitzung, Überladung und Durchstichschäden im Vergleich zu anderen Chemien. Es entzündet sich nicht unter Belastung, da es von Natur aus nicht brennbar ist. Daher ist es die bevorzugte Lösung aus versicherungstechnischer Sicht. Sogar Blei-Säure-Batterien können während des Ladens Wasserstoffgas abgeben oder Säure auslaufen lassen, was sie für automatisierte Lagersysteme weniger geeignet macht.
- **Batteriemanagementsystem (BMS):** Das BMS stellt sicher, dass alle Arten von Fehlern zu einer Abschaltung der Batterie führen und hält die Batterie ausbalanciert und sicher.

Lange Lebensdauer

$LiFePO_4$ hält mehrere tausend Batteriezyklen im Vergleich zu Li-Ion (~1000 Zyklen) und Blei-Säure (~400 Zyklen). Dies gewährleistet eine lange Laufzeit der Roboter ohne notwendige Wartung.

Kontinuierliche Leistung

- Konstante Spannung während des gesamten Entladezyklus gewährleistet kontinuierlichen Betrieb.
- Die Batterie funktioniert gut in einem weiten Temperaturbereich zwischen 2°C und 40°C Umgebungstemperatur.

Umweltfreundlich

- Ungiftige und recycelbare Materialien.
- Kein Kobalt oder Nickel – reduziert Umwelt- und ethische Bedenken.

Leicht und kompakt

- Der 8S-Pack ist unter 100 Wh, macht ihn flugtauglich und einfach zu transportieren.
- Hohe Energiedichte gewährleistet lange Laufzeit im Vergleich zu einer Blei-Säure-Batterie.

Zusammenfassung:

LiFePO₄ ist die sicherste und zuverlässigste Batterietechnologie für moderne Robotik. Sie gewährleistet, dass der Roboter länger, sicherer und intelligenter läuft – mit weniger Sorgen und mehr Leistung.

Links für weitere Informationen:

[Battery University - Summary Table of Lithium-based Batteries](#)

3. IT & Cybersicherheit

3.1 Übersicht

3.1.1 Zweck

Diese Sektion gibt IT-Managern eine kompakte Orientierung für die technische Vorprüfung des NoyesStorage.

3.1.2 Was Sie hier klären

1. Wie ist die Netzgrenze zwischen Kundennetz und dem NoyesStorage aufgebaut?
2. Welche Verbindungen müssen freigegeben werden?
3. Wie sind Benutzerzugriffe, Systemintegrationen und externe Zugänge organisiert?
4. Welche Punkte müssen vor dem Go-Live formal abgehakt sein?

3.1.3 Leseplan (ca. 15 Minuten)

1. [Netzwerk & Internetanbindung](#)
2. [Zugriff & Sicherheitskonzept](#)
3. [Go-Live Checkliste](#)

3.1.4 Internetzugang zum System

Der NoyesStorage benötigt im Regelbetrieb eine aktive Internetanbindung.

So wurde der Internetzugang in Projekten bereits umgesetzt:

- Separates Netzwerksegment zwischen Kundennetz und dem NoyesStorage (empfohlen).
- Direkte Verbindung des NoyesStorage ins Kundennetz.
- Internetzugang über einen separaten 5G/4G/Starlink-Router (vom Kunden bereitgestellt).

Offline-Betrieb ist möglich, aber nur als Sonderfall:

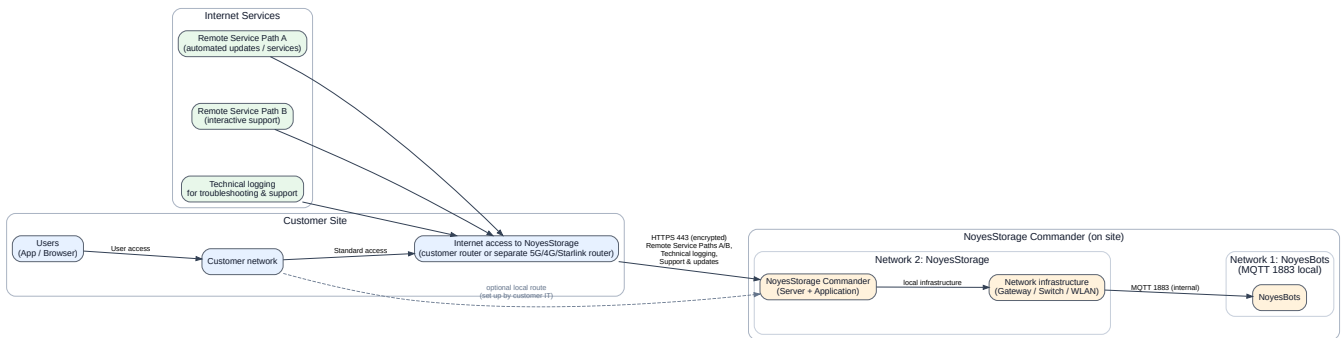
- Bestimmte Funktionen sind nicht verfügbar.
- Kontinuierliche Sicherheits- und Software-Updates können nicht durch Noyes eingespielt werden.
- Der sichere Betrieb während Offline-Phasen muss vom Kunden sichergestellt werden.

3.2 Netzwerk & Internetanbindung

3.2.1 Zweck

Diese Seite beschreibt, wie der NoyesStorage an das Kundennetz angebunden wird und welche Verbindungen für Betrieb, Support und Updates erforderlich sind.

3.2.2 Systemgrenze auf hoher Ebene



3.2.3 Netzwerkverantwortung

- Noyes bringt am Standort eine eigene Netzwerkinfrastruktur und Server mit (Commander).
- Diese interne Noyes-Infrastruktur wird durch Noyes betrieben, konfiguriert und gesichert.
- Kunden-IT verantwortet das Kundennetz, den Internetzugang sowie die Freigaben an der Schnittstelle zum NoyesStorage.

3.2.4 Schnittstelle zum Kundennetz

Für die IT-Freigabe ist der Übergang zwischen Kundennetz und dem NoyesStorage entscheidend. Interne Gerätekommunikation innerhalb der Noyes-Infrastruktur muss nicht durch die Kunden-IT konfiguriert werden. Der Transparenz halber ist sie unten dennoch separat aufgeführt. Der Zugriff aus dem Kundennetz auf den NoyesStorage erfolgt im Regelfall über den Internetpfad. Zusätzlich können lokale Routen eingerichtet werden; das erfolgt ausschließlich durch die Kunden-IT.

3.2.5 Verbindungen an der Schnittstelle (durch Kunden-IT freizugeben)

Richtung	Port	Protokoll	Von	Nach	Zweck	Für Betrieb erforderlich
Eingehend	443	HTTPS/WSS	Autorisierte Benutzer/Systeme	NoyesStorage Commander	App- und API-Zugriff, verschlüsselt per TLS 1.2 oder höher	Ja
Eingehend	80	HTTP	Autorisierte Benutzer/Systeme	NoyesStorage Commander	Redirect auf HTTPS	Ja
Ausgehend	443	HTTPS	NoyesStorage Commander	Support- und Update-Endpunkte	Remote-Support und Updates	Ja
Ausgehend	53	UDP	NoyesStorage Commander	DNS-Service	Namensauflösung	Ja
Ausgehend	123	UDP	NoyesStorage Commander	NTP-Service	Zeitsynchronisierung	Ja

3.2.6 Interne Noyes-Verbindungen (nur zur Transparenz)

Richtung	Port	Protokoll	Von	Nach	Zweck
Intern	1883	MQTT	NoyesBots-Netz	Commander-Broker-Endpunkt	Interner Geräte- und Robotikpfad

3.2.7 Anbindung externer Kundensysteme

Externe Kundensysteme können über die freigegebene Schnittstelle an den NoyesStorage angebunden werden. Für die Erstbewertung sind dabei vor allem die folgenden Punkte relevant:

Thema	Aktueller Stand
Transportweg	HTTPS über Port 443
Verschlüsselung	TLS, Mindeststandard TLS 1.2
Benutzerzugriff	Über authentifizierte Benutzerkonten
Systemintegration	Über API-Schlüssel
Netzwerkempfehlung	Separates Netzwerksegment
Projektspezifische Sonderanforderungen	Werden im Projekt separat geklärt

3.2.8 Internetanbindung ist erforderlich

Der NoyesStorage benötigt im Regelbetrieb eine aktive Internetanbindung.

Gründe:

- schneller Remote-Support,
- kontinuierliche Sicherheits- und Software-Updates,
- internetabhängige Funktionen (z. B. E-Mail-Datenexport).

3.2.9 Übliche Varianten für die Internetanbindung

In Projekten wurden bisher folgende Varianten umgesetzt:

- Separates Netzwerksegment zwischen Kundennetz und dem NoyesStorage (empfohlen).
- Direkte Verbindung des NoyesStorage ins Kundennetz.
- Internetzugang über einen separaten 5G/4G/Starlink-Router (vom Kunden bereitgestellt).

Offline-Betrieb ist nur als Sonderfall möglich. Dabei stehen einzelne Funktionen nicht zur Verfügung, Sicherheits- und Software-Updates können nicht kontinuierlich durch Noyes bereitgestellt werden, und der sichere Betrieb muss in dieser Phase durch den Kunden gewährleistet werden.

3.3 Zugriff & Sicherheitskonzept

3.3.1 Zweck

Diese Seite beschreibt den aktuellen Stand von Zugriff, Sicherheitsmaßnahmen und externen Zugängen für die erste IT-Bewertung.

3.3.2 Benutzerzugriff und Rollen

Feld	Aktueller Stand
Rollenmodell	Für den Kundenbetrieb stehen <code>Admin</code> und <code>User</code> zur Verfügung
Standardzugang für Benutzer	HTTPS über Port 443, bereitgestellt über einen verschlüsselten Tunnel mit TLS-Passthrough
Authentifizierung für Benutzer	Über persönliche Benutzerkonten; im Anwendungskontext werden Zugriffstoken verwendet
Noyes Service-Account	Wird durch Noyes für Service- und Supportzwecke eingerichtet
Passwortpolicy	Mindestens 12 Zeichen sowie mindestens 1 Ziffer, 1 Sonderzeichen, 1 Kleinbuchstabe und 1 Großbuchstabe

Hinweis: SSO und MFA sind derzeit nicht implementiert, können bei Bedarf aber als Erweiterung beauftragt werden. Zusätzlich bestehen systemseitig weitere interne Rollen, die nicht Teil des regulären Kundenbetriebs sind.

3.3.3 Systemintegration

Feld	Aktueller Stand
Integrationspfad	HTTPS über Port 443
Authentifizierung für Integrationen	API-Schlüssel
Verwaltung von API-Schlüsseln	Können durch Admins im System selbst erstellt werden
Netzwerkempfehlung	Separates Netzwerksegment zwischen Kundennetz und dem NoyesStorage

3.3.4 Externe Zugänge

Für Remote-Service und Betriebsstabilität bestehen zwei getrennte, redundante Zugangswege:

- Pfad A: gesicherter Fernzugang für automatisierte Service-, Patch- und Update-Prozesse.
- Pfad B: separater, gesicherter Fernzugang für interaktive Support-Sitzungen.

3.3.5 Sicherheits- und Betriebsmaßnahmen

Der NoyesStorage wird mit festgelegten Sicherheitsmaßnahmen betrieben. Die folgende Übersicht fasst die wichtigsten Schutzmechanismen für den laufenden Betrieb zusammen.

Bereich	Aktueller Stand
Firewall	Host- und Netzfreigaben werden nach definierten Regeln umgesetzt
Verschlüsselung bei Übertragung	TLS für externe Anwendungspfade, Mindeststandard TLS 1.2
Endpoint-Schutz	Endpoint-Überwachung ist vorhanden
Malware-Schutz	Malware-Kontrollen sind vorhanden
Systemhärtung	CIS-orientierte Härtungsbaseline
Updates	Regelbetrieb mit kontinuierlicher Update-Fähigkeit

3.4 Go-Live Checkliste

3.4.1 Zweck

Diese Checkliste bündelt die Punkte, die vor dem produktiven Start gemeinsam bestätigt werden müssen.

3.4.2 Pflichtpunkte vor Go-Live

Punkt	Verantwortlich	Freigabe durch
Netzwerkübersicht freigegeben	Kunden-IT	IT-Leitung / Projektverantwortung
Firewall-Regeln freigegeben	Kunden-IT	IT-Sicherheit / Netzwerk
Freigabeliste externer Ziele (FQDN/IP) freigegeben	Kunden-IT	IT-Sicherheit / Netzwerk
Internetanbindung für den Betrieb bestätigt	Kunden-IT	IT-Leitung
Benutzer- und Rollenmodell für den Kundenbetrieb bestätigt	Kunden-IT	IT-Leitung
Verantwortliche für Benutzer- und Rollenverwaltung benannt	Kunden-IT	IT-Leitung
Verantwortliche für API-Schlüssel und Systemintegrationen benannt	Kunden-IT	IT-Leitung
Incident- und Eskalationskontakte benannt	Kunden-IT + Noyes	Betriebsverantwortliche
Remote-Support-Zugang und Freigabeprozess abgestimmt	Kunden-IT + Noyes	Gemeinsame Freigabe

3.4.3 Vertragsbezug

Mit Kauf und Inbetriebnahme gilt diese Checkliste als vereinbarte betriebliche Grundlage, sofern keine abweichenden Sonderregelungen vertraglich festgehalten wurden.

3.4.4 Abschlusskriterium

Die initiale IT-Freigabe ist abgeschlossen, sobald alle Pflichtpunkte datiert, zugeordnet und freigegeben sind.

4. Bedienung & Integration

4.1 Bedienung über App

4.1.1 Übersicht

Dieses Kapitel beschreibt die Bedienung des NoyesStorage über die App der Noyes Robotics GmbH, die über das Bedienpanel zugänglich ist. Im Folgenden werden die Bedienungsmöglichkeiten erläutert. Vom Startbildschirm der App können alle verfügbaren Bedienungsfunktionen erreicht werden:

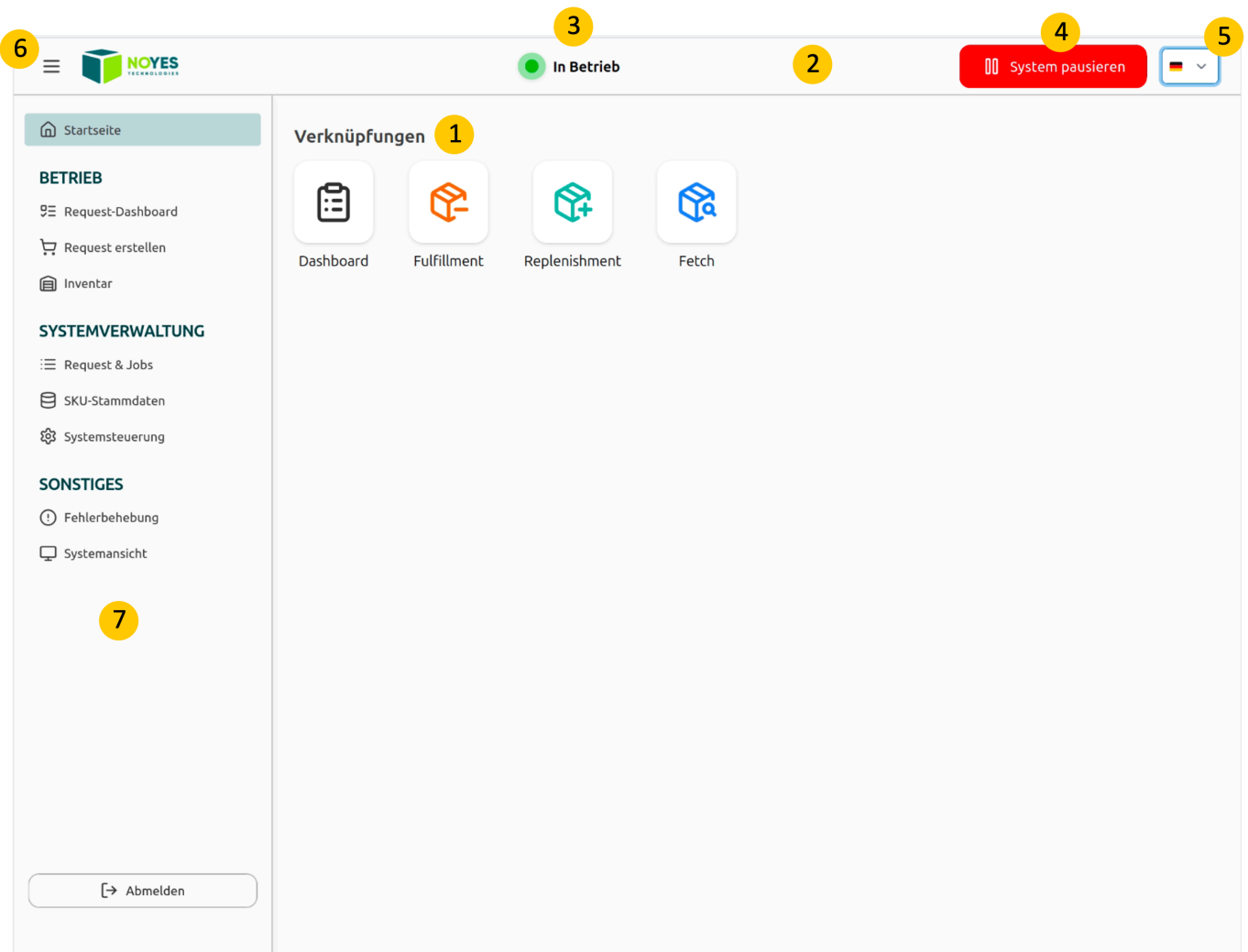


Abbildung: Startbildschirm der App zur Bedienung des NoyesStorage

1. **Verknüpfung zur Auftragserstellung und -abarbeitung**

2. **Kopfzeile**, die zu jeder Zeit mit allen Funktionalitäten zur Verfügung steht

3. **Statusanzeige**: Zeigt zu jeder Zeit den Betriebsstatus des NoyesStorage an:

- **Grün**: In Betrieb
- **Gelb**: Pausiert
- **Rot**: Fehler
- **Grau**: Aus

4. **System pausieren**: Der Betrieb kann jederzeit pausiert werden und führt zum Stillstand des NoyesStorage

5. **Sprachauswahl**: Auswahl der Sprache

6. Auf- und Einklappen der Menüleiste

7. **Menüleiste**:

- **Request-Dashboard**: [Abarbeitung erstellter Requests](#)
- **Request erstellen**: [Zur Erstellung von Fulfillment- oder Replenishment-Requests](#)
- **Inventar**: Übersicht über im NoyesStorage eingelagerte Artikel und Bestände
- **Requests & Jobs**: Übersicht über die Auftragshistorie
- **SKU-Stammdaten**: Stammdatenübersicht und -erstellung
- **Systemsteuerung**: Wartungsfunktionen des NoyesStorage
- **Fehlerbehebung**: Übersicht von Fehlern, Warnungen und Informationen mit Lösungsanweisungen sowie Zugang zum Support
- **Systemansicht**: Digitaler Zwilling des NoyesStorage und dessen Aktivitäten
- **Logout**: des angemeldeten Benutzers

4.1.2 Authentifizierung

Dieser Prozess stellt sicher, dass nur autorisiertes und geschultes Personal Zugang zur NoyesStorage App erhält. Administratoren (Admin oder Super Admin) können Benutzer direkt in der App unter **Einstellungen > Benutzer** anlegen und verwalten. Falls kein Administrator verfügbar ist, unterstützt der Noyes-Support über support@noyes-tech.com.

AUFGABEN UND VERANTWORTLICHKEITEN

- **Store Manager:** Hat Zugriff auf alle Funktionen der NoyesStorage App, einschließlich der Wartungsfunktionen. Ihm stehen sämtliche erforderlichen Zugangsfunktionen zur Verfügung.
 - **Picker:** Ist ausschließlich für die Kommissionierung und Auffüllung von Artikeln zuständig.
-

ANMELDUNG

Hinweis

Sollten Sie noch keine Anmeldedaten erhalten haben, wenden Sie sich an Ihren Administrator. Ist Ihnen dieser nicht bekannt, können Sie auch den Support kontaktieren: support@noyes-tech.com.

1. Öffnen Sie die NoyesStorage App. Falls noch ein anderer Benutzer angemeldet ist, kann dieser **über das Menü abgemeldet werden**.
2. Geben Sie Ihre E-Mail-Adresse (1) sowie Ihr Passwort (2) ein. Bestätigen Sie die Eingabe mit dem Login-Button (4).

Welcome to NoyesStorage



Login

Email

support@noyes-tech.com



1

Password



2

[Forgot password?](#)

3

Login

4

3. Der Login ist nun abgeschlossen.

NEUE BENUTZER ANLEGEN

1. Administratoren (Admin oder Super Admin) können neue Benutzer direkt in der NoyesStorage App anlegen. Öffnen Sie hierzu die Navigation und wählen Sie [Einstellungen](#) > [Benutzer](#). Klicken Sie auf [Benutzer hinzufügen](#), tragen Sie Benutzernamen, E-Mail-Adresse und Rolle ein und speichern Sie den Datensatz.

In Betrieb System pausieren

Benutzerverwaltung

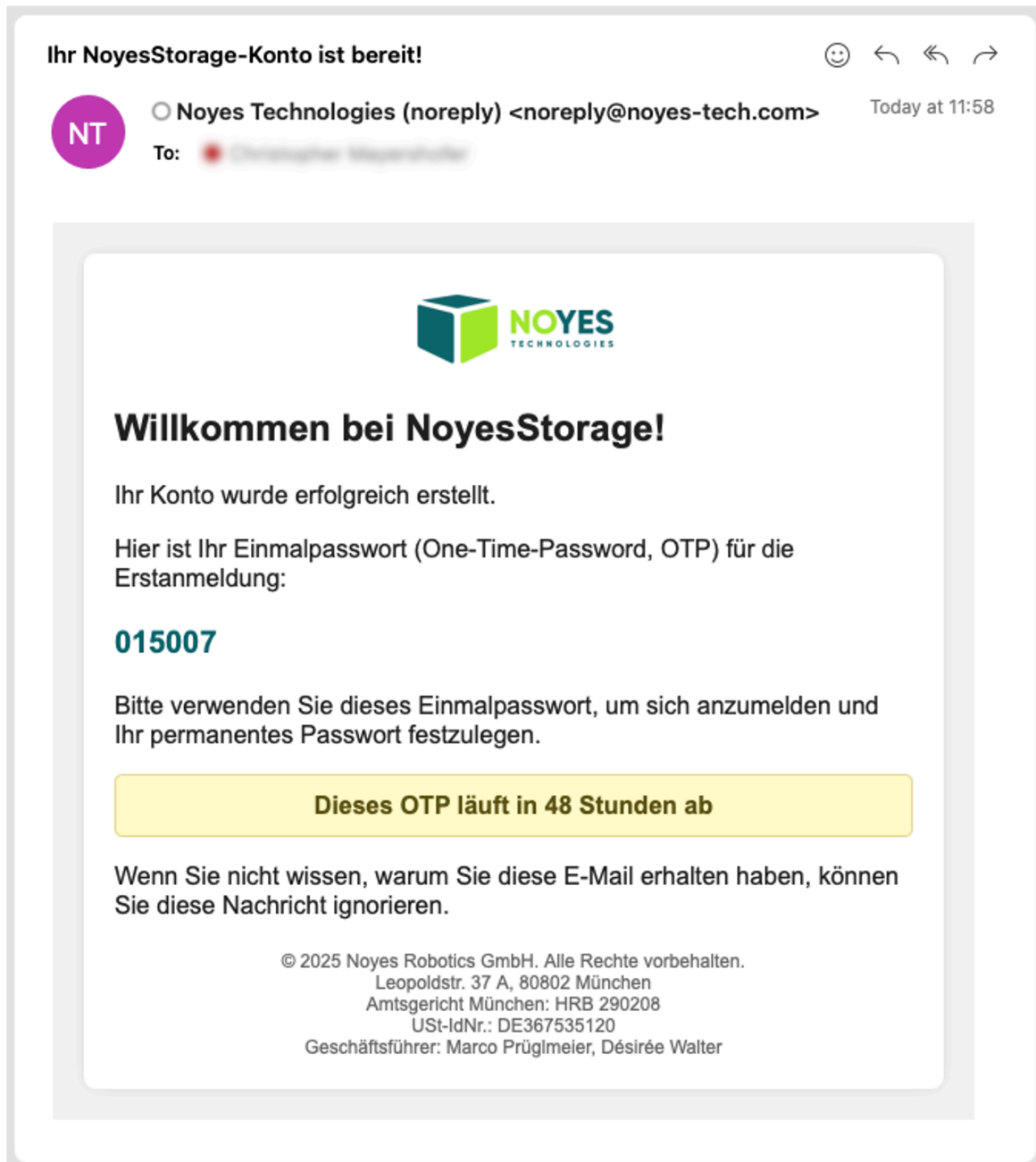
Suchen Sie nach bestehenden Benutzern oder erstellen Sie neue.

Suchen... Alle Filter löschen + Benutzer hinzufügen Benutzer bearbeiten

<input type="checkbox"/>	ID	Benutzername	E-Mail	Rolle	Passwort läuft ab am	OTP	
<input type="checkbox"/>	2	Noyes	test@gmail.com	Admin	N/A	888279	
<input type="checkbox"/>	3	Noyes Hotmail	test@hotmail.com	Super Admin	19 May 2026 21:09	-	
<input type="checkbox"/>	1	Noyes Support	support@noyes-tech.com	Super Admin	29 Mar 2027 10:43	-	

Angezeigte Elemente: 3 / 3 - Ausgewählt: 0 Zeile(n)

2. Daraufhin erhält der neue Benutzer eine E-Mail mit einem Einmal-Passwort (One-Time-Password, OTP) für die erste Anmeldung.



Hinweis für Systeme ohne Internetverbindung

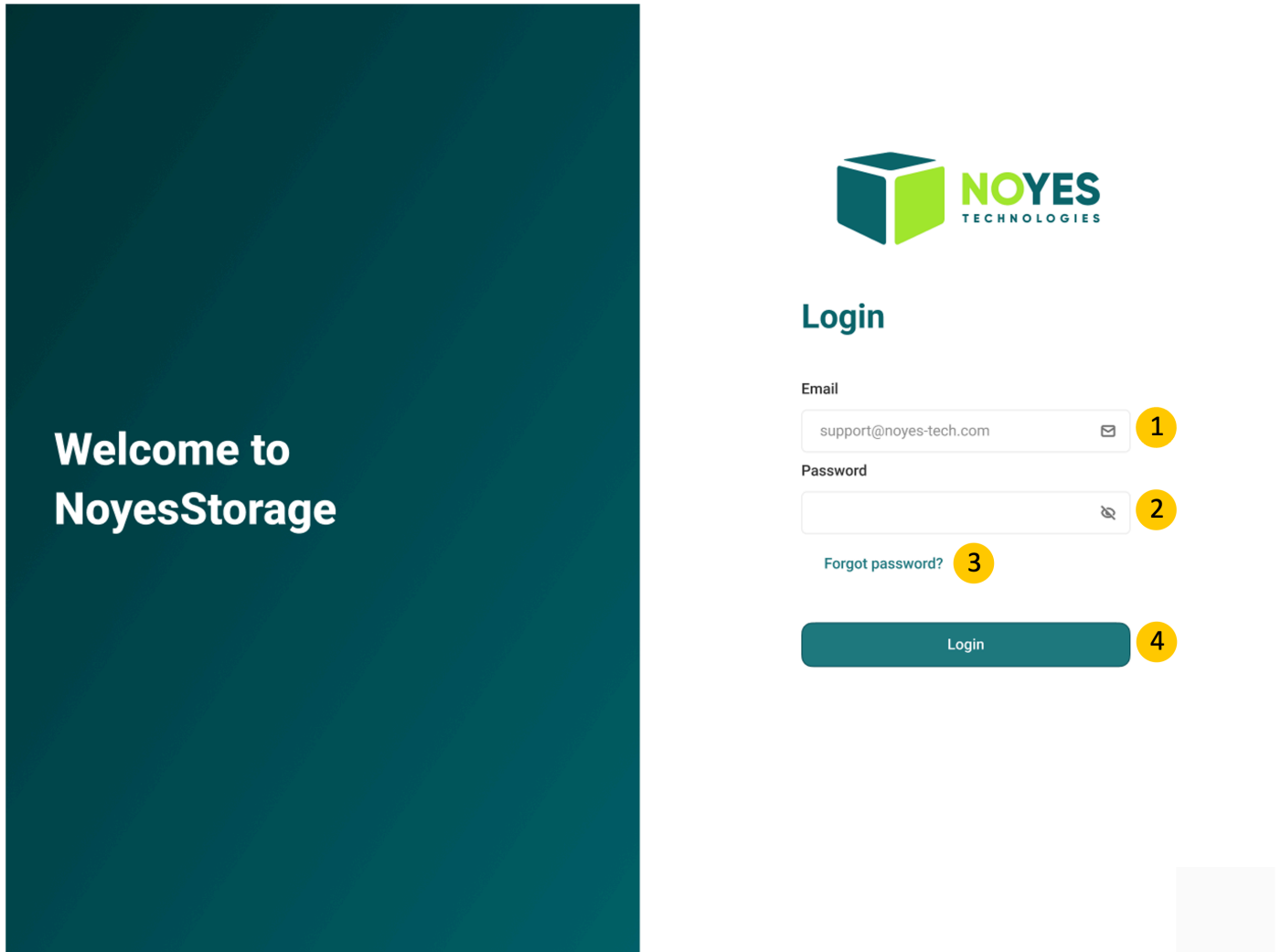
Bei Systemen ohne E-Mail-Versand kann der Administrator das OTP in der Benutzertabelle unter [Einstellungen](#) > [Benutzer](#) in der Spalte [OTP](#) einsehen und dem neuen Benutzer über einen sicheren Kanal mitteilen.

Wichtig

Das One-Time-Password ist nur für eine Dauer von 48 Stunden gültig. Wird diese Zeit überschritten, kann der Benutzer über die Funktion [Passwort vergessen](#) ein neues OTP anfordern. Alternativ wird beim Versuch, sich mit einem abgelaufenen OTP anzumelden, automatisch ein neues OTP per E-Mail versendet. Bei Systemen ohne E-Mail-Versand muss der Administrator ein neues OTP in der Benutzertabelle unter [Einstellungen](#) > [Benutzer](#) in der Spalte [OTP](#) einsehen und dem Benutzer mitteilen.

3. Jetzt öffnet der neue Benutzer die NoyesStorage App. Für den Fall, dass noch ein anderer Benutzer angemeldet ist, kann dieser über das Menü abgemeldet werden.

4. Der neue Benutzer gibt seine E-Mail (1) und das Einmal-Passwort (2) an der NoyesStorage App ein und meldet sich mit Klick auf Login (4) an.



5. Daraufhin legt der neue Benutzer ein individuelles Passwort (1) fest, das für alle zukünftigen Anmeldungen verwendet wird. Bestätigen Sie das Passwort (2) und speichern Sie die Eingabe (3).

Welcome to
NoyesStorage



Set Password

Please enter your new password. Once completed, your password will be successfully reset.

New password

1

Confirm password

2

Set Password

3

Back to Login

4

6. Der Login ist abgeschlossen und das Benutzerkonto ist nun freigeschaltet.

PASSWORT VERGESSEN

1. Öffnen Sie die NoyesStorage App. Für den Fall, dass noch ein anderer Benutzer angemeldet ist, kann dieser über [das Menü](#) abgemeldet werden.
2. Klicken Sie auf [Passwort vergessen](#) (3).

Welcome to NoyesStorage



Login

Email

support@noyes-tech.com



1

Password



2

[Forgot password?](#)

3

Login

4

3. Geben Sie Ihre im System hinterlegte E-Mail-Adresse an und bestätigen Sie über [Send One-Time Password](#) (2), dass Sie Ihr Passwort zurücksetzen möchten.

**Welcome to
NoyesStorage**



Forgot Password

To reset your password, please enter your email address. You will receive a one-time password (OTP) to complete the reset process.

Email

support@noyes-tech.com 

1

Send One-time Password

2

Back to Login

3

4. Prüfen Sie Ihre E-Mail (inklusive Spam-Ordner) und entnehmen Sie das Einmal-Passwort (OTP).

Anfrage zum Zurücksetzen des Passworts



Noyes Technologies (noreply) <noreply@noyes-tech.com>

Today at 15:11

To: Christoph Wagner



Anfrage zum Zurücksetzen des Passworts

Wir haben eine Anfrage zum Zurücksetzen Ihres Passworts erhalten.

Hier ist Ihr Einmalpasswort (One-Time-Password, OTP), um Ihr Passwort zurückzusetzen:

734108

Bitte verwenden Sie dieses Einmalpasswort, um sich anzumelden und Ihr neues Passwort festzulegen.

Dieses OTP läuft in 48 Stunden ab

Wenn Sie nicht wissen, warum Sie diese E-Mail erhalten haben, können Sie diese Nachricht ignorieren.

© 2025 Noyes Robotics GmbH. Alle Rechte vorbehalten.
Leopoldstr. 37 A, 80802 München
Amtsgericht München: HRB 290208
USt-IdNr.: DE367535120
Geschäftsführer: Marco Prüglmeier, Désirée Walter

5. Melden Sie sich mit E-Mail (1) und dem Einmal-Passwort (2) an der NoyesStorage App über [Login](#) (4) an.

Welcome to NoyesStorage



Login

Email

support@noyes-tech.com



1

Password



2

[Forgot password?](#)

3

Login

4

6. Legen Sie ein neues Passwort (1) fest, bestätigen Sie es (2) und speichern Sie die Eingabe (3).

Welcome to NoyesStorage



Set Password

Please enter your new password. Once completed, your password will be successfully reset.

New password

1

Confirm password

2

Set Password

3

Back to Login

4

7. Der Login ist abgeschlossen und Ihr individuelles Passwort ist gespeichert.

4.1.3 Kommissionierung

Ware kommissionieren

Das Ziel dieses Prozesses ist es, einen Request mithilfe der App abzuarbeiten. Der Request wurde entweder über [die App am NoyesStorage](#) oder über die [Programmierschnittstelle](#) erstellt.

Dieser Prozess ist im Menüpunkt **Request-Dashboard** in der App zu finden.

Personal: Picker oder Store Manager

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum **Request-Dashboard**.
3. In der Übersicht sehen Sie alle zu erledigenden Aufgaben (auch Jobs genannt), die entweder in Bearbeitung (1) oder bereit zur Kommissionierung (2) sind.

The screenshot shows the NoyesStorage Request Dashboard. At the top, there is a header with the Noyes logo, a green 'In Betrieb' status indicator, a red 'System pausieren' button, and a language dropdown set to German. Below the header, the main area is titled 'Aufgaben' and contains a description: 'Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.' To the right of this description is a 'Gruppieren nach:' dropdown menu set to 'Requests' (marked with a yellow circle 7) and a filter icon (marked with a yellow circle 6). The dashboard is divided into two main sections: 'In Bearbeitung' (marked with a yellow circle 1) and 'Bereit' (marked with a yellow circle 2). Under 'In Bearbeitung', there are two job cards. The first card (marked with a yellow circle 3) is for a 'Fulfillment' request with ID FO-0205-16:57-cf154f82-a23f-43ac-a54e-addd65084a67 and IDX 9c4f8e99-e33b-40ad-b080-0eb56901bd88. It shows a 'Picking' task for 'MAR-03 MERCH - Coffee ...' with 2 items and location 69. The second card (marked with a yellow circle 5) is for a 'Picking' task for 'MAR-03 MERCH - Coffee ...' with 1 item and location 70. Under 'Bereit', there are two job cards. The first card (marked with a yellow circle 4) is for a 'Picking' request with ID FO-0205-16:57-c... and IDX 141e801a-51c3-47b3-b6b4-d8342dc9da8b. It shows a 'Picking' task for 'MAR-28 MERCH - Pen bla...' with 3 items and location 69. The second card (marked with a yellow circle 4) is for a 'Picking' request with ID FO-0205-16:57-7... and IDX 141e801a-51c3-47b3-b6b4-d8342dc9da8b. It shows a 'Picking' task for 'MAR-04 MERCH - Water b...' with 1 item and location 70. Each card also displays 'Balkon' and 'Carrier' information and a red upward arrow icon.

4. Die Jobs werden nach Requests gruppiert (3). Jeder einzelne Job wird als Karte angezeigt, die alle relevanten Informationen zum Auftrag enthält, zum Beispiel die Auftragsnummer, die Anzahl der Artikel sowie die Ablageposition.
 5. Bereits bearbeitbare Jobs erkennen Sie an den Karten im Bereich **Bereit** (4). Jobs im Bereich **In Bearbeitung** (5) sind noch nicht zur Interaktion verfügbar.
 6. Um nur Jobs für einen bestimmten Balkon anzuzeigen, passen Sie den Filter (6) entsprechend an.
 7. Je nach Anwendungsfall können Sie einzelne Requests nacheinander oder mehrere Requests gruppiert nach SKUs gleichzeitig abarbeiten. Dies lässt sich über die Gruppierung (7) steuern:
8. **Single-Order Kommissionierung (Requests):**

NOYES TECHNOLOGIES In Betrieb System pausieren 🇩🇪

Aufgaben

Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.

Gruppieren nach: Requests Y

In Bearbeitung 1

Fulfillment ID: FO-0205-16:57-cf154f82-a23f-43ac-a54e-addr65084a67
IDX: 9c4f8e99-e33b-40ad-b080-0eb56901bd88 3

Picking Request: FO-0205-16:57-c...
05.02.2025, 16:57 Balkon: 3
Carrier: 422
Box: 1036

MAR-03
MERCH - Coffee ... 5

📦 2 Stk. 📍 69 ⬆️

Fulfillment ID: FO-0205-16:57-7ccf890b-075c-4017-95c8-f751ec15eb90
IDX: 141e801a-51c3-47b3-b6b4-d8342dc9da8b

Picking Request: FO-0205-16:57-7...
05.02.2025, 16:57 Balkon: 3
Carrier: 422
Box: 1036

MAR-03
MERCH - Coffee ...

📦 1 Stk. 📍 70 ⬆️

Bereit 2

Picking Request: FO-0205-16:57-c...
05.02.2025, 16:58 Balkon: 3
Carrier: 417
Box: 1015

MAR-28
MERCH - Pen bla... 4

📦 3 Stk. 📍 69 ⬆️

Picking Request: FO-0205-16:57-7...
05.02.2025, 16:58 Balkon: 2
Carrier: 384
Box: 1037

MAR-04
MERCH - Water b... 6

📦 1 Stk. 📍 70 ⬆️

- a. Die Jobs werden nach Requests gruppiert (3) und untereinander angezeigt.
- b. Bereits bearbeitbare Jobs erkennen Sie an den Karten im Bereich **Bereit** (4).
- c. Jobs im Bereich **In Bearbeitung** (5) sind noch nicht zur Interaktion verfügbar.

The screenshot shows the NOYES Technologies dashboard. At the top, it indicates 'In Betrieb' (Operational) and has a 'System pausieren' (Pause System) button. The main section is titled 'Aufgaben' (Tasks) and contains a list of tasks. One task is highlighted in green, showing details for 'Balkon 3' (MAR-28, MERCH - Pen bla...). The detailed view on the right shows a grid of boxes representing the balcony layout, with one box highlighted in green. A 'Bestätigen' (Confirm) button is visible at the bottom of the detailed view.

Task List:

- Task 1 (Highlighted):** Fulfillment, ID: FO-0205-16:57-cf154f82-a23f-43ac-a54e-addd65084a67, IDX: 9c4f8e99-e33b-40ad-b080-0eb56901bd88. Request: FO-0205-16:57-c... (05.02.2025, 16:58). Location: MAR-28. Merch: MERCH - Pen bla... (3 Stk., 69). Carrier: 417, Box: 1015. Balkon: 3.
- Task 2:** Fulfillment, ID: FO-0205-16:57-7ccf890b-075c-4017-95c8-f751ec15eb90, IDX: 141e801a-51c3-47b3-b6b4-d8342dc9da8b. Request: FO-0205-16:57-7... (05.02.2025, 16:58). Location: MAR-04. Merch: MERCH - Water b... (1 Stk., 70). Carrier: 384, Box: 1037. Balkon: 2.

Job Details (Balkon 3): MAR-28, MERCH - Pen bla... (3 Stk., 69). Buttons: Bestätigen (3), Minimize (4), Close (5).

d. Durch Klicken auf eine bereite Job-Karte öffnen sich die Job-Details (1).

e. Die Job-Details liefern eine Übersicht über die zu erledigende Tätigkeit. Für den Fall einer Kommissionierung (Fulfillment) wird angezeigt, welche Anzahl von welcher SKU aus welchem Balkon und welcher Box entnommen werden muss und wo diese platziert werden sollen.

f. Nach erfolgreicher Entnahme klicken Sie auf **Bestätigen** (3), um Ihre Aktion abzuschließen und den Carrier auf dem Balkon zurück ins System zu senden.

g. Das System wählt automatisch die nächstbeste Job-Karte aus, um einen reibungslosen Ablauf zu gewährleisten.

h. Klicken Sie auf (4), um die Details zu minimieren, oder auf (5), um den Job abzubrechen.

9. Multi-Order Kommissionierung (SKUs):

NOYES TECHNOLOGIES In Betrieb System pausieren 🇩🇪

Aufgaben

Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.

Gruppieren nach: SKUs ⌵

In Bearbeitung

▼ MAR-03, MERCH - Coffee cups ... 1

▼ MAR-04, MERCH - Water bottle

Bereit

Picking Request: FO-0205-16:57-c...
05.02.2025, 18:46

MAR-03
MERCH - Coffee ...

📦 2 Stk. 📍 69

Balkon: 3
Carrier: 422
Box: 1036

2

⬆️

Picking Request: FO-0205-16:57-7...
05.02.2025, 18:46

MAR-03
MERCH - Coffee ...

📦 1 Stk. 📍 70

Balkon: 3
Carrier: 422
Box: 1036

3

⬆️

Picking Request: FO-0205-16:57-7...
05.02.2025, 16:58

MAR-04
MERCH - Water b...

Balkon: 2
Carrier: 384
Box: 1037

a. Die Jobs werden nach SKUs gruppiert (1) und untereinander angezeigt.

b. Dadurch kann dieselbe SKU direkt für mehrere Requests kommissioniert werden, zum Beispiel über die Job-Karten (2) und (3). Das reduziert Wege- und Pickzeiten.

The screenshot displays the Noyes Technologies dashboard. At the top, there is a status bar with 'In Betrieb' and a 'System pausieren' button. The main area is titled 'Aufgaben' and shows a list of tasks under the 'Bereit' section. A task card for 'MAR-03, MERCH - Coffee cups ...' is highlighted. To the right, the 'Job Details' view is open, showing a grid for 'Balkon 3' and a 'Bestätigen' button. Numbered callouts (1-5) indicate key UI elements: 1 points to a job card, 2 to the task details in the job details view, 3 to the 'Bestätigen' button, 4 to the minimize icon, and 5 to the close icon.

c. Durch Klicken auf eine bereite Job-Karte öffnen sich die Job-Details (1).

d. Die Job-Details liefern eine Übersicht über die zu erledigende Tätigkeit. Für den Fall einer Kommissionierung (Fulfillment) wird angezeigt, welche Anzahl von welcher SKU aus welchem Balkon und welcher Box entnommen werden muss und wo diese platziert werden sollen.

e. Nach erfolgreicher Entnahme klicken Sie auf **Bestätigen** (3), um Ihre Aktion abzuschließen und den Carrier auf dem Balkon zurück ins System zu senden.

f. Das System wählt automatisch die nächstbeste Job-Karte aus, um einen reibungslosen Ablauf zu gewährleisten.

g. Klicken Sie auf (4), um die Details zu minimieren, oder auf (5), um den Job abzubrechen.

PICKING-ALARM BEARBEITEN

HINWEIS

Diese Funktion ist standardmäßig ausgeschaltet. Bei Bedarf kann der Noyes Support sie für Ihr System aktivieren.

Ein **Picking-Alarm** ist eine normale operative Aufgabe im Kommissionierprozess. Er wird angezeigt, wenn eine angeforderte SKU oder Menge nicht im NoyesStorage verfügbar ist und deshalb außerhalb des NoyesStorage, zum Beispiel aus einem externen Lagerbereich, bearbeitet werden muss.

Beispiel: Ein Auftrag enthält zwei SKUs. Eine SKU ist im NoyesStorage verfügbar und wird als normaler Picking-Job kommissioniert. Für die fehlende SKU zeigt das System einen **Picking-Alarm** an.

The screenshot displays the 'Request Overview' dashboard. At the top, it shows the system status as 'Running' and a 'Pause System' button. The main content is divided into two sections: 'Ready' and 'Job Details'. The 'Ready' section lists three items: two 'Picking alert' items and one 'Picking' item. The 'Job Details' section shows a 'Balcony' picking alert with a photo of a worker and a 'Confirm' button.

Ein Picking-Alarm wird lokal nach den für Ihren Standort definierten Arbeitsanweisungen bearbeitet. Die Job-Karte zeigt, welche SKU und welche Menge außerhalb des NoyesStorage zu bearbeiten ist. Im Unterschied zu einem normalen Picking-Job benötigt ein Picking-Alarm keinen Carrier, keine Box, keinen Balkon und keine interne Lagerposition im NoyesStorage.

1. Öffnen Sie die **Picking-Alarm**-Karte im **Request-Dashboard**.
2. Lesen Sie in den Job-Details die SKU und Menge ab.
3. Entnehmen oder bearbeiten Sie die Ware außerhalb des NoyesStorage gemäß den lokal definierten Arbeitsanweisungen.
4. Klicken Sie auf **Bestätigen**, sobald die externe Bearbeitung abgeschlossen ist.

Wenn die Ware auch außerhalb des NoyesStorage nicht verfügbar ist, folgen Sie den lokal definierten Arbeitsanweisungen für nicht erfüllbare Positionen.

Normale Picking-Jobs für verfügbare Ware laufen weiterhin wie gewohnt. Der Auftrag kann abgeschlossen werden, sobald alle normalen Picking-Jobs und alle Picking-Alarme bestätigt wurden.

Auftrag abschließen

Sobald alle SKUs für einen Auftrag entnommen oder eingelagert wurden, weist das System auf die Fertigstellung des Auftrags hin, fasst diesen nochmals zusammen und wartet auf Bestätigung. Hierdurch lässt sich der NoyesStorage optimal in Ihren Gesamtprozess einbinden.

HINWEIS

Die Bestätigung des Auftrags ist optional und standardmäßig ausgeschaltet. Über den Noyes Support kann diese Einstellung geändert werden.

Personal: Picker oder Store Manager

1. Sie sind in der NoyesStorage App angemeldet und haben alle Artikel des Auftrags kommissioniert.
2. Der NoyesStorage zeigt die Fertigstellung des Auftrags als **Packing**-Karte an. Durch Klicken auf die Karte öffnen sich die Details.

The screenshot displays the NoyesStorage app interface. At the top, there is a header with the Noyes logo, a status indicator 'In Betrieb', and a 'System pausieren' button. The main content area is titled 'Aufgaben' and shows a list of tasks under the 'Bereit' (Ready) status. A task card is highlighted with a green border and a red arrow. The task card shows a 'Packing' status, a request ID 'FO-0206-08:00-7...', and a location '69'. To the right, the 'Job Details' view is open, showing a summary of the request and a list of units to be packed. The 'Bestätigen' button is located at the bottom right of the details view.

Job Details
Request / FO-0206-08:00-7... / Job / e503c3ad-49c3-4...

Zusammenfassung
Alle erforderlichen Jobs zur Erfüllung des Request FO-0206-08:00-7... wurden erfolgreich abgeschlossen und können jetzt verpackt werden.
Die folgenden Einheiten wurden abgerufen.

MAR-28, MERCH - Pen bla...	3x
MAR-03, MERCH - Coffee ...	2x

Bestätigen

3. In den Details (1) sehen Sie eine Übersicht über den Auftrag, die zugehörigen Artikel und die Ablageposition (2).
4. Um die Fertigstellung des Auftrags zu bestätigen, klicken Sie auf die **Bestätigen**-Schaltfläche (3).

4.1.4 System pausieren

Pausieren und Fortsetzen des NoyesStorage

Ziel dieses Prozesses ist es, den Betrieb des NoyesStorage sofort zu unterbrechen – z. B. für Hardware-Reparaturen, Reinigungen, Änderungen oder das Onboarding von NoyesBots – und anschließend den Betrieb wieder aufzunehmen.

VORSICHT

Risiko des Entladens der NoyesBots

Bei einer Systempausierung besteht das Risiko, dass NoyesBots einen kritischen Ladezustand erreichen, da sie in diesem Zustand nicht automatisiert zur Ladestation geschickt werden können. - Systempausierung nur bei dringendem Stillstand verwenden. - Eine Systempausierung schnellstmöglich, spätestens aber nach 60 Minuten wieder aufheben.

In der NoyesStorage App wird stets ein **System pausieren**-Button als Teil der Statusleiste angezeigt. So pausieren Sie den NoyesStorage:

1. Melden Sie sich in der App an.
2. Klicken Sie auf die Schaltfläche **System pausieren**.
3. Das System zeigt eine Popup-Meldung, ob Sie das System wirklich pausieren möchten, mit den Optionen:
4. **NEIN:** Die Popup-Meldung verschwindet und der Betrieb wird ohne Unterbrechung fortgesetzt.
5. **JA:** Die Popup-Meldung verschwindet und der Betrieb wird unterbrochen.
6. Um den Betrieb wieder aufzunehmen, navigieren Sie zum Bereich **Systemsteuerung**.
7. Klicken Sie auf **Betrieb fortsetzen** – der NoyesStorage nimmt dann seinen Betrieb wieder auf.

4.1.5 Request erstellen

AUFTRAGSERSTELLUNG AUF DER NOYESSTORAGE APP

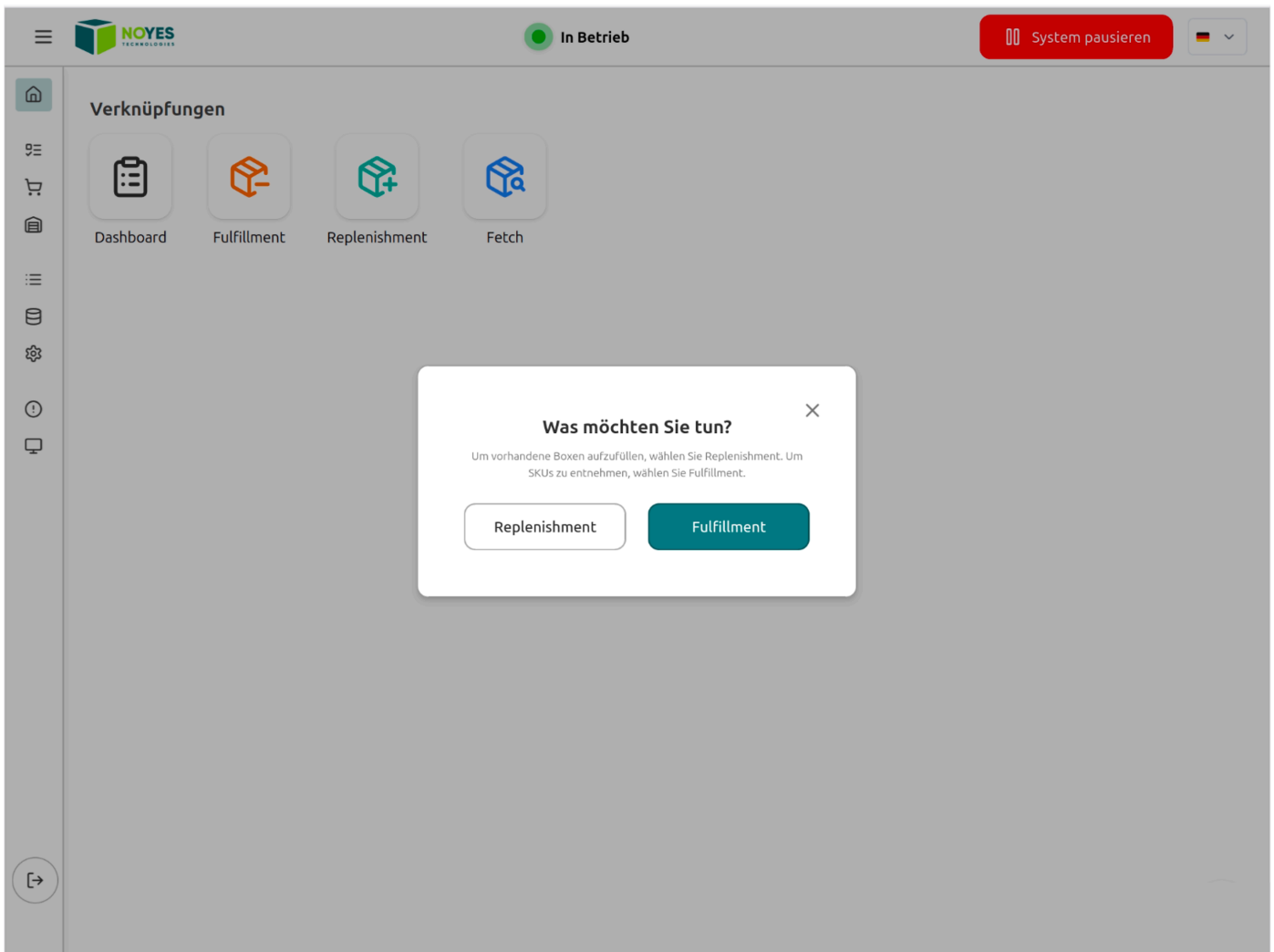
Das Ziel dieses Prozesses ist es, über die App eine Bestellung zu erstellen, um SKUs ein- oder auszulagern.

Dieser Prozess ist im Menüpunkt **Request erstellen** zu finden.

Personal: Store Manager

Ein- und Auslagerung von SKUs

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum Abschnitt **Request erstellen** im Seitenmenü.
3. Das System zeigt eine Popup-Meldung, ob Sie SKUs ein- oder auslagern möchten. Wählen Sie für das Auslagern **Fulfillment** und für das Einlagern **Replenishment**.

















4. Die Übersicht zeigt alle verfügbaren SKUs (1).

NOYES TECHNOLOGIES In Betrieb System pausieren 🇩🇪

Fulfillment Request erstellen

Wählen Sie Produkte und deren Menge aus, um eine Fulfillment Request zu erstellen.

Suchen... Checkout

 H000016 [A], Noyes Storage -... 6200 on stock, 300 replenishable	3 - +	 H001093 [A], ARCHIVED - Hexa... 3985 on stock, 28015 replenishable	0 - +
 H000017 [A], Noyes Storage -... 3500 on stock, 1700 replenishable	0 - +	 INT-14, Screws Diverse ... 3200 on stock, 28800 replenishable	2 - +
 H000072 [A], ARCHIVED - Scre... 2073 on stock, 47927 replenishable	0 - +	 H001166 [A], Noyes Storage -... 1500 on stock, 0 replenishable	0 - +
 H000089 [A], ARCHIVED - Scre... 700 on stock, 300 replenishable	0 - +	 H000026 [A], Noyes Storage -... 600 on stock, 400 replenishable	0 - +
 NBC037 [A], Toolbox - ISO14... 400 on stock, 600 replenishable	0 - +	 H000014 [C], Noyes Storage -... 380 on stock, 70 replenishable	0 - +
 MAR-03, MERCH - Coffee ... 298 on stock, 202 replenishable	0 - +	 H000014 [A], Noyes Storage -... 280 on stock, 20 replenishable	0 - +
 H001035 [A], ARCHIVED - Char... 200 on stock, 0 replenishable	0 - +	 H000136 [D], Noyes Storage -... 167 on stock, 4233 replenishable	0 - +

Angezeigte Elemente: 100 / 1541, Ausgewählt: 2

5. Suchen Sie nach dem/den gewünschten Artikel(n) über das Suchfeld (2) und klicken Sie auf die entsprechende SKU. Die Karte wird grün markiert.
6. Passen Sie die Menge zum Ein- oder Auslagern durch Eingabe im Textfeld oder durch Klicken auf Plus- und Minus-Schaltflächen (3) an.
7. Klicken Sie auf den Checkout (4), um eine Übersicht der ausgewählten Artikel und Mengen zu erhalten und den Request final zu erstellen.
8. Überprüfen (und passen Sie ggf. an) Sie in dieser Übersicht (1) die Artikel und Mengen, bevor Sie den Request absenden (2).

NOYES TECHNOLOGIES

In Betrieb

System pausieren

Fulfillment Request erstellen

Wählen Sie Produkte und deren Menge aus, um eine Fulfillment Request zu erstellen.

Suchen...

Checkout

Ihr Fulfillment Request

Sind Sie sicher, dass Sie einen Request für die folgenden SKUs erstellen möchten?

- H000016 [A], Noyes Storage -...
6200 on stock, 300 replenishable **1** 3
- INT-14, Screws Diverse ...
3200 on stock, 28800 replenishable 2

Jetzt anfordern **2**

Angezeigte Elemente: 100 / 1541, Ausgewählt: 2

Bestimmten Ladungsträger hervorheben

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum Menüpunkt **Inventar**.

NOYES TECHNOLOGIES In Betrieb System pausieren DE

Inventar

Diese Tabelle bietet eine Echtzeitübersicht über Boxen, Träger und Lagerbestände im NoyesStorage. Sie können Träger für Produkte identifizieren, die nachgefüllt werden müssen, oder solche mit Boxen ohne zugewiesene Ladungen und einen Request erstellen, um diese Träger abzurufen.

Suchen... Alle Filter löschen Carrier anfordern

<input type="checkbox"/>	Carrier ID	Box ID	SKU ID	SKU Name	Menge	Menge (max)	Level	Boxengröße
<input type="checkbox"/>	314	783	H000079 [D]	Noyes Storage - Cover bottom short	15	100	1	Large
<input type="checkbox"/>	318	953	OPS-19	Installation - Bosch Boxen small	2	2	1	Small
<input checked="" type="checkbox"/>	318	951	INT-17	Nail Iron (Nageleisen)	1	1	1	Small
<input type="checkbox"/>	318	952	INT-18	Fahrradschloss	1	1	1	Small
<input type="checkbox"/>	318	950	H000026 [A]	Noyes Storage - Screw DIN 34805-2 M6x25-10.9-A2K-TX30 (Side Cover)	600	1000	1	Small
<input checked="" type="checkbox"/>	320	772	H001103 [D]	Noyes Storage - Floor plate inlay 2 (Floorplates)	0	100	1	Medium
<input type="checkbox"/>	320	771	H001104 [C]	ARCHIVED - Inlay 3 (Floorplates)	0	100	1	Medium
<input type="checkbox"/>	321	758	H000016 [A]	Noyes Storage - Srew M6x25 (Floorplates)	6200	6500	1	Large

Angezeigte Elemente: 49 / 146 - Ausgewählt: 2 Zeile(n)

3. In der Übersicht werden alle Ladungsträger und deren Inhalt angezeigt.
4. Suchen Sie den gewünschten Ladungsträger über das Suchfeld (1) oder setzen Sie Filter in der Tabellenkopfzeile (2).
5. Wählen Sie einen oder mehrere Ladungsträger durch Klicken (3) aus.
6. Drücken Sie die Schaltfläche **Carrier anfordern** (4), um die ausgewählten Ladungsträger hervorzuholen.

4.1.6 Requesthistorie

AUFTRÄGE EINSEHEN

Das Ziel dieses Prozesses ist es, Informationen über vergangene und offene Requests zu erhalten. Die Tabelle listet alle bekannten Requests auf. Über die Kopfzeile (1) können Sie den Inhalt der Tabelle filtern und sortieren.

Dieser Prozess befindet sich im Menüpunkt **Requests & Jobs**.

Personal: Store Manager

Request & Jobs

Diese Tabelle bietet einen Echtzeit-Überblick über alle Requests und Jobs. Sie können den Fortschritt laufender Jobs verfolgen, abgeschlossene Requests überprüfen und nach bestimmten Kriterien filtern, um schnell relevante Informationen zu finden.

Request: FULFILLMENT, REPLENISHMENT, FETCH × Alle Filter löschen ▾

Request	ID	Job	SKU ID	Carrier ID	Level	Status	↕
Fulfillment	8a21f22e-248b-477e-b34a-5207899a10be	Picking	H001103 [D]	2264	8	Succeeded	
Fetch	d363e758-5153-4530-8aa7-975f5f86c5d7	Storing	-	2208	6	Succeeded	
Fetch	b184c43e-f55d-4928-a51d-4e6ca3d19180	Storing	-	2266	8	Succeeded	
Fulfillment	8a21f22e-248b-477e-b34a-5207899a10be	Buffering	-	2264	8	Succeeded	
Fulfillment	8a21f22e-248b-477e-b34a-5207899a10be	Storing	-	2264	8	Succeeded	
Fetch	955532e7-1bdf-4c20-9716-2612b04bfccb	Buffering	-	2205	6	Succeeded	
Fetch	955532e7-1bdf-4c20-9716-2612b04bfccb	Retrieving	-	2205	6	Succeeded	
Fetch	955532e7-1bdf-4c20-9716-2612b04bfccb	Checking	-	2205	6	Succeeded	

Angezeigte Elemente: 100 / 80344


4.1.7 Inventar

LAGERBESTAND EINSEHEN

Über den Menüpunkt **Inventar** können Sie den Lagerbestand einsehen. Die Tabelle zeigt alle im System befindlichen Boxen, deren Zuordnung zum Ladungsträger sowie den in der Box gelagerten Artikel.

Zur Auffindung der richtigen Ware stehen ein Suchfeld sowie Filter- und Sortierfunktionen zur Verfügung.


Personal: Store Manager

 ● **In Betrieb** System pausieren 🇩🇪 ▼

Inventar

Diese Tabelle bietet eine Echtzeitübersicht über Boxen, Träger und Lagerbestände im NoyesStorage. Sie können Träger für Produkte identifizieren, die nachgefüllt werden müssen, oder solche mit Boxen ohne zugewiesene Ladungen und einen Request erstellen, um diese Träger abzurufen.

Alle Filter löschen ▼ Carrier anfordern

<input type="checkbox"/>	Carrier ID ▼	Box ID ▼	SKU ID ▼	SKU Name ▼	Menge ▼	Menge (max) ▼	Level ▼	Boxengröße ▼	
<input type="checkbox"/>	314	783	H000079 [D]	Noyes Storage - Cover bottom short	15	100	1	Large	
<input type="checkbox"/>	318	953	OPS-19	Installation - Bosch Boxen small	2	2	1	Small	
<input checked="" type="checkbox"/>	318	951	INT-17	Nail Iron (Nageleisen)	1	1	1	Small	
<input type="checkbox"/>	318	952	INT-18	Fahrradschloss	1	1	1	Small	
<input type="checkbox"/>	318	950	H000026 [A]	Noyes Storage - Screw DIN 34805-2 M6x25-10.9-A2K-TX30 (Side Cover)	600	1000	1	Small	
<input checked="" type="checkbox"/>	320	772	H001103 [D]	Noyes Storage - Floor plate inlay 2 (Floorplates)	0	100	1	Medium	
<input type="checkbox"/>	320	771	H001104 [C]	ARCHIVED - Inlay 3 (Floorplates)	0	100	1	Medium	
<input type="checkbox"/>	321	758	H000016 [A]	Noyes Storage - Srew M6x25 (Floorplates)	6200	6500	1	Large	

Angezeigte Elemente: 49 / 146 - Ausgewählt: 2 Zeile(n)

4.1.8 Management von SKUs

MANAGEMENT VON STOCK KEEPING UNIT (SKU) STAMMDATEN

Eine SKU (Stock Keeping Unit) ist eine eindeutige Kennung zur Identifizierung und Verwaltung von Produktvarianten im NoyesStorage. SKU-Stammdaten helfen, zu unterscheiden, welche Varianten im Lager verfügbar sind.

Zu den SKU-Stammdaten gehören der eindeutige Identifikator (ID) und der Name. Benutzer können die SKU-Stammdaten über die App einfach hinzufügen. Einmal angelegt, stehen diese Daten für weitere Prozessschritte zur Verfügung.

Personal: Store Manager

Anzeigen von SKU-Stammdaten

Zugriff auf die Stammdaten erfolgt über den Reiter **SKU-Stammdaten**.

The screenshot shows the 'SKU Stammdaten' screen in the NoyesStorage app. The interface includes a top navigation bar with the Noyes logo, a status indicator 'In Betrieb', and a 'System pausieren' button. A sidebar on the left contains navigation icons. The main content area features a search bar (1) and a table (2) of SKUs. The table has columns for ID, Name, Einheit, Bei Leerstand, and Aktualisiert. A '+ Hinzufügen' button (4) is located above the table. A yellow circle (3) highlights the search bar. The table contains 12 rows of SKU data.


ID	Name	Einheit	Bei Leerstand	Aktualisiert
100000000001	Hagebuttentee	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:37
1000000000585	Professorentee, 500g	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:38
100000100000	Professorentee, 250g	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:39
100000200000	NoyesBot, Gen 7	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:41
100000300000	NoyesBot, Gen7.1	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:40
100000400000	NoyesBot, Gen 8	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:41
100000500000	Weihrauch	Gramm (g)	Nichts tun	13 Feb 2025 15:44
100000600000	Basilikumblätter	Gramm (g)	Nichts tun	13 Feb 2025 15:44
100000700000	Brennnesselsamen	Gramm (g)	Nichts tun	13 Feb 2025 15:44
100000900000	Hagebuttentee	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:44
100001100000	Elevator	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:45

1. Navigieren Sie zum Menü und wählen Sie den Reiter **SKU-Stammdaten**.
2. Eine Tabelle zeigt alle bekannten SKU-Stammdaten und deren Attribute.
3. Eine Suchleiste (1) sowie Filter- und Sortierfunktionen in der Tabellenkopfzeile (2) unterstützen Sie beim Finden bestimmter SKUs.

Erstellen von SKU-Stammdaten

In der Ansicht SKU-Stammdaten können Sie neue SKUs anlegen, die anschließend einer Box zugewiesen werden.

SKU hinzufügen

 ● In Betrieb System pausieren 🇩🇪

SKU Stammdaten

Eine Stock Keeping Unit (SKU) ist eine Produktvariante in Ihrem Inventar. Sie kann benutzt werden um Lagerbestände über Ein- und Auslagerungen zu verfolgen. Hier können SKU Stammdaten verwaltet werden. Sie können SKUs erstellen, aktualisieren und löschen. Nutzen Sie Filter, um bestimmte SKUs schnell zu finden.

Alle Filter löschen

+ Hinzufügen

Bearbeiten

Löschen

>

<input type="checkbox"/>	ID	Name	Einheit	Bei Leerstand	Aktualisiert	
<input type="checkbox"/>	100000000001	Hagebuttentee	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:37	
<input type="checkbox"/>	1000000000585	Professorentee, 500g	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:38	
<input type="checkbox"/>	100000100000	Professorentee, 250g	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:39	
<input type="checkbox"/>	100000200000	NoyesBot, Gen 7	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:41	
<input type="checkbox"/>	100000300000	NoyesBot, Gen7.1	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:40	
<input type="checkbox"/>	100000400000	NoyesBot, Gen 8	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:41	
<input type="checkbox"/>	100000500000	Weihrauch	Gramm (g)	Nichts tun	13 Feb 2025 15:44	
<input type="checkbox"/>	100000600000	Basilikumblätter	Gramm (g)	Nichts tun	13 Feb 2025 15:44	
<input type="checkbox"/>	100000700000	Brennnesselsamen	Gramm (g)	Nichts tun	13 Feb 2025 15:44	
<input type="checkbox"/>	100000900000	Hagebuttentee	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:44	
<input type="checkbox"/>	100001100000	Elevator	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:45	

Angezeigte Elemente: 50 / 1525 - Ausgewählt: 0 Zeile(n)

Tippen Sie auf die Schaltfläche **Hinzufügen** (4), um die Eingabemaske zu öffnen.

Daten eingeben

Tragen Sie in der Eingabemaske (1) die erforderlichen Informationen in die entsprechenden Felder ein:

- **ID (2):** Eindeutige Identifikation der SKU. *Hinweis: Eine spätere Änderung der ID ist nicht möglich.*
- **Name (3):** Bezeichnung der SKU. *Änderungen sind später möglich.*
- **Maßeinheit (4):** Einheit, in der diese SKU gelagert und entnommen wird. *Änderungen sind später möglich.*
- **Verhalten bei Leerstand (5):** Legt fest, wie verfahren wird, sobald die letzte Ware aus einer Box entnommen wurde.

Vorgang abschließen

Bestätigen Sie die Eingabe, indem Sie auf **SKU hinzufügen** (6) tippen.

Ändern von SKU-Stammdaten

In der Ansicht **SKU-Stammdaten** können Sie bestehende Stammdaten bearbeiten.

HINWEIS

Da die ID der SKU als eindeutiges Identifizierungsmerkmal dient, kann sie nach der Erstellung nicht mehr geändert werden. Möchten Sie eine andere ID verwenden, löschen Sie die betreffende SKU und legen Sie eine neue mit der gewünschten ID an.

SKU auswählen

Klicken Sie auf die Zeile (1) der gewünschten SKU, um diese auszuwählen. Die Zeile wird hervorgehoben.

NOYES In Betrieb System pausieren

SKU Stammdaten

Eine Stock Keeping Unit (SKU) ist eine Produktvariante in Ihrem Inventar. Sie kann benutzt werden um Lagerbestände über Ein- und Auslagerungen zu verfolgen. Hier können SKU Stammdaten verwaltet werden. Sie können SKUs erstellen, aktualisieren und löschen. Nutzen Sie Filter, um bestimmte SKUs schnell zu finden.

Suchen... Alle Filter löschen + Hinzufügen Bearbeiten Löschen

ID	Name	Einheit	Bei Leerstand	Aktualisiert
100000000001	Hagebuttentee	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:37
1000000000585	Professorentee, 500g	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:38
100000100000	Professorentee, 250g	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:39
100000200000	NoyesBot, Gen 7	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:41
100000300000	NoyesBot, Gen7.1	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:40
100000400000	NoyesBot, Gen 8	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:41
100000500000	Weihrauch	Gramm (g)	Nichts tun	13 Feb 2025 15:44
100000600000	Basilikumblätter	Gramm (g)	Nichts tun	13 Feb 2025 15:44
100000700000	Brennesselsamen	Gramm (g)	Nichts tun	13 Feb 2025 15:44
100000900000	Hagebuttentee	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:44
100001100000	Elevator	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:45

Angezeigte Elemente: 50 / 1525 - Ausgewählt: 1 Zeile(n)

Bearbeiten öffnen

Tippen Sie anschließend auf die Schaltfläche **Bearbeiten** (2), um die Details anzuzeigen und zu bearbeiten.

Daten anpassen und speichern

NOYES TECHNOLOGIES

In Betrieb

System pausieren

SKU Stammdaten

Eine Stock Keeping Unit (SKU) ist eine Produkt- und Auslagerungen zu verfolgen. Hier können Sie die Stammdaten bearbeiten. Nutzen Sie Filter, um bestimmte SKUs zu finden.

Suchen...

ID	Aktualisiert
100000000001	13 Feb 2025 15:37
1000000000585	13 Feb 2025 15:38
100000100000	13 Feb 2025 15:39
100000200000	13 Feb 2025 15:41
100000300000	13 Feb 2025 15:40
100000400000	13 Feb 2025 15:41
100000500000	13 Feb 2025 15:44
100000600000	13 Feb 2025 15:44
100000700000	13 Feb 2025 15:44
100000900000	13 Feb 2025 15:44
100001100000	13 Feb 2025 15:45

Bestandsverwaltungseinheit (SKU) bearbeiten

Bearbeiten Sie die Felder einer bestehenden SKU aus den NoyesStorage-Masterdaten.

ID: 100000500000

Name: Weihrauch

Maßeinheit: **Gramm (g)**

Speichern

Angezeigte Elemente: 50 / 1525 - Ausgewählt: 1 Zeile(n)

In der Übersicht (1) können Sie die gespeicherten Stammdaten ändern. Bestätigen Sie Ihre Änderungen durch Tippen auf **Speichern** (2).

Löschen von SKU-Stammdaten

In der Ansicht **SKU-Stammdaten** können Sie SKU-Stammdaten löschen.

HINWEIS

Eine SKU kann erst gelöscht werden, wenn alle zugehörigen Beladungen aus dem System entfernt wurden.

SKU auswählen

NOYES In Betrieb System pausieren 🇩🇪

SKU Stammdaten

Eine Stock Keeping Unit (SKU) ist eine Produktvariante in Ihrem Inventar. Sie kann benutzt werden um Lagerbestände über Ein- und Auslagerungen zu verfolgen. Hier können SKU Stammdaten verwaltet werden. Sie können SKUs erstellen, aktualisieren und löschen. Nutzen Sie Filter, um bestimmte SKUs schnell zu finden.

Suchen... Alle Filter löschen + Hinzufügen Bearbeiten Löschen

ID	Name	Einheit	Bei Leerstand	Aktualisiert
<input type="checkbox"/> 100000000001	Hagebuttentee	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:37
<input type="checkbox"/> 1000000000585	Professorentee, 500g	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:38
<input type="checkbox"/> 100000100000	Professorentee, 250g	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:39
<input type="checkbox"/> 100000200000	NoyesBot, Gen 7	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:41
<input type="checkbox"/> 100000300000	NoyesBot, Gen7.1	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:40
<input type="checkbox"/> 100000400000	NoyesBot, Gen 8	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:41
<input checked="" type="checkbox"/> 100000500000	Weihrauch	Gramm (g)	Nichts tun	13 Feb 2025 15:44
<input type="checkbox"/> 100000600000	Basilikumblätter	Gramm (g)	Nichts tun	13 Feb 2025 15:44
<input type="checkbox"/> 100000700000	Brennnesselsamen	Gramm (g)	Nichts tun	13 Feb 2025 15:44
<input type="checkbox"/> 100000900000	Hagebuttentee	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:44
<input type="checkbox"/> 100001100000	Elevator	Stück (Stk)	Zuweisung aufheben	13 Feb 2025 15:45

Angezeigte Elemente: 50 / 1525 - Ausgewählt: 1 Zeile(n)

Wählen Sie die gewünschte SKU aus, indem Sie auf die entsprechende Zeile (1) klicken. Die Zeile wird hervorgehoben.

Löschen initiieren

Tippen Sie auf die Schaltfläche Löschen (3), um den Löschvorgang einzuleiten.

Löschvorgang bestätigen

Bestätigen Sie den Löschvorgang in der daraufhin angezeigten Abfrage.

4.1.9 Management von Boxen

MANAGEMENT VON BOXEN

Um Ware in Boxen anzulegen, muss zunächst die Boxgröße, die auf einem Ladungsträger verstaute ist, hinterlegt werden. Boxen können angepasst werden, wenn der entsprechende Ladungsträger am Balkon steht und das System auf Nutzerinput wartet (z. B. während eines Kommissioniervorgangs).

Dieser Abschnitt beschreibt das Erstellen, Anpassen und Löschen von Boxen auf einem Ladungsträger.

Der NoyesStorage unterscheidet folgende 4 Boxgrößen: - **Big**: Box mit 600×400 mm Grundfläche, füllt den gesamten Ladungsträger aus. - **Medium**: Box mit 300×400 mm Grundfläche, füllt den halben Ladungsträger aus. - **Small**: Box mit 300×200 mm Grundfläche, füllt einen Viertel-Ladungsträger aus. - **Mini**: Box mit 150×200 mm Grundfläche, füllt einen Achtel-Ladungsträger aus.

Erstellen von Boxen auf einem Ladungsträger

1. Bestellen Sie einen Ladungsträger zum Balkon

The screenshot shows the NoyesStorage interface. At the top, there is a status bar with the Noyes logo, a green 'In Betrieb' indicator, a red 'System pausieren' button, and a language dropdown set to German. The main content area is titled 'Aufgaben' and contains a 'Bereit' section with a 'Fetch' button and a 'Checking' button. A job card is visible with details: 'Request: CO-0206-11:46-d...', 'Balkon: 2', and 'Carrier: 366'. To the right, the 'Job Details' section for 'Balkon 2' is shown, featuring a diagram of the loading platform with four numbered yellow circles (1, 2, 3, 4) indicating specific areas. A 'Bestätigen' button is located at the bottom of the job details panel.

2. Öffnen Sie durch Klicken auf die Job-Karte die Übersicht der Job-Details. Diese zeigt:

- einen freien Stellplatz (Boxgröße Medium) (2)
- eine leere Box (3)
- eine Box mit Beladung (4)

3. Passen Sie den Ladungsträger und dessen Boxen/Beladungen durch Klicken auf die Bearbeiten-Schaltfläche (1) an.

4. **Neue Seite:**

NOYES TECHNOLOGIES ● In Betrieb System pausieren 🇩🇪

Aufgaben

Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.

Gruppieren nach: Requests ⌵

Bereit

Fetch ID: CO-0206-11:46-d0b6e65b-6f36-4338-bd9f-e93ac4c47bf4
IDX: 6758b113-eaf6-45fc-9444-9daa8618429f

Checking Request: CO-0206-11:46-d...
06.02.2025, 11:47 Balkon: 2
Carrier: 366

Carrier bearbeiten

Kisten und Ladungen erstellen, löschen und anpassen

↶ ↷ Komprimieren + Erweitern 🗑️ Ladung Entfernen →

1
+

Ladung Bitte wählen Sie eine Box aus

Änderungen speichern

Die Box wird nun erstellt, erscheint in der Übersicht und ist für weitere Bearbeitungen (am grünen Rand markiert) vorselektiert.

5. Fügen Sie eine neue Box hinzu, indem Sie auf den mit einem Plus gekennzeichneten Bereich (1) klicken.

The screenshot shows the NOYES dashboard interface. At the top, there is a header with the NOYES logo, a status indicator 'In Betrieb', and a 'System pausieren' button. The main content area is titled 'Aufgaben' and contains a 'Bereit' section with a 'Fetch' button and a 'Checking' button. A request card is displayed with details: 'Request: CO-0206-11:46-d...', '06.02.2025, 11:47', 'Balkon: 2', and 'Carrier: 366'. To the right, the 'Carrier bearbeiten' section is active, featuring a grid of boxes. A large green box is selected and labeled '1'. Above the grid are buttons for 'Komprimieren' (2), 'Erweitern' (3), and 'Ladung Entfernen'. Below the grid is a 'Ladung' section with a 'Ladung hinzufügen...' button. At the bottom of the 'Carrier bearbeiten' section is a 'Änderungen speichern' button (4). A back arrow button (5) is located at the top left of the 'Carrier bearbeiten' section.

6. Passen Sie die Größe der selektierten Box (1) mittels der Schaltflächen **Komprimieren** (2) oder **Erweitern** (3) an.

7. Nach Abschluss der Änderungen klicken Sie auf **Änderungen Speichern** (4) oder die **Zurück**-Schaltfläche (5), um zum Kommissioniervorgang zurückzukehren oder weitere Anpassungen vorzunehmen.

Löschen von Boxen auf einem Ladungsträger

Boxen können von einem Ladungsträger entfernt werden. Zuerst muss jedoch die Beladung der Box entfernt werden (vgl. Abschnitt [Beladung entfernen](#)). Dieser

Abschnitt beschreibt das Entfernen einer leeren Box.

1. Bestellen Sie einen Ladungsträger zum Balkon

The screenshot shows the NOYES dashboard interface. At the top, there is a navigation bar with the NOYES logo, a status indicator 'In Betrieb', and a 'System pausieren' button. The main content area is titled 'Aufgaben' and contains a list of jobs under the 'Bereit' status. One job card is highlighted with a green border, showing details like 'Request: CO-0206-11:46-d...', 'Balkon: 2', and 'Carrier: 366'. To the right, the 'Job Details' section displays a 'Balkon 2' layout with four numbered boxes (1, 2, 3, 4) and a 'Bestätigen' button. Box 1 is a dashed outline, box 2 contains a plus sign, box 3 is empty, and box 4 contains a pattern of dots.

2. Öffnen Sie durch Klicken auf die Job-Karte die Übersicht der Job-Details, welche die Boxen anzeigt:

3. (2) freier Stellplatz (Boxgröße Medium)

4. (3) leere Box

5. (4) Box mit Beladung

6. Passen Sie den Ladungsträger und dessen Inhalte durch Klicken auf die Bearbeiten-Schaltfläche (1) an.

7. **Neue Seite:**

NOYES TECHNOLOGIES In Betrieb System pausieren 🇩🇪

Aufgaben

Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.

Gruppieren nach: Requests ⌵ ⌵

Bereit

Fetch ID: CO-0206-11:46-d0b6e65b-6f36-4338-bd9f-e93ac4c47bf4
 IDX: 6758b113-eaf6-45fc-9444-9daa8618429f

Checking Request: CO-0206-11:46-d... Balkon: 2
 06.02.2025, 11:47 Carrier: 366

⬆️

Carrier bearbeiten

Kisten und Ladungen erstellen, löschen und anpassen

⌵ Komprimieren + Erweitern 🗑 Ladung Entfernen ➔ 1

Ladung Ladung hinzufügen... >

Änderungen speichern

8. Klicken Sie auf **Box entfernen**, um die grün markierte Box zu löschen.

NOYES TECHNOLOGIES In Betrieb System pausieren 🇩🇪

Aufgaben

Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.

Gruppieren nach: Requests ⌵ ⌵

Bereit

Fetch ID: CO-0206-11:46-d0b6e65b-6f36-4338-bd9f-e93ac4c47bf4
IDX: 6758b113-eaf6-45fc-9444-9daa8618429f

Checking Request: CO-0206-11:46-d...
06.02.2025, 11:47

Balkon: 2
Carrier: 366

⬆️

Carrier bearbeiten

Kisten und Ladungen erstellen, löschen und anpassen

←
Box Entfernen
→

Ladung
Ladung hinzufügen... >

Änderungen speichern

9. Dieser Vorgang bedarf der finalen Bestätigung durch den Nutzer.

4.1.10 Management von Beladung

MANAGEMENT VON BELADUNG

Zuletzt wird Ware einer Box zugewiesen, sodass der NoyesStorage den vorhandenen Bestand kennt. Die Beladung von Boxen kann angepasst werden, wenn der entsprechende Ladungsträger am Balkon steht und das System auf Nutzerinput wartet (z. B. während eines Kommissioniervorgangs).

Der folgende Abschnitt beschreibt das Hinzufügen und Entfernen einer Beladung.

Beladung zu einer Box hinzufügen

1. Bestellen Sie einen Ladungsträger zum Balkon

The screenshot shows the NoyesStorage management interface. At the top, there is a navigation bar with the Noyes logo, a status indicator 'In Betrieb', and a 'System pausieren' button. The main content area is titled 'Aufgaben' and displays a job card for 'Bereit'. The job card includes a 'Fetch' button, a 'Checking' button, and details for 'Request: CO-0206-11:46-d...' and 'Carrier: 366'. To the right, the 'Job Details' section shows a 'Balkon 2' layout with four boxes. Box 1 is a large dashed box with a plus sign, Box 2 is a medium box, Box 3 is an empty box, and Box 4 is a box with a pattern. A 'Bestätigen' button is located below the balcony layout.

2. Öffnen Sie die Übersicht der Job-Details, die die Boxen anzeigt:

3. (2) freier Stellplatz (Boxgröße Medium)

4. (3) leere Box

5. (4) Box mit Beladung

6. Passen Sie den Ladungsträger und dessen Inhalte durch Klicken auf die Bearbeiten-Schaltfläche (1) an.

7. **Neue Seite:**

NOYES TECHNOLOGIES In Betrieb System pausieren 🇩🇪

Aufgaben

Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.

Gruppieren nach: Requests ⌵ ⌵

Bereit

Fetch ID: CO-0206-11:46-d0b6e65b-6f36-4338-bd9f-e93ac4c47bf4
IDX: 6758b113-eaf6-45fc-9444-9daa8618429f

Checking Request: CO-0206-11:46-d...
06.02.2025, 11:47 Balkon: 2
Carrier: 366

⬆

Carrier bearbeiten

Kisten und Ladungen erstellen, löschen und anpassen

↶ Komprimieren + Erweitern 🗑 Ladung Entfernen →

1

LadungLadung hinzufügen... ➤

Änderungen speichern

8. Wählen Sie die Box, die Sie beladen möchten, durch Klicken (1) und anschließend auf **Ladung hinzufügen** (2).
9. Ein neues Eingabefeld öffnet sich:

NOYES TECHNOLOGIES In Betrieb System pausieren 🇩🇪

Aufgaben
Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.

Gruppieren nach: Requests ⌵

Bereit

Fetch ID: CO-0206-11:46-d0b6e65b-6f36-4338-bd9f-e93ac4c47bf4
IDX: 6758b113-eaf6-45fc-9444-9daa8618429f

Checking Request: CO-0206-11:46-d... 06.02.2025, 11:47 Balkon: 2
Carrier: 366

⬆️

Ladung bearbeiten
Ladung erstellen

SKU * 1

H000054 [D], Noyes Storage - Tube pillar high

Menge 2

23

Maximale Menge

50

Änderungen speichern

0. Füllen Sie die erforderlichen Informationen aus: Wählen Sie die SKU (1) und geben Sie die einzulagernde sowie die maximal mögliche Menge an. Bestätigen Sie mit **Änderungen Speichern**.

1. Die Beladung ist nun der Box zugewiesen und im NoyesStorage verfügbar.

Beladung von einer Box entfernen

1. Bestellen Sie einen Ladungsträger zum Balkon

The screenshot shows the NOYES dashboard interface. At the top, there is a status bar with 'In Betrieb' and a 'System pausieren' button. The main content area is divided into two sections: 'Aufgaben' (Tasks) and 'Job Details'.

Aufgaben: This section shows a task titled 'Bereit' (Ready) with a 'Fetch' button. Below it, a 'Checking' button is visible. The task details include:

- ID: CO-0206-11:46-d0b6e65b-6f36-4338-bd9f-e93ac4c47bf4
- IDX: 6758b113-eaf6-45fc-9444-9daa8618429f
- Request: CO-0206-11:46-d... (06.02.2025, 11:47)
- Balkon: 2
- Carrier: 366

Job Details: This section is titled 'Balkon 2' and shows a grid of boxes. The boxes are numbered 1 through 4:

- 1: Edit icon (pencil)
- 2: A box with a plus sign (+)
- 3: An empty box
- 4: A box containing a grid of dots (representing a load)

 Below the grid is a 'Bestätigen' (Confirm) button.

The left sidebar contains various navigation icons, including a home icon, a list icon, a cart icon, a document icon, a settings icon, a clock icon, and a monitor icon.

2. Öffnen Sie die Übersicht der Job-Details, die die Boxen anzeigt:
3. (2) freier Stellplatz (Boxgröße Medium)
4. (3) leere Box
5. (4) Box mit Beladung
6. Passen Sie den Ladungsträger und dessen Inhalte durch Klicken auf die Bearbeiten-Schaltfläche (1) an.
7. **Neue Seite:**

Aufgaben

Dieses Dashboard zeigt offene Jobs, die zur Interaktion bereit sind. Klicken Sie auf eine Karte, um Details anzuzeigen.

Gruppieren nach: Requests

Bereit

Fetch ID: CO-0206-11:46-d0b6e65b-6f36-4338-bd9f-e93ac4c47bf4
 IDX: 6758b113-eaf6-45fc-9444-9daa8618429f

Checking Request: CO-0206-11:46-d... 06.02.2025, 11:47
 Balkon: 2
 Carrier: 366

Carrier bearbeiten
 Kisten und Ladungen erstellen, löschen und anpassen

Komprimieren + Erweitern Ladung Entfernen

Ladung Ladung hinzufügen...

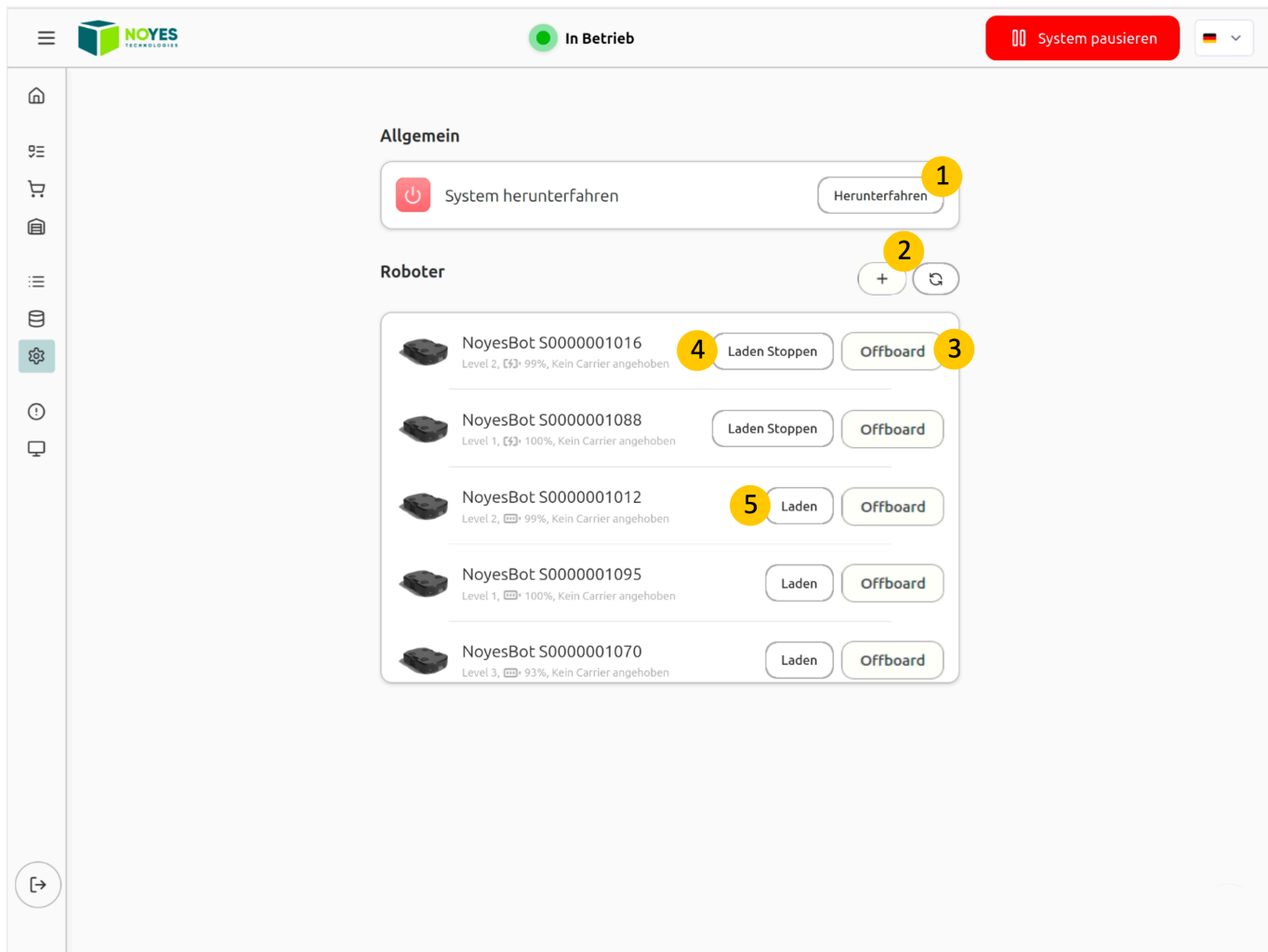
Änderungen speichern

8. Wählen Sie die Box, deren Beladung Sie entfernen möchten, durch Klicken (1).
9. Klicken Sie auf **Ladung entfernen** (2).
10. Dieser Vorgang bedarf der finalen Bestätigung durch den Nutzer.
11. Die Beladung ist nun entfernt und steht im NoyesStorage nicht mehr zur Verfügung.

4.1.11 Systemsteuerung

SYSTEMSTEUERUNG

Im Menü **Systemsteuerung** können grundlegende Systemprozesse gestartet werden, die für den täglichen Betrieb üblicherweise nicht erforderlich sind.



NoyesBot Onboarding

Das Ziel dieses Prozesses ist es, NoyesBots zum NoyesStorage hinzuzufügen – entweder bei der Erstinbetriebnahme oder wenn ein zusätzlicher Bot hinzugefügt werden soll.

Dieser Prozess findet sich im Menü **Systemsteuerung** unter Punkt (2).

Personal: Store Manager

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum Abschnitt **Systemsteuerung**.
3. Klicken Sie auf die Plus-Schaltfläche im Bereich **Roboter** (2), um einen neuen NoyesBot hinzuzufügen.
4. Sie werden automatisch auf das Dashboard weitergeleitet.
5. Folgen Sie den weiteren Anweisungen auf dem Bildschirm.

NoyesBot Offboarding

Das Ziel dieses Prozesses ist es, einen NoyesBot aus dem NoyesStorage zu entfernen (Offboarding), um ihn zu warten, Fehler zu beheben oder auszutauschen.

Dieser Prozess befindet sich im Menü **Systemsteuerung** unter Punkt (3).

Personal: Store Manager

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum Abschnitt **Systemsteuerung**.
3. Wählen Sie den NoyesBot, den Sie entfernen möchten, in der Roboterübersicht aus.
4. Klicken Sie auf **Offboard** (3), um den ausgewählten NoyesBot aus der Flotte zu entfernen.
5. Sie werden automatisch auf das Dashboard weitergeleitet.
6. Folgen Sie den weiteren Anweisungen auf dem Bildschirm.

NoyesStorage starten

Das Ziel dieses Prozesses ist es, den NoyesStorage zu starten, nachdem dieser heruntergefahren wurde.

Voraussetzungen:

- Der Storage wurde gemäß dem nachfolgenden Prozess zum Herunterfahren heruntergefahren.

Dieser Prozess befindet sich im Menü **Systemsteuerung**.

Personal: Store Manager

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum Abschnitt **Systemsteuerung**.
3. Klicken Sie auf **Starten**, um den NoyesStorage zu starten.
4. Nachdem der NoyesStorage hochgefahren wurde, werden Sie gefragt, ob Sie bestehende Jobs fortführen oder abrechnen möchten:
5. **Fortführen:** Alle übermittelten Aufträge, die vor dem Herunterfahren nicht vollendet wurden, werden wieder aufgenommen.
6. **Cancel:** Unvollendete Aufträge werden abgebrochen und nicht fortgeführt.

NoyesStorage herunterfahren

Ziel dieses Prozesses ist es, den NoyesStorage herunterzufahren (z. B. bei einem Fehler, umfangreichen Wartungsarbeiten oder Betriebsurlaub).

Dieser Prozess befindet sich im Menü **Systemsteuerung**.

Personal: Store Manager

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum Abschnitt **Systemsteuerung**.
3. Klicken Sie auf **Herunterfahren**, um den NoyesStorage anzuhalten und herunterzufahren.
4. Ein Popup erscheint und fragt nach dem Grund für den Neustart. Wählen Sie einen Grund und fügen Sie optional weitere Informationen im Notizfeld hinzu – Ihre Angaben helfen dem Entwicklerteam, den NoyesStorage kontinuierlich zu verbessern.
5. Fahren Sie das System herunter, indem Sie auf **Aktion Bestätigen** klicken.

NoyesBot Laden Beenden

Das Ziel dieses Prozesses ist es, den Ladevorgang eines NoyesBot manuell zu beenden (z. B. wenn der Roboter für einen Auftrag benötigt wird).

Dieser Prozess befindet sich im Menü **Systemsteuerung** (Punkt 4).

Personal: Store Manager

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum Abschnitt **Systemsteuerung**.
3. Wählen Sie den NoyesBot in der Roboterübersicht aus.
4. Klicken Sie auf **Laden stoppen**, um den Ladevorgang des ausgewählten NoyesBot zu beenden.

NoyesBot zum Laden Schicken

Das Ziel dieses Prozesses ist es, einen NoyesBot manuell zum Laden zu schicken (z. B. wenn zu einem späteren Zeitpunkt eine volle Batterie benötigt wird).

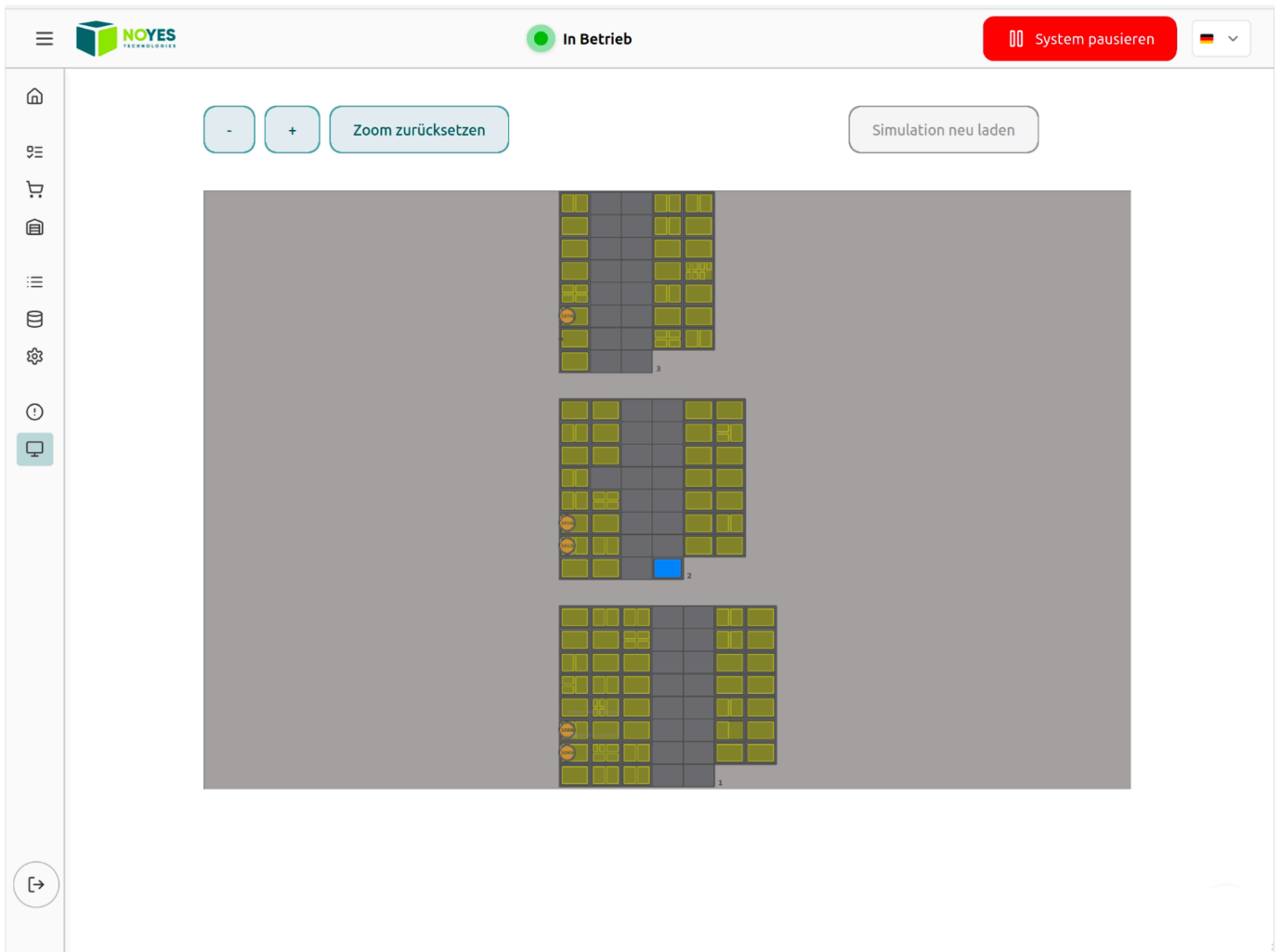
Dieser Prozess befindet sich im Menü **Systemsteuerung** (Punkt 5).

Personal: Store Manager

1. Melden Sie sich in der NoyesStorage App an.
2. Navigieren Sie zum Abschnitt **Systemsteuerung**.
3. Wählen Sie den NoyesBot in der Roboterübersicht aus.
4. Klicken Sie auf **Laden**, um den ausgewählten NoyesBot zum Laden zu schicken.

Systemansicht

Der Benutzer kann mithilfe des digitalen Zwillings einen detaillierten Einblick in das System erhalten. Der digitale Zwilling stellt den Zustand der Hardware – insbesondere der Roboter, Ladungsträger, Boxen und Ladestationen – digital dar. Auf diese Ansicht kann über den Menüpunkt **Systemansicht** zugegriffen werden. Sie bietet einen tiefgehenden Einblick in das System und wird primär zur Fehlerbehebung eingesetzt.



4.1.12 Datenexport

Die Exportfunktion ermöglicht es, gefilterte Daten aus Tabellen per E-Mail zu versenden – direkt aus der App heraus. So können relevante Informationen einfach mit Teammitgliedern oder externen Partnern geteilt werden.

Export starten

Der Export wird direkt über die jeweilige Tabelle ausgelöst.

The screenshot shows the 'SKU Stammdaten' interface in the NOYES application. At the top, there is a status bar with 'In Betrieb' and a 'System pausieren' button. Below the header, a table lists various SKUs. The 'Tabelle exportieren' button is highlighted with a yellow circle containing the number '1'. The table has columns for ID, Name, Einheit, Auf Lager, Nachfüllbar, Bei Leerstand, and Aktualisiert. Each row includes a checkbox and a button to 'Zuweisung aufheben' or 'Nichts tun'.

ID	Name	Einheit	Auf Lager	Nachfüllbar	Bei Leerstand	Aktualisiert
H001163 [B]	Balcony - Rubber Top Short	Stück (Stk.)	2	98	Zuweisung aufheben	04 Feb 2025 10:28
H001080 [D]	Balcony - Safety bumper Short	Stück (Stk.)	0	0	Zuweisung aufheben	31 Oct 2024 17:03
H001080 [C]	Balcony - Safety bumper Short	Stück (Stk.)	28	42	Zuweisung aufheben	31 Oct 2024 17:00
DEMO-03	Banane	Stück (Stk.)	0	0	Nichts tun	05 Nov 2024 16:19
MAR-30 Merch	Booth Supplies Plants	Stück (Stk.)	1	0	Zuweisung aufheben	21 Mar 2025 14:42
Tool	Bosch PST900 PEL	Stück (Stk.)	1	0	Zuweisung aufheben	25 Feb 2025 15:29
NBC015 [A]	Charging - Daisy Chain cable	Stück (Stk.)	9	41	Zuweisung aufheben	18 Nov 2024 14:23
NBC011 [A]	Charging Station Assembly	Stück (Stk.)	3	9	Zuweisung aufheben	07 Nov 2024 15:51
chris-1	chris first product	Stück (Stk.)	0	0	Zuweisung aufheben	29 Oct 2024 16:04
V-1200067843	Coffee Cup	Stück (Stk.)	0	0	Zuweisung aufheben	19 Dec 2024 23:36

1. Tippen Sie auf die Schaltfläche **Exportieren** (1) oben rechts in der Tabellenansicht. Der Exportdialog wird geöffnet.

HINWEIS

Der Export ist nur mit aktiver Internetverbindung möglich und auf den Export von 20.000 Elementen je Export begrenzt.

E-Mail-Export konfigurieren

Im Dialog wählen Sie Format und Empfängeradresse aus und senden die Daten direkt per E-Mail.

The screenshot shows the 'SKU Stammdaten' page in the NOYES system. A modal dialog box is open, titled 'Wie sollen wir diese Daten exportieren?'. The dialog contains the following elements:

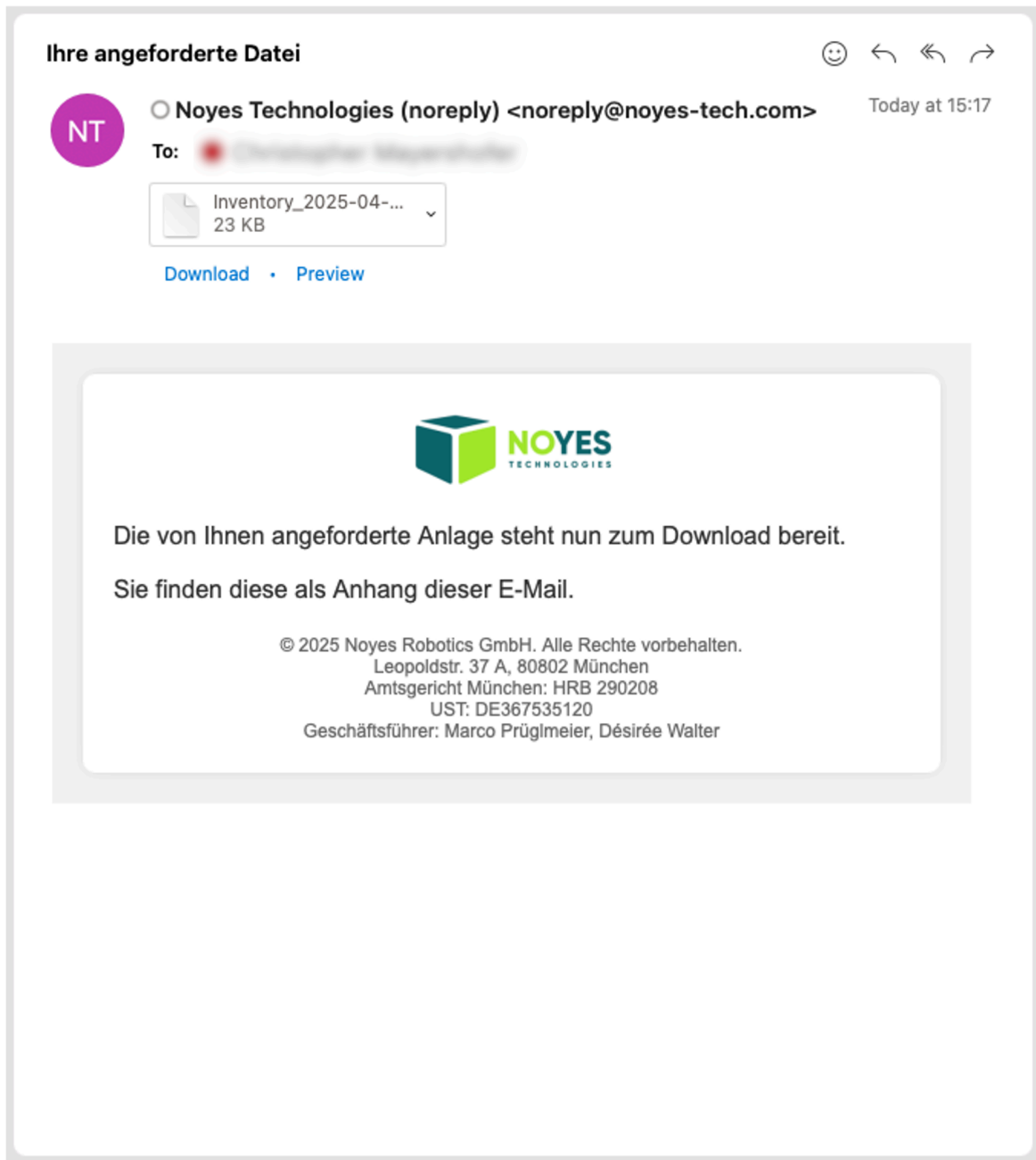
- Format:** A dropdown menu with 'CSV' selected. A yellow circle with the number '1' is next to it.
- Ziel:** A dropdown menu with 'Email (support@noyes-tech.com)' selected. A yellow circle with the number '2' is next to it.
- Buttons:** A green button labeled 'Daten exportieren' with a yellow circle and the number '3' next to it.

The background shows a table of SKUs with columns for ID, Name, and Aktualisiert. The table is partially obscured by the dialog box.

- 1. Format wählen (1):** Wählen Sie zwischen CSV oder XLS. Das Trennzeichen wird automatisch entsprechend der Spracheinstellung gesetzt.
- 2. Empfängeradresse angeben (2):** Standardmäßig ist die E-Mail-Adresse des aktuell angemeldeten Nutzers vorausgefüllt. Alternativ können andere Benutzer ausgewählt oder eine Adresse manuell eingetragen werden.
- 3. Export auslösen (3):** Tippen Sie auf **Daten exportieren**, um den Export zu starten. Die Datei wird an die gewählte Adresse verschickt.

Gesendete E-Mail

Nach dem erfolgreichen Export erhält der Empfänger eine E-Mail mit der exportierten Datei als Anhang.



1. Die E-Mail enthält eine kurze Beschreibung der exportierten Tabelle.
2. Die Datei ist als Anhang im gewählten Format (CSV oder XLS) beigefügt.

Einschränkungen & Hinweise

- Der Export enthält ausschließlich die aktuell gefilterten Daten und sichtbaren Spalten der Tabelle.
- Der Export per E-Mail ist nur bei aktiver Internetverbindung möglich.
- Die maximale Dateigröße für E-Mail-Anhänge beträgt **10 MB**.

4.1.13 Lagerstrategie

ÜBERSICHT

Der NoyesStorage unterstützt zwei Betriebsarten:

1. **Entnahme und Lagerung von Lagerplätzen (Carrier)**
2. **Entnahme und Lagerung von SKUs** (als Fulfillment und Replenishment bezeichnet)

Diese Dokumentation beschreibt die Betriebsart der SKU-basierten Entnahme und Lagerung, insbesondere die **Lagerstrategie**, die die Regeln festlegt, welche Artikel als nächstes entnommen oder nachgefüllt werden sollen. Ein klares Verständnis dieser Regeln stellt sicher, dass der NoyesStorage in entsprechende Prozesse eingebunden werden kann.

Die Lagerstrategie kann global durch den Support für den NoyesStorage festgelegt werden und gilt für alle SKUs.

Lagerstrategie 1: Geschwindigkeit (Standardstrategie)

ZIEL

Priorisierung der Entnahmegeschwindigkeit über alle anderen Einschränkungen oder Inventarregeln.

OPERATIVE LOGIK

- **Fulfillment:** Entnahme von Artikeln aus der Box, die den Balkon am schnellsten erreicht, unabhängig von Erstellungs- oder Ablaufdaten.
- **Replenishment:** Einlagerung von Artikeln in die Box, die den Balkon am schnellsten erreicht, unabhängig von Erstellungs- oder Ablaufdaten.

BEISPIELE

Szenario:

Box	Aktuelle Menge	Maximale Menge	Erstellungsdatum	ETA zum Balkon
Box 1	40	100	-	3 Minuten
Box 2	80	100	-	4 Minuten
Box 3	100	100	-	2 Minuten

Fulfillment: Entnahme von 80 Artikeln - Entnahme von 80 Artikeln aus Box 3, weil diese am schnellsten zum Balkon gelangt.

Replenishment: Einlagerung von 50 Artikeln - Einlagerung von 50 Artikeln in Box 1, weil diese am schnellsten zum Balkon gelangt.

Lagerstrategie 2: FIFO (First In, First Out)

ZIEL

Entnahme des ältesten Bestands zuerst, basierend auf dem Erstellungsdatum der Boxbeladung.

OPERATIVE LOGIK

- **Fulfillment:** Priorisierung von Boxen mit dem frühesten Erstellungsdatum.
- **Replenishment:** Einlagerung durch Erstellung neuer, voll gefüllter Boxen.

HINWEIS

Diese Lagerstrategie unterstützt derzeit keine Auffüllung bestehender Boxen, da sonst die FIFO-Integrität verletzt würde.

BEISPIELE

Box	Aktuelle Menge	Maximale Menge	Erstellungsdatum	ETA zum Balkon
Box 1	100	100	vor 10 Tagen	2 Minuten
Box 2	100	100	vor 5 Tagen	4 Minuten

Fulfillment: Entnahme von 100 Artikeln - Entnahme von 100 Artikeln aus Box 1 (am ältesten erstellt)

Replenishment: Einlagerung von 100 Artikeln - Erstellung einer neuen Box und Zuweisung von 100 Artikeln

Zusammenfassung

NoyesStorage unterstützt die Lagerstrategien "Geschwindigkeit" (Standard) und "FIFO". Jede Strategie optimiert unterschiedliche operationale Bedürfnisse.

4.2 Integration über API

4.2.1 Übersicht

NoyesStorage ist so konzipiert, dass er eigenständig mittels App betrieben werden kann, ohne eine Integration zu erfordern. Soll der NoyesStorage automatisiert mit anderen Systemen kommunizieren, steht eine Schnittstelle (engl. *Application Programming Interface, API*) zur Verfügung. Dieser Abschnitt bietet eine Schnellstartanleitung zur Interaktion mit NoyesStorage über seine API und ermöglicht so die Integration von NoyesStorage in Anwendungsfälle wie:

- die Übergabe von Bestelldaten aus einem Online-Shop an NoyesStorage,
- das automatische Drucken von Rechnungen, wenn Kundenaufträge kommissioniert werden, und
- die Synchronisation des ERP-Systems mit den Lagerbeständen in NoyesStorage.

Die NoyesStorage-API basiert auf REST, verwendet das JSON-Datenformat für Anfragen und Antworten und nutzt gängige HTTP-Methoden (siehe Tabelle unten), um eine konsistente und vorhersehbare Schnittstelle bereitzustellen. Integrationen nutzen die öffentliche REST-API unter `/api/v2/...` (OpenAPI unter `/api/openapi.json`), z. B. `POST /api/v2/requests` für Abruf-, Kommissionier- und Nachschubaufträge.

Die API, ihre Endpunkte und Parameter sind ausführlich online dokumentiert unter docs.noyes-tech.com.

HTTP KOMMUNIKATION IN DER NOYESSTORAGE API

Methode	Erklärung
POST	Create new resources or send data to NoyesStorage (e.g., submitting a fulfillment request). Example: <code>/api/v2/requests</code>
GET	Retrieve data from NoyesStorage (e.g., checking the status of requests). Example: <code>/api/v2/requests</code>
PUT	Update existing resources.
DELETE	Remove resources from NoyesStorage.

STANDARD HTTP ANTWORTEN UND STATUS DER NOYESSTORAGE API

Status	Erklärung
200 OK	The request succeeded.
201 Created	A new resource was successfully created.
202 Accepted	The request has been accepted but is being processed asynchronously.
400 Bad Request	The request had a syntax error that could not be resolved.
401 Unauthorized	Authentication is required.
404 Not Found	The requested resource does not exist.
500 Internal Server Error	The server encountered an error.

Um die Schnittstelle vor böswilligen Nutzern zu schützen, benötigen alle Endpunkte eine Authentifizierung über einen API-Schlüssel. Jede Anfrage muss diesen API-Schlüssel im Header enthalten, andernfalls wird die Anfrage abgelehnt. Zusätzlich werden nur Anfragen akzeptiert, die über HTTPS gesendet werden. Dies stellt sicher, dass die Daten vor der Übertragung verschlüsselt und nur von der NoyesStorage-API entschlüsselt werden können. Anfragen, die über unverschlüsseltes HTTP oder ohne gültigen API-Schlüssel gesendet werden, schlagen fehl.

API-KEY VERWALTEN

Hinweis

Nur verfügbar für Admin-Nutzer.

- Öffnen Sie in der NoyesStorage App das Menü `Einstellungen` und wählen Sie `API-Key`.
- Bestehende Schlüssel können Sie dort einsehen und kopieren; erstellen Sie bei Bedarf einen neuen Schlüssel für Integrationen.

- Bewahren Sie den Schlüssel sicher auf und teilen Sie ihn nur mit autorisierten Systemen.

API-Key-Verwaltung
Verwalten Sie API-Key fur Systemzugriff und Authentifizierung.

Suchen... Alle Filter loschen + API-Key hinzufugen API-Key loschen

ID	Name	API-Key	Lauft ab am	Zuletzt verwendet
3	ERP Integration	4ca0...	31 Dec 9999 23:59	N/A

Angezeigte Elemente: 1 / 1 - Ausgewahlt: 0 Zeile(n)

AUTHENTIFIZIERUNG UND API-KEY

Produktive Integrationen authentifizieren jede Anfrage mit einem API-Key, der in der NoyesStorage App ausgegeben wurde. Senden Sie den Schlussel als Bearer Token im HTTP-Header `Authorization`:

```
Authorization: Bearer <api-key>
```

Behandeln Sie API-Keys wie Passwortern: nicht in Quellcode einchecken, nur in Secret Stores oder geschutzten Umgebungsvariablen speichern und bei Verdacht auf Weitergabe in der App loschen und neu erstellen.

FILTERN, SUCHEN, SORTIEREN UND PAGINATION

Listen-Endpunkte wie `GET /api/v2/skus`, `GET /api/v2/inventory_view`, `GET /api/v2/requests`, `GET /api/v2/jobs` und `GET /api/v2/events` unterstutzen je nach Ressource Filterparameter, Suche, Sortierung und Pagination.

Filter verwenden das Schema `field__operator=value`. Hufige Operatoren sind `__eq` fur exakte Treffer, `__gte` und `__lte` fur Bereiche sowie `__ilike` fur teilweise, gro-/kleinschreibungsunabhangige Treffer. Bei `__in` wird der Query-Parameter pro Wert wiederholt:

```
GET /api/v2/inventory_view?level_id__in=1&level_id__in=2
```

Kommagetrennte Listen wie `?level_id__in=1,2` werden nicht als mehrere Werte interpretiert.

Der Parameter `search` nutzt `field:value`-Ausdrucke. Die Suche ist teilweise und gro-/kleinschreibungsunabhangig. Die unterstutzten Suchfelder sind endpunktspezifisch und in der `search`-Parameterbeschreibung des jeweiligen Endpunkts aufgefuhrt:

```
GET /api/v2/skus?search=id:tee
```

Mehrere Suchbegriffe können mit den großgeschriebenen Operatoren **AND** und **OR** kombiniert werden. Eine Ebene von Klammern wird unterstützt:

```
GET /api/v2/inventory_view?search=sku_id:tee AND (box_id:2 OR carrier_id:2)
```

Mit ***:value** wird über alle für den jeweiligen Endpunkt unterstützten Suchfelder gesucht:

```
GET /api/v2/inventory_view?search=:tee
```

Der Parameter **sort_by** akzeptiert kommagetrennte Feldnamen. Ein Feld ohne Vorzeichen oder mit **+** sortiert aufsteigend, **-** sortiert absteigend:

```
GET /api/v2/requests?sort_by=created_at,-updated_at
```

Pagination verwendet **page** und **size**. Die Seitennummerierung beginnt bei **page=1**; ohne abweichende Anfrage liefert die API standardmäßig **page=1** und **size=50**:

```
GET /api/v2/skus?page=1&size=50
```

Funktionsweise der Anfrageausführung

Das NoyesStorage-System arbeitet mit einer Warteschlange. Ein spezifisches Arbeitspaket kann durch das Senden einer Anfrage angefordert werden. Intern wird diese Anfrage in mehrere Jobs aufgeteilt, die nacheinander ausgeführt werden. Sowohl Anfragen als auch Jobs folgen einem vordefinierten Lebenszyklus, der durch ihren Status ausgedrückt wird.

REQUESTS

Anfragen (engl. *Requests*) können an den NoyesStorage gesendet werden. Diese Anfrage wird dann intern in eine Reihe von Jobs übersetzt. Die folgende Tabelle listet die verfügbaren Anfragen und deren Beschreibungen auf. Über die API können *Fetch*, *Fulfillment* und *Replenishment* Anfragen an den NoyesStorage gesendet werden, um die Aus- und Einlagerung automatisiert abzuwickeln.

Requests im NoyesStorage

Request	Description
FULFILLMENT	A Logistic Request consisting of Pick-Entities (ProductEntity).
REPLENISHMENT	A Logistic Request consisting of Refill-Entities (ProductEntity).
FETCH	A Logistic Request consisting of Fetch-Carrier-Entities.
RFID_MAINTENANCE	A System Control Request to write and/or check RFID Tags on a specified Level.
ONBOARD	A System Control Request to add a Noyes Bot to any Level.
OFFBOARD	A System Control Request to remove a specific Noyes Bot from a Level.
PAUSE	A System Control Request to remove a specific Noyes Bot from a Level.
RESUME	A System Control Request to allow all Bots to get new commands after a PAUSE Request.
CHARGING	A System Control Request to move a Noyes Bot to a Charging Station.
UNCHARGING	A System Control Request to remove a Noyes Bot from a Charging Station.
UPDATE_CONTENT_CODE	An Inventory Update Request to update the content code of a carrier.
RECHARGE	A System internal Request sent from the Charging Manager to the Store Orchestrator.
LEVEL_INITIALIZATION	An internal Request Type used for initializing each Level of a Storage by connecting to the bots, updating the Database, and clearing the Highway.

JOBS

Jobs sind die einzelnen Aufgaben, die im System ausgeführt werden. Jede Anfrage kann mehrere Jobs generieren, die jeweils einem vordefinierten Lebenszyklus folgen. Die folgende Tabelle liefert einen Überblick über die im NoyesStorage verwendeten Jobs.

Job	Description
BUFFERING	Moves a Carrier to a Buffer Space on the same Level.
RETRIEVING	Moves a Carrier to a Balcony on the same Level.
STORING	Moves a Carrier from the Balcony.
PICKING	Lets the user pick a given amount out of a Box.
PICKING_ALERT	Lets the user handle fulfillment quantities outside the NoyesStorage when the external picking feature is enabled.
REFILLING	Lets the user refill a given amount into a Box.
CHECKING	Lets the user check a given Carrier Label.
PACKING	Lets the user check the summary of a Request after all Entities have been worked on.
ONBOARDING	Lets the user onboard a Bot to any level.
ONBOARDING_BD	Adds a Bot to the Database and informs all Brain Components about it.
OFFBOARDING	Moves a Bot to the Balcony.
OFFBOARDING_BD	Lets the user remove a Bot from the Balcony.
OFFBOARDING_BC	Moves a Bot to the Balcony.
PAUSE	Ensures no Bot gets new commands.
RESUME	Allows all Bots to get new commands after a PAUSE.
MOVING_BOT	Moves a Bot to a target location/rotation/lifting state.
CONNECT_TO_BOTS	Establishes a connection to all Bots that are in the Database.
REFRESH_LEVEL	Updates the digital twin in the Bot Coordinator to the Database.
UNCHARGE_ALL	Removes all Bots from the Charging Station.
CLEAR_AUTOBAHN	Removes all Carriers from the Highway.
CHARGING	Moves a given bot to a Charging Station.
UNCHARGING	Removes a given bot from the Charging Station.
CHECK_RFID_TAG	Checks the values written on an RFID Tag at the given Location and Rotation.
WRITE_RFID_TAG	Writes values to an RFID Tag at the given Location and Rotation.
CHANGE_RFID_WRITING_MODE	Changes whether or not a bot should ignore the values on an RFID Tag while driving.

Hinweis zu PICKING_ALERT

PICKING_ALERT erscheint nur, wenn die externe Kommissionierung für fehlende oder außerhalb des NoyesStorage zu bearbeitende Fulfillment-Mengen aktiviert ist. In diesem Fall können angenommene Fulfillment-Requests strukturierte Warnhinweise zu fehlenden oder extern zu bearbeitenden Mengen enthalten, und nachgelagerte Job-Daten können **PICKING_ALERT**-Jobs enthalten.

JOB STATES

Jeder Job durchläuft einen Lebenszyklus, der durch seinen Status definiert ist. Die folgenden Statusübergänge sind in der Tabelle dargestellt.

State	Description
ACCEPTED	Job is ACCEPTED. These transitions are allowed: EXECUTING, ERROR, CANCELLING, ABORTED.
EXECUTING	Job is EXECUTING and waiting to be triggered. These transitions are allowed: ERROR, CANCELLING, ABORTED.
SUCCEDED	Job is SUCCEDED. No transitions are allowed. This is a final state.
CANCELLING	Job is CANCELLING. A user requested this job to be cancelled. These transitions are allowed: CANCELLED, ABORTED.
CANCELLED	Job is CANCELLED. A user requested this job to be cancelled. The CANCELLING was successful. This is a final state.
ABORTED	Job is ABORTED due to a system problem. The system will try to recover from this. This is a final state.

Quickstart: Ein einfaches Integrationsbeispiel

Dieser Abschnitt bietet eine einfache Anleitung zur Nutzung der API und konzentriert sich auf zentrale Interaktionen über Endpunkte. Die Beispiele werden in Python demonstriert, jedoch können auch andere Sprachen wie JavaScript, Java oder C# verwendet werden.

Anforderungen

1. Python 3.7+ (oder eine ähnliche Programmiersprache, die HTTP-Requests senden kann).
2. API Keys für die Authentifizierung an der NoyesStorage API.

Im Folgenden finden Sie ein Beispiel in Python, das zeigt, wie die Integration mit der API von NoyesStorage – insbesondere dem Retrieval-Endpunkt – erfolgen kann. Nach der vollständigen Implementierung wird jeder Abschnitt im Detail erläutert.

```
import json
from typing import Dict
import time
import uuid
import requests
import logging
from threading import Timer

"""
Examples of Noyes B2B payload

v1/core/retrieval/{request_id}

{
  "priority": 10,
  "desired_time": "2024-10-08T07:50:36.064257",
  "source": "",
  "request_type": "FULFILLMENT",
  "external_id": "string",
  "total_entity_count": 0,
  "entities": [
    {
      "entity_type": "Product",
      "sku_id": "string",
      "quantity": 0
    },
    {
      "entity_type": "Carrier",
      "carrier_id": "string",
      "content_codes": [
        0
      ]
    }
  ]
}
"""

class SimpleERPIntegration():
    """ An example class that sends requests to the Noyes B2B API and
    repeatedly checks the status of those requests until they are finished.
    """
    REQ_UPDATE_PERIOD = 10 # seconds

    def __init__(self, store_keys: Dict, b2b_url: str) -> None:
        self._request_dicts: Dict[uuid.UUID, Dict[str]] = {}
        self._store_keys = store_keys
        self.b2b_url = b2b_url

    def send_request(self, payload: Dict) -> str:
        headers = self._store_keys
        logging.info(f'sending request {payload["external_id"]} via b2b')
        send_time = time.time()
        rsp = requests.post(self.b2b_url + 'core/retrieval', headers=headers, data=json.dumps(payload), timeout=1)
```

```

assert rsp.status_code == 202, f'got bad response {rsp} {rsp.reason} {rsp.text}'
logging.info(f'sent request {payload["external_id"]} request_id={rsp.json()["request_id"]} via b2b in {time.time()-send_time:.2f}')
self.subscribe_to_updates(rsp.json()["request_id"])
return rsp.json()["request_id"]

def subscribe_to_updates(self, request_id: uuid.UUID):
    """Creates a timer for this request which repeatedly asks the B2B for updates.
    self._check_request_callback(request_id) will be called at a rate of self.REQ_UPDATE_PERIOD.
    """
    self._request_dicts[request_id] = {}
    self._request_dicts[request_id]['timer'] = Timer(interval=self.REQ_UPDATE_PERIOD, function=lambda: self._check_request_callback(request_id))
    self._request_dicts[request_id]['last_request_update'] = ''
    self._request_dicts[request_id]['last_job_update'] = ''

def unsubscribe_from_updates(self, request_id: uuid.UUID):
    """
    Cancels the timer created for this request_id in subscribe_to_updates.
    """
    if request_id in self._request_dicts.keys():
        if 'timer' in self._request_dicts[request_id].keys():
            self._request_dicts[request_id]['timer'].cancel()
        self._request_dicts[request_id] = None
    del self._request_dicts[request_id]

def _check_request_callback(self, request_id: uuid.UUID):
    """
    Calls the Noyes B2B endpoint for the status of the specified request.
    Processes the response in _process_request_updates.
    Unsubscribes from request updates if the request is completed.
    """
    url = self.b2b_url + f'core/retrieval/{str(request_id)}'
    try:
        rsp = requests.get(url, headers=self._store_keys, timeout=0.9*self.REQ_UPDATE_PERIOD)
        assert rsp.status_code == 200, f'got bad response {rsp} {rsp.reason} {rsp.text}'
    except requests.exceptions.ReadTimeout as e:
        logging.warn(f'got error while checking requests {e}.')
        return
    request_update = json.loads(rsp.text)
    self._process_request_updates(request_update)

    if request_update['status'] in ['FAILED', 'SUCCEEDED', 'ABORTED']:
        self.unsubscribe_from_updates(request_id)

def _process_request_updates(self, request_update):
    """
    Determines when a subscribed request has updated information and provides a placeholder
    for user-defined request update callbacks.
    """
    # Parse out different portions of the message.
    request_id = request_update['request_id']
    request_updates = {k: v for k, v in request_update.items() if k not in ['jobs', 'content_code_updates', 'last_updated']}
    dump = json.dumps(request_updates)
    if request_id not in self._request_dicts.keys():
        self.subscribe_to_updates(request_id=request_id)
        logging.warn(f'got request update for {request_id} but it wasn't expected, tracking now but that\'s weird')

    if dump != self._request_dicts[request_id]['last_request_update']:
        pass # Place your request_update callback here.

    self._request_dicts[request_id]['last_request_update'] = dump

```

ERLÄUTERUNGEN ZUM CODE

1. Klasseninitialisierung

Die Klasse `SimpleERPIntegration` verwaltet alle Interaktionen mit der NoyesStorage-API. Sie wird mit folgenden Parametern initialisiert:

1. `store_keys`: Ein Dictionary, das die für die Authentifizierung erforderlichen Schlüssel enthält (z. B. `store_id`, `tenant_id`, `api_key`).
2. `b2b_url`: Die Basis-URL für die API.

Beispiel:

```

erp_integration = SimpleERPIntegration(
    store_keys={
        'store_id': 'YOUR_STORE_ID',
        'tenant_id': 'YOUR_TENANT_ID',
        'api_key': 'YOUR_API_KEY'
    },
    b2b_url='https://demo-b2b.noyes-tech.com/'
)

```

1. Anfrage senden (`send_request`)

Diese Methode wird verwendet, um eine neue Entnahmeanfrage an das NoyesStorage-System zu senden.

2. Verwendeter Endpunkt: `POST /core/retrieval`

3. Die Methode erstellt ein Payload und sendet eine `POST`-Anfrage an den Endpunkt `/core/retrieval`.

4. Bei Erfolg gibt die API den Statuscode `202 Accepted` und eine `request_id` zurück.

Wichtige Punkte: - Der Endpunkt `/core/retrieval` ist die Hauptschnittstelle zur Erstellung neuer Entnahmeaufträge. - Die zurückgegebene `request_id` ist wichtig, um den Status der Anfrage zu verfolgen.

1. Abonnieren von Updates (`subscribe_to_updates`)

Nach dem Abschicken einer Anfrage wird eine regelmäßige Abfrage eingerichtet (alle 10 Sekunden), um den Status der Anfrage und der zugehörigen Jobs zu verfolgen.

2. Überprüfung des Anfragestatus (`__check_request_callback`)

Diese Methode prüft den Status der Anfrage, indem sie eine `GET`-Anfrage an den Endpunkt `/core/retrieval/{request_id}` sendet. Die Antwort enthält den Fortschritt (z. B. `SUCCEEDED`, `FAILED` oder `ABORTED`).

3. Verarbeitung von Updates (`__process_request_updates`)

Die von der API empfangenen Updates werden in dieser Methode verarbeitet. Sie protokolliert den Status und löst je nach Zustand der Anfrage erforderliche Callbacks aus. Sobald der Job abgeschlossen ist (egal ob `SUCCEEDED`, `FAILED` oder `ABORTED`), beendet das System die Abfrage der API nach weiteren Updates.

Mögliche Erweiterungen

Dieses einfache Integrations-Setup dient als Vorlage zur Nutzung der NoyesStorage-API und kann erweitert werden, um die Integration von NoyesStorage in Ihre Produktionsumgebung anzupassen.

Im Folgenden sind einige Beispiele aufgeführt, wie Kunden die API nutzen, um NoyesStorage in ihre Prozesse zu integrieren:

DRUCKEN VON RECHNUNGEN FÜR KOMMISSIONIERTE ANFRAGEN

Eine Erweiterung, die von einem unserer Kunden genutzt wird, umfasst das automatische Erstellen und Drucken von Rechnungen für Anfragen, die aktuell kommissioniert werden. Diese Rechnungen werden vor dem Verpacken in die Kundenbestellung gelegt.

Im obigen Beispiel kann dies erreicht werden, indem die Methode `__process_request_updates` so modifiziert wird, dass der Rechnungsdruck ausgelöst wird, sobald eine Anfrage in den Status `EXECUTING` übergeht.

```
def __process_request_updates(self, request_update):
    request_id = request_update['request_id']
    request_status = request_update['status']

    if request_status == "EXECUTING":
        self.print_invoice_for_request(request_update)

    logging.info(f'Update for request {request_id}: {request_status}')

def print_invoice_for_request(self, request_update):
    invoice_data = f"Invoice for request {request_update['request_id']}\n"
    invoice_data += f"Items: {request_update['entities']}\n"
    invoice_data += f"Total Quantity: {sum(item['quantity'] for item in request_update['entities'])}\n"

    logging.info(f"Printing invoice:\n{invoice_data}")
```

Mit dieser Erweiterung wird eine Rechnung automatisch erstellt und protokolliert (oder gedruckt), sobald eine Anfrage in den Status `EXECUTING` übergeht.

AUTOMATISCHE BESTÄTIGUNG EINES PICKS ÜBER EINEN SCANNER

Eine zusätzliche Erweiterung umfasst die Verwendung eines Handscanners zur automatischen Bestätigung von Picks. Wenn der Scanner den Barcode eines Artikels liest, kann die Integration den Pick für die entsprechende Anfrage am NoyesStorage automatisch bestätigen.

Hierfür muss die Methode `__process_request_updates` angepasst werden, um zu erkennen, wann ein Artikel gescannt wurde. Anschließend kann eine Anfrage zur Bestätigung des Picks über den entsprechenden Endpunkt gesendet werden (z. B. `POST /v1/core/requests/{request_id}/jobs/{job_id}/trigger`).

```
def __process_request_updates(self, request_update):
    request_id = request_update['request_id']
    request_status = request_update['status']

    if request_status == "EXECUTING" and self.is_item_scanned(request_update):
        self.confirm_pick(request_id)

    logging.info(f'Update for request {request_id}: {request_status}')

def is_item_scanned(self, request_update):
    scanned_item_sku = "scanned_item_sku_example" # Example from the scanner
    return any(item['sku_id'] == scanned_item_sku for item in request_update['entities'])

def confirm_pick(self, request_id):
    url = self.b2b_url + f'/core/retrieval/{str(request_id)}/confirm'
    try:
```

```
rsp = requests.put(url, headers=self._store_keys)
assert rsp.status_code == 200, f"Failed to confirm pick: {rsp.status_code}"
logging.info(f'Pick confirmed for request {request_id}')
except requests.exceptions.RequestException as e:
    logging.error(f'Error confirming pick: {e}')
```

WEITERE ANWENDUNGSFÄLLE FÜR EINE API-INTEGRATION

- **Bestandsanpassung:** Automatische Aktualisierung der Bestandsführungssysteme basierend auf den abgeschlossenen Anfragen am NoyesStorage.
- **Verfolgung von Bestellungen:** Synchronisieren der NoyesStorage-Anfragen mit einem externen Bestellverfolgungssystem für Echtzeit-Updates.
- **Benachrichtigungssystem:** Auslösen von SMS- oder E-Mail-Benachrichtigungen, wenn kritische Jobs abgeschlossen oder verzögert werden.
- **Automatisierte Nachbestellung:** Überwachen der Lagerbestände und automatisches Erstellen von Nachschubanfragen basierend auf festgelegten Schwellenwerten.

Diese Beispiele zeigen, wie sich die Integrationsvorlage an spezifische Anforderungen anpassen lässt. Die NoyesStorage-API bietet die nötige Flexibilität, um Prozesse über Systemgrenzen hinweg zu automatisieren und zu optimieren. Dieses Beispiel verdeutlicht, wie die Integration der NoyesStorage-API eingerichtet, Anfragen gesendet und deren Status verfolgt werden kann. Die Einrichtung kann erweitert werden, um Prozesse wie die Rechnungserstellung, automatische Bestätigungen oder Nachbestellungen zu individualisieren.

API Endpoints (English)

Hinweis

Die API-Dokumentation ist derzeit nur in englischer Sprache verfügbar.

Hinweise zu Authentifizierung, Filtern, Suche, Sortierung und Pagination finden Sie in der [Integrationsübersicht](#).

4.2.2 The Noyes API 3e11ee9a7

Public OpenAPI documentation for NoyesStorage integrations. Customer integrations authenticate with API keys issued in the NoyesStorage App. Send the issued key in the Authorization header as a bearer token.

Contact: support@noyes-tech.com

Authentication

GET /API/V2/AUTH/API_KEYS

List API keys

Description

Returns paginated API-key records for administration, including the key values shown to authorized administrators.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters


Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>expires_at__gte</code>	query	string		No	Filter by expires at greater than or equal to this value.
<code>expires_at__lte</code>	query	string		No	Filter by expires at less than or equal to this value.
<code>id__eq</code>	query	string		No	Filter by exact id.
<code>id__ilike</code>	query	string		No	Filter by partial, case-insensitive match on id.
<code>last_used__gte</code>	query	string		No	Filter by last used greater than or equal to this value.
<code>last_used__lte</code>	query	string		No	Filter by last used less than or equal to this value.
<code>name__eq</code>	query	string		No	Filter by exact name.
<code>name__ilike</code>	query	string		No	Filter by partial, case-insensitive match on name.
<code>page</code>	query	integer	1	No	Result page to return. Pagination starts at page=1.
<code>role__eq</code>	query	string		No	Filter by exact role.
<code>role__in</code>	query	array		No	Filter by one or more values.
<code>search</code>	query	string		No	Search expression using supported field:value terms for this endpoint. Matching is partial and case-insensitive. Combine terms with uppercase AND/OR and one level of parentheses. Use *:value to search all supported fields. Supported fields: id, name, role.
<code>size</code>	query	integer	50	No	Maximum number of items to return on one page.
<code>sort_by</code>	query	string		No	Comma-separated sort fields. Use plain field or +field for ascending and -field for descending, for example created_at,-updated_at.

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "items": [
    {
      "api_key": "string",
      "expires_at": "2022-04-13T15:42:05.901Z",
      "expires_at_utc": "2022-04-13T15:42:05.901Z",
      "id": 0,
      "last_used": "2022-04-13T15:42:05.901Z",
      "last_used_utc": "2022-04-13T15:42:05.901Z",
      "name": "string",
      "role": "Super Admin"
    }
  ],
  "page": 0,
  "pages": 0,
  "size": 0,
  "total": 0
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "items": {
      "items": {
        "$ref": "#/components/schemas/ApiKeyInfo"
      },
      "title": "Items",
      "type": "array"
    },
    "page": {
      "minimum": 1.0,
      "title": "Page",
      "type": "integer"
    },
    "pages": {
      "minimum": 0.0,
      "title": "Pages",
      "type": "integer"
    },
    "size": {
      "minimum": 1.0,
      "title": "Size",
      "type": "integer"
    },
    "total": {
      "minimum": 0.0,
      "title": "Total",
      "type": "integer"
    }
  },
  "required": [
    "items",
    "total",
    "page",
    "size"
  ],
  "title": "Page[ApiKeyInfo]",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

DELETE /APIV2/AUTH/API_KEYS/{API_KEY_ID}

Delete an API key

Description

Deletes the selected API key. Integrations using that key can no longer authenticate after deletion.


Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
api_key_id	path	integer		No	API key id.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "deleted_key": "string",
  "deleted_key_id": 0,
  "deleted_key_name": "string",
  "message": "string"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "deleted_key": {
      "title": "Deleted Key",
      "type": "string"
    },
    "deleted_key_id": {
      "title": "Deleted Key Id",
      "type": "integer"
    },
    "deleted_key_name": {
      "title": "Deleted Key Name",
      "type": "string"
    },
    "message": {
      "title": "Message",
      "type": "string"
    }
  },
  "required": [
    "message",
    "deleted_key",
    "deleted_key_id",
    "deleted_key_name"
  ],
  "title": "ApiKeyDeleteResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET

/API/V2/AUTH/AUTHENTICATE

Validate an authenticated request

 **Description**

Checks the current request for Traefik ForwardAuth and returns whether it may continue to the protected service.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses

200 OK

application/json

```
{
  "settings": {},
  "user": {
    "email": "derp@meme.org",
    "id": 0,
    "role": "string",
    "username": "string"
  }
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

 **Schema of the response body**

```
{
  "properties": {
    "settings": {
      "title": "Settings",
      "type": "object"
    },
    "user": {
      "$ref": "#/components/schemas/AuthenticatedSubject"
    }
  },
  "required": [
    "user",
    "settings"
  ],
  "title": "AuthenticateResponse",
  "type": "object"
}
```

POST

/API/V2/AUTH/CREATE_ACCOUNT

Create a user account

Description

Creates a new NoyesStorage App user account with the requested login details and role.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "email": "derp@meme.org",
  "locale": null,
  "role": null,
  "username": "string"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "description": "Mixin that automatically normalizes email fields.",
  "properties": {
    "email": {
      "format": "email",
      "title": "Email",
      "type": "string"
    },
    "locale": {
      "allOf": [
        {
          "$ref": "#/components/schemas/LocaleEnum"
        }
      ],
      "default": "en"
    },
    "role": {
      "allOf": [
        {
          "$ref": "#/components/schemas/RoleEnum"
        }
      ],
      "default": "User"
    },
    "username": {
      "title": "Username",
      "type": "string"
    }
  },
  "required": [
    "username",
    "email"
  ],
  "title": "SignupRequest",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "email": "derp@meme.org",
  "id": 0,
  "username": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "email": {
      "format": "email",
      "title": "Email",
      "type": "string"
    },
    "id": {
      "title": "Id",
      "type": "integer"
    },
    "username": {
      "title": "Username",
      "type": "string"
    }
  },
  "required": [
    "id",
    "username",
    "email"
  ],
  "title": "SignUpResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST

/API/V2/AUTH/FORGOT_PASSWORD

Request a password reset code

 **Description**

Creates a one-time password reset code for the supplied user so the password can be reset.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.


Request body

application/json

```
{
  "email": "derp@meme.org",
  "locale": null
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

 **Schema of the request body**

```
{
  "description": "Mixin that automatically normalizes email fields.",
  "properties": {
    "email": {
      "format": "email",
      "title": "Email",
      "type": "string"
    },
    "locale": {
      "allOf": [
        {
          "$ref": "#/components/schemas/LocaleEnum"
        }
      ],
      "default": "en"
    }
  },
  "required": [
    "email"
  ],
  "title": "ForgotPasswordRequest",
  "type": "object"
}
```

Responses


200 OK 422 Unprocessable Entity
application/json

Schema of the response body

```
[REDACTED]
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST

/API/V2/AUTH/GENERATE_API_KEY

Generate an API key

Description

Creates a new API key for system integrations. Store the returned key securely; clients use it for authenticated API calls.


Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "name": "string"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "properties": {
    "name": {
      "title": "Name",
      "type": "string"
    }
  },
  "required": [
    "name"
  ],
  "title": "ApiKeyCreateInput",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "api_key": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "api_key": {
      "title": "Api Key",
      "type": "string"
    }
  },
  "required": [
    "api_key"
  ],
  "title": "ApiKeyCreateResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST /API/V2/AUTH/LOGIN

Log in a user

Description

Authenticates a user and returns tokens for subsequent authenticated requests.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "email": "derp@meme.org",
  "locale": null,
  "password": "string"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "description": "Mixin that automatically normalizes email fields.",
  "properties": {
    "email": {
      "format": "email",
      "title": "Email",
      "type": "string"
    },
    "locale": {
      "allOf": [
        {
          "$ref": "#/components/schemas/LocaleEnum"
        }
      ],
      "default": "en"
    },
    "password": {
      "title": "Password",
      "type": "string"
    }
  },
  "required": [
    "email",
    "password"
  ],
  "title": "LoginRequest",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "access_token": "string",
  "refresh_token": "string",
  "token_type": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "access_token": {
      "title": "Access Token",
      "type": "string"
    },
    "refresh_token": {
      "title": "Refresh Token",
      "type": "string"
    },
    "token_type": {
      "title": "Token Type",
      "type": "string"
    }
  },
  "required": [
    "access_token",
    "token_type",
    "refresh_token"
  ],
  "title": "LoginResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET

/API/V2/AUTH/PASSWORD_POLICY

Get the password policy

 **Description**

Returns the active password rules so clients can validate new passwords before submission.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses

200 OK

application/json

```
{
  "min_length": 0,
  "require_digit": true,
  "require_lowercase": true,
  "require_special": true,
  "require_uppercase": true,
  "special_characters": "string",
  "special_characters_pattern": "string"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "min_length": {
      "title": "Min Length",
      "type": "integer"
    },
    "require_digit": {
      "title": "Require Digit",
      "type": "boolean"
    },
    "require_lowercase": {
      "title": "Require Lowercase",
      "type": "boolean"
    },
    "require_special": {
      "title": "Require Special",
      "type": "boolean"
    },
    "require_uppercase": {
      "title": "Require Uppercase",
      "type": "boolean"
    },
    "special_characters": {
      "title": "Special Characters",
      "type": "string"
    },
    "special_characters_pattern": {
      "title": "Special Characters Pattern",
      "type": "string"
    }
  },
  "required": [
    "min_length",
    "require_digit",
    "require_special",
    "require_lowercase",
    "require_uppercase",
    "special_characters",
    "special_characters_pattern"
  ],
  "title": "PasswordPolicyResponse",
  "type": "object"
}
```

POST

/API/V2/AUTH/REFRESH_TOKEN

Refresh an access token

Description

Returns a new access token when the supplied refresh token is still valid.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "refresh_token": "string"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.




Schema of the request body

```
{
  "properties": {
    "refresh_token": {
      "title": "Refresh Token",
      "type": "string"
    }
  },
  "required": [
    "refresh_token"
  ],
  "title": "RefreshTokenRequest",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "access_token": "string",
  "refresh_token": "string",
  "token_type": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "access_token": {
      "title": "Access Token",
      "type": "string"
    },
    "refresh_token": {
      "title": "Refresh Token",
      "type": "string"
    },
    "token_type": {
      "title": "Token Type",
      "type": "string"
    }
  },
  "required": [
    "access_token",
    "refresh_token",
    "token_type"
  ],
  "title": "TokenResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST

/API/V2/AUTH/RESET_PASSWORD

Reset a password

 **Description**

Applies a new password using a valid password reset code.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.


Request body

application/json

```
{
  "email": "derp@meme.org",
  "locale": "string",
  "new_password": "string",
  "old_password": "string",
  "otp": "string"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

 **Schema of the request body**

```
{
  "description": "Mixin that automatically normalizes email fields.",
  "properties": {
    "email": {
      "format": "email",
      "title": "Email",
      "type": "string"
    },
    "locale": {
      "default": "en",
      "title": "Locale",
      "type": "string"
    },
    "new_password": {
      "title": "New Password",
      "type": "string"
    },
    "old_password": {
      "title": "Old Password",
      "type": "string"
    },
    "otp": {
      "title": "Otp",
      "type": "string"
    }
  },
  "required": [
    "email",
    "new_password"
  ],
  "title": "ResetPasswordRequest",
  "type": "object"
}
```

Responses


200 OK 422 Unprocessable Entity
application/json

Schema of the response body

```
[REDACTED]
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

PATCH /API/V2/AUTH/UPDATE_USER

Update a user

Description

Updates the username, role, or both for an existing NoyesStorage App user.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "email": "derp@meme.org",
  "role": "Super Admin",
  "user_id": 0,
  "username": "string"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "description": "Mixin that automatically normalizes email fields.",
  "properties": {
    "email": {
      "format": "email",
      "title": "Email",
      "type": "string"
    },
    "role": {
      "$ref": "#/components/schemas/RoleEnum"
    },
    "user_id": {
      "title": "User Id",
      "type": "integer"
    },
    "username": {
      "title": "Username",
      "type": "string"
    }
  },
  "title": "UserUpdateRequest",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "email": "derp@meme.org",
  "id": 0,
  "role": "Super Admin",
  "username": "string"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "email": {
      "format": "email",
      "title": "Email",
      "type": "string"
    },
    "id": {
      "title": "Id",
      "type": "integer"
    },
    "role": {
      "$ref": "#/components/schemas/RoleEnum"
    },
    "username": {
      "title": "Username",
      "type": "string"
    }
  },
  "required": [
    "id",
    "username",
    "email",
    "role"
  ],
  "title": "UserUpdateResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET

/API/V2/AUTH/USERS

List users

 **Description**

Returns paginated user accounts with optional search, sorting, and filters for administration.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.


Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>email__eq</code>	query	string		No	Filter by exact email.
<code>email__ilike</code>	query	string		No	Filter by partial, case-insensitive match on email.
<code>id__eq</code>	query	string		No	Filter by exact id.
<code>id__ilike</code>	query	string		No	Filter by partial, case-insensitive match on id.
<code>otp_expires_at__gte</code>	query	string		No	Filter by otp expires at greater than or equal to this value.
<code>otp_expires_at__lte</code>	query	string		No	Filter by otp expires at less than or equal to this value.
<code>page</code>	query	integer	1	No	Result page to return. Pagination starts at page=1.
<code>password_expires_at__gte</code>	query	string		No	Filter by password expires at greater than or equal to this value.
<code>password_expires_at__lte</code>	query	string		No	Filter by password expires at less than or equal to this value.
<code>role__eq</code>	query	string		No	Filter by exact role.
<code>role__in</code>	query	array		No	Filter by one or more values.
<code>search</code>	query	string		No	Search expression using supported field:value terms for this endpoint. Matching is partial and case-insensitive. Combine terms with uppercase AND/OR and one level of parentheses. Use *:value to search all supported fields. Supported fields: id, username, email, role.
<code>size</code>	query	integer	50	No	Maximum number of items to return on one page.
<code>sort_by</code>	query	string		No	Comma-separated sort fields. Use plain field or +field for ascending and -field for descending, for example created_at,-updated_at.
<code>username__eq</code>	query	string		No	Filter by exact username.
<code>username__ilike</code>	query	string		No	Filter by partial, case-insensitive match on username.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "items": [
    {
      "email": "derp@meme.org",
      "id": 0,
      "otp": "string",
      "otp_expires_at": "2022-04-13T15:42:05.901Z",
      "otp_expires_at_utc": "2022-04-13T15:42:05.901Z",
      "password_expires_at": "2022-04-13T15:42:05.901Z",
      "password_expires_at_utc": "2022-04-13T15:42:05.901Z",
      "role": "Super Admin",
      "username": "string"
    }
  ],
  "page": 0,
  "pages": 0,
  "size": 0,
  "total": 0
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "items": {
      "items": {
        "$ref": "#/components/schemas/UserSchema"
      },
      "title": "Items",
      "type": "array"
    },
    "page": {
      "minimum": 1.0,
      "title": "Page",
      "type": "integer"
    },
    "pages": {
      "minimum": 0.0,
      "title": "Pages",
      "type": "integer"
    },
    "size": {
      "minimum": 1.0,
      "title": "Size",
      "type": "integer"
    },
    "total": {
      "minimum": 0.0,
      "title": "Total",
      "type": "integer"
    }
  },
  "required": [
    "items",
    "total",
    "page",
    "size"
  ],
  "title": "Page[UserSchema]",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

System

GET /APIV2/BALCONIES

List balconies

Description

Returns the available picking-station balconies configured for the storage system.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses

200 OK
application/json

```
[
  {
    "id": 0
  }
]
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "items": {
    "$ref": "#/components/schemas/BalconyResponse"
  },
  "title": "Response Get Balconies Api V2 Balconies Get",
  "type": "array"
}
```

GET

/API/V2/LEVELS

List levels

 **Description**

Returns the storage levels configured in the database.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses


200 OK

application/json

```
[
  {
    "height": 0,
    "id": 0
  }
]
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

 **Schema of the response body**

```
{
  "items": {
    "$ref": "#/components/schemas/LevelResponse"
  },
  "title": "Response Get Levels From Db Api V2 Levels Get",
  "type": "array"
}
```

GET

/API/V2/REPORT

Generate a support report

 **Description**

Creates a diagnostic support report archive on the system and returns its location.

Input parameters


Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses

200 OK

application/json

```
{
  "report": "string",
  "report_path": "string",
  "status": "string",
  "tar_filename": "string"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Response model for report generation endpoint",
  "properties": {
    "report": {
      "default": "generated",
      "description": "Deprecated: Use status instead",
      "title": "Report",
      "type": "string"
    },
    "report_path": {
      "description": "Path where the report is saved",
      "title": "Report Path",
      "type": "string"
    },
    "status": {
      "description": "Status of report generation",
      "title": "Status",
      "type": "string"
    },
    "tar_filename": {
      "description": "Name of the generated tar file",
      "title": "Tar Filename",
      "type": "string"
    }
  },
  "required": [
    "status",
    "tar_filename",
    "report_path"
  ],
  "title": "ReportResponse",
  "type": "object"
}
```

POST /APIV2/RESTART_BRAIN

Restart the brain service

Description

Restarts the orchestration brain service. Active system state can be interrupted while the service restarts.


Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>timeout</code>	query		60	No	Maximum time in seconds to wait for the operation.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "brain_status": "string",
  "details": "string",
  "status": "string",
  "testing_script": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Standard response model for maintenance operations like starting/stopping brain or tests",
  "properties": {
    "brain_status": {
      "description": "Deprecated: Use status instead",
      "title": "Brain Status",
      "type": "string"
    },
    "details": {
      "title": "Details",
      "type": "string"
    },
    "status": {
      "title": "Status",
      "type": "string"
    },
    "testing_script": {
      "description": "Deprecated: Use status instead",
      "title": "Testing Script",
      "type": "string"
    }
  },
  "required": [
    "status",
    "details"
  ],
  "title": "StorageResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET

/API/V2/STORAGE_STATUS

Get storage status

 **Description**

Returns the current high-level storage status, such as RUNNING, PAUSED, STARTING_UP, MAINTENANCE, ERROR, OFF, or SHUTTING_DOWN.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses


200 OK

application/json

```
{
  "storage_status": "RUNNING"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

 **Schema of the response body**

```
{
  "properties": {
    "storage_status": {
      "$ref": "#/components/schemas/StorageStatus"
    }
  },
  "required": [
    "storage_status"
  ],
  "title": "StorageStatusResponse",
  "type": "object"
}
```

POST

/API/V2/SYSTEM/CHARGE_ALL_BOTS

Charge all bots

 **Description**

Creates charging work for all selected NoyesBots, typically before shutdown or maintenance.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "force": true
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the request body

```
{
  "properties": {
    "force": {
      "default": true,
      "description": "Force charge all bots.",
      "title": "Force",
      "type": "boolean"
    }
  },
  "title": "ChargeAllBotsRequest",
  "type": "object"
}
```


Responses

200 OK 422 Unprocessable Entity
application/json

Schema of the response body

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

Bot

DELETE /API/V2/BOT

Remove bots from the database

Description

Deletes selected NoyesBot records directly from the database. This is a maintenance action outside the normal offboarding flow and can require manual intervention by someone physically on site.


Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
bot_idx	query	array		No	⚠⚠⚠ DO NOT USE ⚠⚠⚠ Removes bots from database by passing a list of bot IDs; * for all bots

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "message": "string",
  "success": true
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Response model for bot removal operations.",
  "properties": {
    "message": {
      "title": "Message",
      "type": "string"
    },
    "success": {
      "title": "Success",
      "type": "boolean"
    }
  },
  "required": [
    "success",
    "message"
  ],
  "title": "BotRemoveResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET

/API/V2/BOTS

List bots

 **Description**

Returns the NoyesBots known to the system, optionally filtered by bot id, level, or charging state.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses

200 OK

application/json

```
[
  {
    "battery_level": 0,
    "id": 0,
    "is_charging": true,
    "level_id": 0,
    "lifted_carrier": 0
  }
]
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

 **Schema of the response body**

```
{
  "items": {
    "$ref": "#/components/schemas/BotResponse"
  },
  "title": "Response Get Bots Api V2 Bots Get",
  "type": "array"
}
```

Inventory

POST

/API/V2/BOX/SWAP-REGION

Swap region section codes

 **Description**

Atomically applies a set of section-code updates for a region swap so box placement metadata stays consistent.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "updates": [
    {
      "box_id": 0,
      "section_code": null
    }
  ]
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.




Schema of the request body

```
{
  "properties": {
    "updates": {
      "items": {
        "$ref": "#/components/schemas/BoxSectionCodeUpdate"
      },
      "title": "Updates",
      "type": "array"
    }
  },
  "required": [
    "updates"
  ],
  "title": "BoxRegionSwapRequest",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "boxes": [
    {
      "id": 0,
      "section_code": 0,
      "section_code_32": 0
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "boxes": {
      "items": {
        "$ref": "#/components/schemas/BoxResponse"
      },
      "title": "Boxes",
      "type": "array"
    }
  },
  "required": [
    "boxes"
  ],
  "title": "BoxRegionSwapResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST

/APIV2/BOX/SWAP/{SOURCE_BOX_ID}/{TARGET_BOX_ID}

Swap two box section codes

 **Description**

Atomically swaps section-code placement metadata between two existing boxes.


Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
source_box_id	path	integer		No	Source box id.
target_box_id	path	integer		No	Target box id.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "source_box": {
    "id": 0,
    "section_code": 0,
    "section_code_32": 0
  },
  "target_box": null
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "source_box": {
      "$ref": "#/components/schemas/BoxResponse"
    },
    "target_box": {
      "$ref": "#/components/schemas/BoxResponse"
    }
  },
  "required": [
    "source_box",
    "target_box"
  ],
  "title": "BoxSwapResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

DELETE /APIV2/BOX/{BOX_ID}

Delete a box

 **Description**

Deletes an empty box from its carrier and removes its box metadata from the inventory model.


Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
box_id	path	integer		No	Box id.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "id": 0,
  "section_code": 0,
  "section_code_32": 0
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "id": {
      "title": "Id",
      "type": "integer"
    },
    "section_code": {
      "description": "Legacy section code stored as integer with binary digits (8-bit semantics). String inputs are accepted for compatibility: <=8 chars (legacy), exactly 16 chars, or exactly 32 chars.",
      "title": "Section Code",
      "type": "integer"
    },
    "section_code_32": {
      "description": "Canonical 32-bit section code stored as integer with binary digits. This field is the source of truth for overlap/place-ment logic.",
      "title": "Section Code 32",
      "type": "integer"
    }
  },
  "required": [
    "section_code",
    "id"
  ],
  "title": "BoxResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

PATCH /API/V2/BOX/{BOX_ID}

Update a box

Description

Updates editable metadata for an existing box, such as section-code placement.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>box_id</code>	path	string		No	Box id.

Request body

application/json

```
{
  "section_code": null
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "properties": {
    "section_code": {
      "anyOf": [
        {
          "type": "string"
        },
        {
          "type": "integer"
        }
      ],
      "description": "Legacy section code stored as integer with binary digits (8-bit semantics). String inputs are accepted for compatibility: <=8 chars (legacy), exactly 16 chars, or exactly 32 chars.",
      "title": "Section Code"
    }
  },
  "title": "BoxPatch",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "id": 0,
  "section_code": 0,
  "section_code_32": 0
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "id": {
      "title": "Id",
      "type": "integer"
    },
    "section_code": {
      "description": "Legacy section code stored as integer with binary digits (8-bit semantics). String inputs are accepted for compatibility: <=8 chars (legacy), exactly 16 chars, or exactly 32 chars.",
      "title": "Section Code",
      "type": "integer"
    },
    "section_code_32": {
      "description": "Canonical 32-bit section code stored as integer with binary digits. This field is the source of truth for overlap/placement logic.",
      "title": "Section Code 32",
      "type": "integer"
    }
  },
  "required": [
    "section_code",
    "id"
  ],
  "title": "BoxResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST /API/V2/BOX/{CARRIER_ID}

Create a box on a carrier

Description

Creates a new box on the selected carrier with the supplied section-code placement metadata.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
carrier_id	path	string		No	Carrier id.

Request body

application/json

```
{
  "section_code": null
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the request body

```
{
  "properties": {
    "section_code": {
      "anyOf": [
        {
          "type": "string"
        },
        {
          "type": "integer"
        }
      ],
      "description": "Legacy section code stored as integer with binary digits (8-bit semantics). String inputs are accepted for compatibility: <=8 chars (legacy), exactly 16 chars, or exactly 32 chars.",
      "title": "Section Code"
    }
  },
  "title": "BoxCreate",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "id": 0,
  "section_code": 0,
  "section_code_32": 0
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "id": {
      "title": "Id",
      "type": "integer"
    },
    "section_code": {
      "description": "Legacy section code stored as integer with binary digits (8-bit semantics). String inputs are accepted for compatibility: <=8 chars (legacy), exactly 16 chars, or exactly 32 chars.",
      "title": "Section Code",
      "type": "integer"
    },
    "section_code_32": {
      "description": "Canonical 32-bit section code stored as integer with binary digits. This field is the source of truth for overlap/placement logic.",
      "title": "Section Code 32",
      "type": "integer"
    }
  },
  "required": [
    "section_code",
    "id"
  ],
  "title": "BoxResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET

/API/V2/INVENTORY_VIEW

List inventory

 **Description**

Returns the inventory view, including carriers, boxes, load information, and empty storage spaces with optional filters, search, sorting, and pagination.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>box_height_string_max_eq</code>	query	string		No	Filter by exact box height string max.
<code>box_height_string_max_ilike</code>	query	string		No	Filter by partial, case-insensitive match on box height string max.
<code>box_id_eq</code>	query	string		No	Filter by exact box id. Use 'none' for empty spaces.
<code>box_id_ilike</code>	query	string		No	Filter by partial, case-insensitive match on box id.
<code>box_size_string_eq</code>	query	string		No	Filter by exact box size string.
<code>box_size_string_ilike</code>	query	string		No	Filter by partial, case-insensitive match on box size string.
<code>box_size_string_in</code>	query	array		No	Filter by one or more values.
<code>carrier_id_eq</code>	query	string		No	Filter by exact carrier id.
<code>carrier_id_ilike</code>	query	string		No	Filter by partial, case-insensitive match on carrier id.
<code>content_codes_ilike</code>	query	string		No	Filter by partial, case-insensitive match on content codes.
<code>depth_eq</code>	query	integer		No	Filter by exact depth.
<code>depth_gte</code>	query	integer		No	Filter by depth greater than or equal to this value.
<code>depth_ilike</code>	query	string		No	Filter by partial, case-insensitive match on depth.
<code>depth_in</code>	query	array		No	Filter by one or more values.
<code>depth_lte</code>	query	integer		No	Filter by depth less than or equal to this value.
<code>eta_int_ilike</code>	query	string		No	Filter by partial, case-insensitive match on eta int.
<code>item_count_eq</code>	query	string		No	Filter by exact item count.
<code>item_count_ilike</code>	query	string		No	Filter by partial, case-insensitive match on item count.
<code>item_count_max_eq</code>	query	string		No	Filter by exact item count max.
<code>item_count_max_ilike</code>	query	string		No	Filter by partial, case-insensitive match on item count max.
<code>level_id_in</code>	query	array		No	Filter by one or more values.
<code>measurement_unit_in</code>	query	array		No	Filter by one or more values.
<code>page</code>	query	integer	1	No	Result page to return. Pagination starts at page=1.
<code>position_x_eq</code>	query	integer		No	Filter by exact position x.
<code>position_x_ilike</code>	query	string		No	Filter by partial, case-insensitive match on position x.
<code>position_y_eq</code>	query	integer		No	Filter by exact position y.


Parameter	In	Type	Default	Nullable	Description
<code>position_y__ilike</code>	query	string		No	Filter by partial, case-insensitive match on position y.
<code>search</code>	query	string		No	Search expression using supported field:value terms for this endpoint. Matching is partial and case-insensitive. Combine terms with uppercase AND/OR and one level of parentheses. Use *:value to search all supported fields. Supported fields: carrier_id, box_size_string, box_height_string_max, section_code, section_code_32, sku_id, sku_name.
<code>section_code_32__eq</code>	query	string		No	Filter by exact section code 32.
<code>section_code__eq</code>	query	string		No	Filter by exact section code.
<code>size</code>	query	integer	50	No	Maximum number of items to return on one page.
<code>sku_id__eq</code>	query	string		No	Filter by exact SKU id. Use 'none' for boxes without SKU load.
<code>sku_id__ilike</code>	query	string		No	Filter by partial, case-insensitive match on sku id.
<code>sku_name__ilike</code>	query	string		No	Filter by partial, case-insensitive match on sku name.
<code>sort_by</code>	query	string		No	Comma-separated sort fields. Use plain field or +field for ascending and -field for descending, for example created_at,-updated_at.
<code>system_quantity__eq</code>	query	string		No	Filter by exact system quantity.
<code>system_quantity__ilike</code>	query	string		No	Filter by partial, case-insensitive match on system quantity.
<code>updated_at__gte</code>	query	string		No	Filter by updated at greater than or equal to this value.
<code>updated_at__lte</code>	query	string		No	Filter by updated at less than or equal to this value.

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "items": [
    {
      "box_height_string_max": 0,
      "box_id": 0,
      "box_size": 0,
      "box_size_string": "string",
      "carrier_id": "string",
      "content_codes": "string",
      "depth": 0,
      "eta": 10.12,
      "eta_int": 0,
      "id": "string",
      "item_count": 0,
      "item_count_max": 0,
      "level_id": 0,
      "measurement_unit": "string",
      "on_empty_sku_action": "string",
      "position_x": 0,
      "position_y": 0,
      "section_code": "string",
      "section_code_32": 0,
      "sku_id": "string",
      "sku_name": "string",
      "system_quantity": 0,
      "updated_at": "2022-04-13T15:42:05.901Z",
      "updated_at_utc": "2022-04-13T15:42:05.901Z"
    }
  ],
  "page": 0,
  "pages": 0,
  "size": 0,
  "total": 0
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "items": {
      "items": {
        "$ref": "#/components/schemas/InventoryViewResponse"
      },
      "title": "Items",
      "type": "array"
    },
    "page": {
      "minimum": 1.0,
      "title": "Page",
      "type": "integer"
    },
    "pages": {
      "minimum": 0.0,
      "title": "Pages",
      "type": "integer"
    },
    "size": {
      "minimum": 1.0,
      "title": "Size",
      "type": "integer"
    },
    "total": {
      "minimum": 0.0,
      "title": "Total",
      "type": "integer"
    }
  },
  "required": [
    "items",
    "total",
    "page",
    "size"
  ],
  "title": "Page[InventoryViewResponse]",
  "type": "object"
}
```


application/json

```
{
```

```

    "detail": [
      {
        "loc": [
          null
        ],
        "msg": "string",
        "type": "string"
      }
    ]
  }
}

```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```

{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}

```

DELETE /API/V2/LOAD/{BOX_ID}

Delete load from a box

Description

Removes load information from the selected box and updates inventory quantities accordingly.


Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
box_id	path	integer		No	Box id.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "item_count": 0,
  "item_count_max": 0,
  "name": "string",
  "sku_id": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "item_count": {
      "title": "Item Count",
      "type": "integer"
    },
    "item_count_max": {
      "title": "Item Count Max",
      "type": "integer"
    },
    "name": {
      "title": "Name",
      "type": "string"
    },
    "sku_id": {
      "title": "Sku Id",
      "type": "string"
    }
  },
  "required": [
    "sku_id",
    "item_count",
    "item_count_max"
  ],
  "title": "LoadResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

PATCH /API/V2/LOAD/{BOX_ID}

Update load in a box

Description

Updates the SKU and quantity stored in the selected box.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
box_id	path	integer		No	Box id.

Request body

application/json

```
{
  "item_count": 0,
  "item_count_max": 0,
  "sku_id": "string"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "properties": {
    "item_count": {
      "title": "Item Count",
      "type": "integer"
    },
    "item_count_max": {
      "title": "Item Count Max",
      "type": "integer"
    },
    "sku_id": {
      "title": "Sku Id",
      "type": "string"
    }
  },
  "title": "LoadPatch",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "item_count": 0,
  "item_count_max": 0,
  "name": "string",
  "sku_id": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "item_count": {
      "title": "Item Count",
      "type": "integer"
    },
    "item_count_max": {
      "title": "Item Count Max",
      "type": "integer"
    },
    "name": {
      "title": "Name",
      "type": "string"
    },
    "sku_id": {
      "title": "Sku Id",
      "type": "string"
    }
  },
  "required": [
    "sku_id",
    "item_count",
    "item_count_max"
  ],
  "title": "LoadResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST /API/V2/LOAD/{BOX_ID}

Create load in a box

Description

Adds SKU load information to the selected box and updates inventory quantities accordingly.


Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
box_id	path	integer		No	Box id.

Request body

application/json

```
{
  "item_count": 0,
  "item_count_max": 0,
  "sku_id": "string"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the request body


```
{
  "properties": {
    "item_count": {
      "title": "Item Count",
      "type": "integer"
    },
    "item_count_max": {
      "title": "Item Count Max",
      "type": "integer"
    },
    "sku_id": {
      "title": "Sku Id",
      "type": "string"
    }
  },
  "required": [
    "sku_id",
    "item_count",
    "item_count_max"
  ],
  "title": "LoadCreate",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "item_count": 0,
  "item_count_max": 0,
  "name": "string",
  "sku_id": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "item_count": {
      "title": "Item Count",
      "type": "integer"
    },
    "item_count_max": {
      "title": "Item Count Max",
      "type": "integer"
    },
    "name": {
      "title": "Name",
      "type": "string"
    },
    "sku_id": {
      "title": "Sku Id",
      "type": "string"
    }
  },
  "required": [
    "sku_id",
    "item_count",
    "item_count_max"
  ],
  "title": "LoadResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET

/APIV2/EVENTS

List events

 **Description**

Returns paginated system events with optional filters, search, and sorting for diagnostics and integration monitoring.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.


Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>bot_id_eq</code>	query	string		No	Filter by exact bot id.
<code>component_eq</code>	query	string		No	Filter by exact component.
<code>component_ilike</code>	query	string		No	Filter by partial, case-insensitive match on component.
<code>component_in</code>	query	array		No	Filter by one or more values.
<code>created_at_gte</code>	query	string		No	Filter by created at greater than or equal to this value.
<code>created_at_lte</code>	query	string		No	Filter by created at less than or equal to this value.
<code>hex_value_eq</code>	query	string		No	Filter by exact hex value.
<code>hex_value_ilike</code>	query	string		No	Filter by partial, case-insensitive match on hex value.
<code>level_id_eq</code>	query			No	Filter by exact level id.
<code>page</code>	query	integer	1	No	Result page to return. Pagination starts at page=1.
<code>request_id_eq</code>	query	string		No	Filter by exact request id.
<code>request_id_ilike</code>	query	string		No	Filter by partial, case-insensitive match on request id.
<code>search</code>	query	string		No	Search expression using supported field:value terms for this endpoint. Matching is partial and case-insensitive. Combine terms with uppercase AND/OR and one level of parentheses. Use *.value to search all supported fields. Supported fields: description, component.
<code>severity_eq</code>	query	integer		No	Filter by exact severity.
<code>severity_gte</code>	query	integer		No	Filter by severity greater than or equal to this value.
<code>severity_in</code>	query	array		No	Filter by one or more values.
<code>severity_lte</code>	query	integer		No	Filter by severity less than or equal to this value.
<code>size</code>	query	integer	50	No	Maximum number of items to return on one page.
<code>sort_by</code>	query	string		No	Comma-separated sort fields. Use plain field or +field for ascending and -field for descending, for example created_at,-updated_at.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "items": [
    {
      "bot_id": "string",
      "component": "string",
      "created_at": "2022-04-13T15:42:05.901Z",
      "created_at_utc": "2022-04-13T15:42:05.901Z",
      "description": "string",
      "hex_value": "string",
      "level_id": 0,
      "request_id": "string",
      "severity": 0
    }
  ],
  "page": 0,
  "pages": 0,
  "size": 0,
  "total": 0
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "items": {
      "items": {
        "$ref": "#/components/schemas/EventsResponse"
      },
      "title": "Items",
      "type": "array"
    },
    "page": {
      "minimum": 1.0,
      "title": "Page",
      "type": "integer"
    },
    "pages": {
      "minimum": 0.0,
      "title": "Pages",
      "type": "integer"
    },
    "size": {
      "minimum": 1.0,
      "title": "Size",
      "type": "integer"
    },
    "total": {
      "minimum": 0.0,
      "title": "Total",
      "type": "integer"
    }
  },
  "required": [
    "items",
    "total",
    "page",
    "size"
  ],
  "title": "Page[EventsResponse]",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

Health

GET /APIV2/HEALTH

Check API health

Description

Returns OK when the API service is running and able to answer requests.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses

200 OK
application/json

Schema of the response body

Requests And Jobs

GET /APIV2/JOBS

List jobs

Description

Returns paginated jobs created from requests, with optional filters and sorting for operational tracking.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.


Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>balcony_id_eq</code>	query	integer		No	Filter by exact balcony id.
<code>created_at_gte</code>	query	string		No	Filter by created at greater than or equal to this value.
<code>created_at_lte</code>	query	string		No	Filter by created at less than or equal to this value.
<code>id_eq</code>	query			No	Filter by exact id.
<code>level_id_eq</code>	query	integer		No	Filter by exact level id.
<code>page</code>	query	integer	1	No	Result page to return. Pagination starts at page=1.
<code>request_id_eq</code>	query			No	Filter by exact request id.
<code>request_id_in</code>	query	array		No	Filter by one or more values.
<code>size</code>	query	integer	50	No	Maximum number of items to return on one page.
<code>sort_by</code>	query	string	- created_at	No	Comma-separated sort fields. Use plain field or +field for ascending and -field for descending, for example created_at,-updated_at.
<code>status_eq</code>	query	string		No	Filter by exact status.
<code>status_in</code>	query	array		No	Filter by one or more values.
<code>type_eq</code>	query	string		No	Filter by exact type.
<code>type_in</code>	query	array		No	Filter by one or more values.
<code>updated_at_gte</code>	query	string		No	Filter by updated at greater than or equal to this value.
<code>updated_at_lte</code>	query	string		No	Filter by updated at less than or equal to this value.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "items": [
    {
      "balcony_id": 0,
      "bot_id": 0,
      "box_id": 0,
      "carrier_id": "string",
      "created_at": "2022-04-13T15:42:05.901Z",
      "created_at_utc": "2022-04-13T15:42:05.901Z",
      "id": "string",
      "level_id": 0,
      "measurement_unit": "string",
      "on_empty_sku_action": "string",
      "quantity": 0,
      "request_id": "string",
      "sku_id": "string",
      "sku_name": "string",
      "status": "ACCEPTED",
      "type": "BUFFERING",
      "updated_at": "2022-04-13T15:42:05.901Z",
      "updated_at_utc": "2022-04-13T15:42:05.901Z"
    }
  ],
  "page": 0,
  "pages": 0,
  "size": 0,
  "total": 0
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the response body

```
{
  "properties": {
    "items": {
      "items": {
        "$ref": "#/components/schemas/JobResponse"
      },
      "title": "Items",
      "type": "array"
    },
    "page": {
      "minimum": 1.0,
      "title": "Page",
      "type": "integer"
    },
    "pages": {
      "minimum": 0.0,
      "title": "Pages",
      "type": "integer"
    },
    "size": {
      "minimum": 1.0,
      "title": "Size",
      "type": "integer"
    },
    "total": {
      "minimum": 0.0,
      "title": "Total",
      "type": "integer"
    }
  },
  "required": [
    "items",
    "total",
    "page",
    "size"
  ],
  "title": "Page[JobResponse]",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
    }
  ]
}
```

```
    "type": "string"
  }
]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET /API/V2/REQUESTS

List requests

Description

Returns paginated requests with optional filters and sorting for tracking storage work.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters


Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
counter__eq	query	integer		No	Filter by exact counter.
created_at__gte	query	string		No	Filter by created at greater than or equal to this value.
created_at__lte	query	string		No	Filter by created at less than or equal to this value.
id__eq	query			No	Filter by exact id.
id__in	query	array		No	Filter by one or more values.
page	query	integer	1	No	Result page to return. Pagination starts at page=1.
placing_pos__eq	query	integer		No	Filter by exact placing pos.
priority__eq	query	integer		No	Filter by exact priority.
size	query	integer	50	No	Maximum number of items to return on one page.
sort_by	query	string	- created_at	No	Comma-separated sort fields. Use plain field or +field for ascending and -field for descending, for example created_at,-updated_at.
status__eq	query	string		No	Filter by exact status.
status__in	query	array		No	Filter by one or more values.
type__eq	query	string		No	Filter by exact type.
type__in	query	array		No	Filter by one or more values.
updated_at__gte	query	string		No	Filter by updated at greater than or equal to this value.
updated_at__lte	query	string		No	Filter by updated at less than or equal to this value.

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "items": [
    {
      "bot_id": 0,
      "counter": 0,
      "created_at": "2022-04-13T15:42:05.901Z",
      "created_at_utc": "2022-04-13T15:42:05.901Z",
      "entities": [
        {
          "cancelled_quantity": 0,
          "completed_quantity": 0,
          "entity_type": "Sku",
          "id": "string",
          "job_type": "string",
          "succeeded_quantity": 0,
          "total_quantity": 0
        }
      ],
      "id": "string",
      "level_id": 0,
      "placing_pos": 0,
      "priority": 0,
      "status": "CREATED",
      "type": "FULFILLMENT",
      "updated_at": "2022-04-13T15:42:05.901Z",
      "updated_at_utc": "2022-04-13T15:42:05.901Z",
      "warnings": [
        {}
      ]
    }
  ],
  "page": 0,
  "pages": 0,
  "size": 0,
  "total": 0
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "items": {
      "items": {
        "$ref": "#/components/schemas/RequestResponseComposite"
      },
      "title": "Items",
      "type": "array"
    },
    "page": {
      "minimum": 1.0,
      "title": "Page",
      "type": "integer"
    },
    "pages": {
      "minimum": 0.0,
      "title": "Pages",
      "type": "integer"
    },
    "size": {
      "minimum": 1.0,
      "title": "Size",
      "type": "integer"
    },
    "total": {
      "minimum": 0.0,
      "title": "Total",
      "type": "integer"
    }
  },
  "required": [
    "items",
    "total",
    "page",
    "size"
  ],
  "title": "Page[RequestResponseComposite]",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST /APIV2/REQUESTS

Create a request

Description

Creates storage work such as fulfillment, replenishment, fetch, maintenance, onboarding, offboarding, pause, resume, charging, or uncharging requests.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json


Schema of the request body

```
{
  "anyOf": [
    {
      "$ref": "#/components/schemas/FetchRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/FulfillmentRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/ReplenishmentRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/OnboardingRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/OffboardingRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/RFIDWriteRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/PauseRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/ResumeRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/ChargingRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/UnchargingRequestCreateInput"
    },
    {
      "$ref": "#/components/schemas/CarrierOnboardingRequestCreateInput"
    }
  ],
  "title": "Request Data"
}
```

Responses

200 OK 400 Bad Request 422 Unprocessable Entity
application/json

```
{
  "bot_id": 0,
  "counter": 0,
  "created_at": "2022-04-13T15:42:05.901Z",
  "created_at_utc": "2022-04-13T15:42:05.901Z",
  "entities": [
    {
      "cancelled_quantity": 0,
      "completed_quantity": 0,
      "entity_type": "Sku",
      "id": "string",
      "job_type": "string",
      "succeeded_quantity": 0,
      "total_quantity": 0
    }
  ],
  "id": "string",
  "level_id": 0,
  "placing_pos": 0,
  "priority": 0,
  "status": "CREATED",
  "type": "FULFILLMENT",
  "updated_at": "2022-04-13T15:42:05.901Z",
  "updated_at_utc": "2022-04-13T15:42:05.901Z",
  "warnings": [
    {}
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Base for API responses with Berlin-local naive datetimes and explicit UTC *_utc fields.\n\nJSON encoding: naive fields use Berlin local wall time; aware fields use UTC (ISO 8601, +00:00).",
  "properties": {
    "bot_id": {
      "title": "Bot Id",
      "type": "integer"
    },
    "counter": {
      "title": "Counter",
      "type": "integer"
    },
    "created_at": {
      "format": "date-time",
      "title": "Created At",
      "type": "string"
    },
    "created_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Created At Utc",
      "type": "string"
    },
    "entities": {
      "items": {
        "$ref": "#/components/schemas/EntityResponse"
      },
      "title": "Entities",
      "type": "array"
    },
    "id": {
      "title": "Id",
      "type": "string"
    },
    "level_id": {
      "title": "Level Id",
      "type": "integer"
    },
    "placing_pos": {
      "title": "Placing Pos",
      "type": "integer"
    },
    "priority": {
      "title": "Priority",
      "type": "integer"
    }
  }
}
```

application/json


```
{
  "detail": "Request abc123 was aborted due to validation errors: SKU001: 0x5082 - Not enough quantity; CARRIER002: 0x5083 - Carrier does not exist",
  "status": 400
}
```

```
"entity_reedback": [
  {
    "entity_id": "SKU001",
    "entity_type": "Sku",
    "error_code": "0x5082",
    "human_string": "Not enough quantity"
  },
  {
    "entity_id": "CARRIER002",
    "entity_type": "Carrier",
    "error_code": "0x5083",
    "human_string": "Carrier does not exist"
  }
],
"request_id": "abc123"
}
```

Schema of the response body

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST /API/V2/TRIGGER/{JOB_ID}

Trigger a job

Description

Sends a trigger for the selected job so the waiting job can continue execution.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
job_id	path	string		No	Which job to trigger

Request body

application/json

```
{
  "TriggerStatus": null
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.



Schema of the request body

```
{
  "description": "Schema for job trigger requests",
  "properties": {
    "TriggerStatus": {
      "allOf": [
        {
          "$ref": "#/components/schemas/TriggerStatus"
        }
      ],
      "default": "SUCCEEDED_TRIGGER"
    }
  },
  "title": "TriggerRequest",
  "type": "object"
}
```

Responses


200 OK 422 Unprocessable Entity
application/json

Schema of the response body

```
{
  "description": "Schema for job trigger responses",
  "properties": {},
  "title": "TriggerResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

Settings

GET /API/V2/SETTINGS

List settings

Description

Returns all system settings visible to the requesting user. Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters


Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses

200 OK

application/json

```
{
  "items": [
    {
      "default_value_json": null,
      "enum_values": [
        "string"
      ],
      "key": "string",
      "required_role": [
        "string"
      ],
      "restart_required": true,
      "scope": "string",
      "user_idx": 0,
      "value_json": null,
      "value_type": "string"
    }
  ],
  "total": 0
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Schema for list of settings response.",
  "properties": {
    "items": {
      "items": {
        "$ref": "#/components/schemas/SettingListItemResponse"
      },
      "title": "Items",
      "type": "array"
    },
    "total": {
      "title": "Total",
      "type": "integer"
    }
  },
  "required": [
    "items",
    "total"
  ],
  "title": "SettingsListResponse",
  "type": "object"
}
```

POST /API/V2/SETTINGS

Create or update a setting

Description

Creates a system setting, or updates the existing setting when the key already exists. Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "default_value_json": null,
  "key": "string",
  "required_role": [
    "string"
  ],
  "restart_required": true,
  "value_json": null
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "description": "Schema for creating a new setting.",
  "properties": {
    "default_value_json": {
      "title": "Default Value Json"
    },
    "key": {
      "title": "Key",
      "type": "string"
    },
    "required_role": {
      "items": {
        "type": "string"
      },
      "title": "Required Role",
      "type": "array"
    },
    "restart_required": {
      "default": false,
      "title": "Restart Required",
      "type": "boolean"
    },
    "value_json": {
      "title": "Value Json"
    }
  },
  "required": [
    "key"
  ],
  "title": "SettingCreate",
  "type": "object"
}
```

Responses

201 Created 422 Unprocessable Entity
application/json

```
{
  "created_at": "2022-04-13T15:42:05.901Z",
  "created_at_utc": "2022-04-13T15:42:05.901Z",
  "default_value_json": null,
  "id": "8d84fa7f-d014-4cfd-8169-fc12244bde8a",
  "key": "string",
  "required_role": [
    "string"
  ],
  "restart_required": true,
  "updated_at": "2022-04-13T15:42:05.901Z",
  "updated_at_utc": "2022-04-13T15:42:05.901Z",
  "user_idx": 0,
  "value_json": null
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the response body

```
{
  "description": "Schema for setting response.",
  "properties": {
    "created_at": {
      "format": "date-time",
      "title": "Created At",
      "type": "string"
    },
    "created_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Created At Utc",
      "type": "string"
    },
    "default_value_json": {
      "title": "Default Value Json"
    },
    "id": {
      "format": "uuid",
      "title": "Id",
      "type": "string"
    },
    "key": {
      "title": "Key",
      "type": "string"
    },
    "required_role": {
      "items": {
        "type": "string"
      },
      "title": "Required Role",
      "type": "array"
    },
    "restart_required": {
      "default": false,
      "title": "Restart Required",
      "type": "boolean"
    },
    "updated_at": {
      "format": "date-time",
      "title": "Updated At",
      "type": "string"
    },
    "updated_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the"
    }
  }
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET /API/V2/SETTINGS/SYSTEM

Get system settings

Description

Returns resolved system settings for app-facing keys.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Responses

200 OK
application/json

```
{
  "settings": {}
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Resolved settings map for app-facing keys.",
  "properties": {
    "settings": {
      "title": "Settings",
      "type": "object"
    }
  },
  "required": [
    "settings"
  ],
  "title": "ResolvedSettingsResponse",
  "type": "object"
}
```

PATCH /API/V2/SETTINGS/SYSTEM

Patch system settings

 **Description**

Applies update/reset operations to app-facing system settings.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.


Request body

application/json

```
{
  "resets": [
    "string"
  ],
  "updates": {}
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


 **Schema of the request body**

```
{
  "description": "Batch update/reset payload for scoped settings.",
  "properties": {
    "resets": {
      "items": {
        "type": "string"
      },
      "title": "Resets",
      "type": "array"
    },
    "updates": {
      "title": "Updates",
      "type": "object"
    }
  },
  "title": "SettingsPatchRequest",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{  
  "settings": {}  
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{  
  "description": "Resolved settings map for app-facing keys.",  
  "properties": {  
    "settings": {  
      "title": "Settings",  
      "type": "object"  
    }  
  },  
  "required": [  
    "settings"  
  ],  
  "title": "ResolvedSettingsResponse",  
  "type": "object"  
}
```

application/json

```
{  
  "detail": [  
    {  
      "loc": [  
        null  
      ],  
      "msg": "string",  
      "type": "string"  
    }  
  ]  
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{  
  "properties": {  
    "detail": {  
      "items": {  
        "$ref": "#/components/schemas/ValidationError"  
      },  
      "title": "Detail",  
      "type": "array"  
    }  
  },  
  "title": "HTTPValidationError",  
  "type": "object"  
}
```

GET

/API/V2/SETTINGS/USERS/{USER_IDX}

Get user settings

 **Description**

Returns resolved user-scoped settings for a user.


Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>user_idx</code>	path	integer		No	

Responses

200 OK 422 Unprocessable Entity
application/json

```
{  
  "settings": {}  
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{  
  "description": "Resolved settings map for app-facing keys.",  
  "properties": {  
    "settings": {  
      "title": "Settings",  
      "type": "object"  
    }  
  },  
  "required": [  
    "settings"  
  ],  
  "title": "ResolvedSettingsResponse",  
  "type": "object"  
}
```

application/json

```
{  
  "detail": [  
    {  
      "loc": [  
        null  
      ],  
      "msg": "string",  
      "type": "string"  
    }  
  ]  
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{  
  "properties": {  
    "detail": {  
      "items": {  
        "$ref": "#/components/schemas/ValidationError"  
      },  
      "title": "Detail",  
      "type": "array"  
    }  
  },  
  "title": "HTTPValidationError",  
  "type": "object"  
}
```

PATCH /API/V2/SETTINGS/USERS/{USER_IDX}

Patch user settings

Description

Applies update/reset operations to user-scoped settings.


Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>user_idx</code>	path	integer		No	

Request body

application/json

```
{
  "resets": [
    "string"
  ],
  "updates": {}
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "description": "Batch update/reset payload for scoped settings.",
  "properties": {
    "resets": {
      "items": {
        "type": "string"
      },
      "title": "Resets",
      "type": "array"
    },
    "updates": {
      "title": "Updates",
      "type": "object"
    }
  },
  "title": "SettingsPatchRequest",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{  
  "settings": {}  
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{  
  "description": "Resolved settings map for app-facing keys.",  
  "properties": {  
    "settings": {  
      "title": "Settings",  
      "type": "object"  
    }  
  },  
  "required": [  
    "settings"  
  ],  
  "title": "ResolvedSettingsResponse",  
  "type": "object"  
}
```

application/json

```
{  
  "detail": [  
    {  
      "loc": [  
        null  
      ],  
      "msg": "string",  
      "type": "string"  
    }  
  ]  
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{  
  "properties": {  
    "detail": {  
      "items": {  
        "$ref": "#/components/schemas/ValidationError"  
      },  
      "title": "Detail",  
      "type": "array"  
    }  
  },  
  "title": "HTTPValidationError",  
  "type": "object"  
}
```

DELETE /API/V2/SETTINGS/{KEY}

Reset a setting

Description

Deletes the stored value for a system setting so the system falls back to its default behavior.

Input parameters

Parameter	In	Type	Default	Nullable	Description
BearerAuth	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
key	path	string		No	Setting key.

Responses

200 OK 422 Unprocessable Entity
application/json

Schema of the response body

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

GET /API/V2/SETTINGS/{KEY}

Get a setting

Description

Returns the current value and metadata for one system setting. Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters


Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>key</code>	path	string		No	Setting key.

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "created_at": "2022-04-13T15:42:05.901Z",
  "created_at_utc": "2022-04-13T15:42:05.901Z",
  "default_value_json": null,
  "id": "f7ff9d51-ccb7-4326-9703-92c446317834",
  "key": "string",
  "required_role": [
    "string"
  ],
  "restart_required": true,
  "updated_at": "2022-04-13T15:42:05.901Z",
  "updated_at_utc": "2022-04-13T15:42:05.901Z",
  "user_idx": 0,
  "value_json": null
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Schema for setting response.",
  "properties": {
    "created_at": {
      "format": "date-time",
      "title": "Created At",
      "type": "string"
    },
    "created_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Created At Utc",
      "type": "string"
    },
    "default_value_json": {
      "title": "Default Value Json"
    },
    "id": {
      "format": "uuid",
      "title": "Id",
      "type": "string"
    },
    "key": {
      "title": "Key",
      "type": "string"
    },
    "required_role": {
      "items": {
        "type": "string"
      },
      "title": "Required Role",
      "type": "array"
    },
    "restart_required": {
      "default": false,
      "title": "Restart Required",
      "type": "boolean"
    },
    "updated_at": {
      "format": "date-time",
      "title": "Updated At",
      "type": "string"
    },
    "updated_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the"
    }
  }
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

⚠ This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

PATCH /APIV2/SETTINGS/{KEY}

Update a setting

Description

Updates the value of one system setting. The new value can change system behavior immediately. Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>key</code>	path	string		No	Setting key.

Request body

application/json

```
{
  "value_json": null
}
```

⚠ This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the request body

```
{
  "description": "Schema for updating a setting value.",
  "properties": {
    "value_json": {
      "title": "Value Json"
    }
  },
  "title": "SettingUpdate",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "created_at": "2022-04-13T15:42:05.901Z",
  "created_at_utc": "2022-04-13T15:42:05.901Z",
  "default_value_json": null,
  "id": "5c4da9a9-1267-48a1-b102-655070e6efba",
  "key": "string",
  "required_role": [
    "string"
  ],
  "restart_required": true,
  "updated_at": "2022-04-13T15:42:05.901Z",
  "updated_at_utc": "2022-04-13T15:42:05.901Z",
  "user_idx": 0,
  "value_json": null
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Schema for setting response.",
  "properties": {
    "created_at": {
      "format": "date-time",
      "title": "Created At",
      "type": "string"
    },
    "created_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Created At Utc",
      "type": "string"
    },
    "default_value_json": {
      "title": "Default Value Json"
    },
    "id": {
      "format": "uuid",
      "title": "Id",
      "type": "string"
    },
    "key": {
      "title": "Key",
      "type": "string"
    },
    "required_role": {
      "items": {
        "type": "string"
      },
      "title": "Required Role",
      "type": "array"
    },
    "restart_required": {
      "default": false,
      "title": "Restart Required",
      "type": "boolean"
    },
    "updated_at": {
      "format": "date-time",
      "title": "Updated At",
      "type": "string"
    },
    "updated_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the"
    }
  }
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

⚠ This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

SKU

POST /API/V2/SKU

Create an SKU

Description

Creates SKU master data used for inventory, fulfillment, and replenishment.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

application/json

```
{
  "id": "string",
  "measurement_unit": null,
  "name": "string",
  "on_empty_sku_action": null
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.



Schema of the request body

```
{
  "properties": {
    "id": {
      "title": "Id",
      "type": "string"
    },
    "measurement_unit": {
      "allOf": [
        {
          "$ref": "#/components/schemas/MeasurementUnit"
        }
      ],
      "default": "PIECE"
    },
    "name": {
      "title": "Name",
      "type": "string"
    },
    "on_empty_sku_action": {
      "allOf": [
        {
          "$ref": "#/components/schemas/OnEmptySKUAction"
        }
      ],
      "default": "UNASSIGN"
    }
  },
  "required": [
    "id"
  ],
  "title": "Sku",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "carrier_utilization": 10.12,
  "created_at": "2022-04-13T15:42:05.901Z",
  "created_at_utc": "2022-04-13T15:42:05.901Z",
  "deleted_at": "2022-04-13T15:42:05.901Z",
  "deleted_at_utc": "2022-04-13T15:42:05.901Z",
  "fulfillable": 0,
  "id": "string",
  "image_url": "string",
  "load_count": 0,
  "max_quantity": 0,
  "measurement_unit": null,
  "name": "string",
  "on_empty_sku_action": null,
  "replenishable": 0,
  "total_negative_inventory_bookings": 0,
  "total_positive_inventory_bookings": 0,
  "total_quantity": 0,
  "updated_at": "2022-04-13T15:42:05.901Z",
  "updated_at_utc": "2022-04-13T15:42:05.901Z"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the response body

```
{
  "description": "Base for API responses with Berlin-local naive datetimes and explicit UTC *_utc fields.\n\nJSON encoding: naive fields use Berlin local wall time; aware fields use UTC (ISO 8601, +00:00).",
  "properties": {
    "carrier_utilization": {
      "title": "Carrier Utilization",
      "type": "number"
    },
    "created_at": {
      "format": "date-time",
      "title": "Created At",
      "type": "string"
    },
    "created_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Created At Utc",
      "type": "string"
    },
    "deleted_at": {
      "format": "date-time",
      "title": "Deleted At",
      "type": "string"
    },
    "deleted_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Deleted At Utc",
      "type": "string"
    },
    "fulfillable": {
      "title": "Fulfillable",
      "type": "integer"
    },
    "id": {
      "title": "Id",
      "type": "string"
    },
    "image_url": {
      "title": "Image Url",
      "type": "string"
    },
    "load_count": {
      "title": "Load Count"
    }
  }
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

```
}
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

POST /API/V2/SKU/IMAGE

Upload an SKU image

Description

Uploads an image for an SKU to the local file server and records it for use by storage workflows and the UI.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>overwrite_existing_image</code>	query	boolean	False	No	If true, replaces an existing image if one exists.

Request body

multipart/form-data

```
{
  "picture": "TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQ=",
  "sku_id": "string"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the request body

```
{
  "properties": {
    "picture": {
      "format": "binary",
      "title": "Picture",
      "type": "string"
    },
    "sku_id": {
      "description": "The Sku ID of the SKU e.g. 'H000123'",
      "title": "Sku ID",
      "type": "string"
    }
  },
  "required": [
    "sku_id",
    "picture"
  ],
  "title": "Body_upload_image_api_v2_sku_image_post",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "carrier_utilization": 10.12,
  "created_at": "2022-04-13T15:42:05.901Z",
  "created_at_utc": "2022-04-13T15:42:05.901Z",
  "deleted_at": "2022-04-13T15:42:05.901Z",
  "deleted_at_utc": "2022-04-13T15:42:05.901Z",
  "fulfillable": 0,
  "id": "string",
  "image_url": "string",
  "load_count": 0,
  "max_quantity": 0,
  "measurement_unit": null,
  "name": "string",
  "on_empty_sku_action": null,
  "replenishable": 0,
  "total_negative_inventory_bookings": 0,
  "total_positive_inventory_bookings": 0,
  "total_quantity": 0,
  "updated_at": "2022-04-13T15:42:05.901Z",
  "updated_at_utc": "2022-04-13T15:42:05.901Z"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the response body

```
{
  "description": "Base for API responses with Berlin-local naive datetimes and explicit UTC *_utc fields.\n\nJSON encoding: naive fields use Berlin local wall time; aware fields use UTC (ISO 8601, +00:00).",
  "properties": {
    "carrier_utilization": {
      "title": "Carrier Utilization",
      "type": "number"
    },
    "created_at": {
      "format": "date-time",
      "title": "Created At",
      "type": "string"
    },
    "created_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Created At Utc",
      "type": "string"
    },
    "deleted_at": {
      "format": "date-time",
      "title": "Deleted At",
      "type": "string"
    },
    "deleted_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Deleted At Utc",
      "type": "string"
    },
    "fulfillable": {
      "title": "Fulfillable",
      "type": "integer"
    },
    "id": {
      "title": "Id",
      "type": "string"
    },
    "image_url": {
      "title": "Image Url",
      "type": "string"
    },
    "load_count": {
      "title": "Load Count"
    }
  }
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

```
}
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

DELETE /API/V2/SKU/{ID}

Delete an SKU

Description

Deletes SKU master data by id. Existing inventory or active work can prevent deletion.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>id</code>	path	string		No	Resource id.

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "carrier_utilization": 10.12,
  "created_at": "2022-04-13T15:42:05.901Z",
  "created_at_utc": "2022-04-13T15:42:05.901Z",
  "deleted_at": "2022-04-13T15:42:05.901Z",
  "deleted_at_utc": "2022-04-13T15:42:05.901Z",
  "fulfillable": 0,
  "id": "string",
  "image_url": "string",
  "load_count": 0,
  "max_quantity": 0,
  "measurement_unit": null,
  "name": "string",
  "on_empty_sku_action": null,
  "replenishable": 0,
  "total_negative_inventory_bookings": 0,
  "total_positive_inventory_bookings": 0,
  "total_quantity": 0,
  "updated_at": "2022-04-13T15:42:05.901Z",
  "updated_at_utc": "2022-04-13T15:42:05.901Z"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the response body

```
{
  "description": "Base for API responses with Berlin-local naive datetimes and explicit UTC *_utc fields.\n\nJSON encoding: naive fields use Berlin local wall time; aware fields use UTC (ISO 8601, +00:00).",
  "properties": {
    "carrier_utilization": {
      "title": "Carrier Utilization",
      "type": "number"
    },
    "created_at": {
      "format": "date-time",
      "title": "Created At",
      "type": "string"
    },
    "created_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Created At Utc",
      "type": "string"
    },
    "deleted_at": {
      "format": "date-time",
      "title": "Deleted At",
      "type": "string"
    },
    "deleted_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Deleted At Utc",
      "type": "string"
    },
    "fulfillable": {
      "title": "Fulfillable",
      "type": "integer"
    },
    "id": {
      "title": "Id",
      "type": "string"
    },
    "image_url": {
      "title": "Image Url",
      "type": "string"
    },
    "load_count": {
      "title": "Load Count"
    }
  }
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

```
}
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

PATCH /API/V2/SKU/{ID}

Update an SKU

Description

Updates SKU master data such as name, unit, or handling behavior.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>id</code>	path	string		No	Resource id.

Request body

application/json

```
{
  "measurement_unit": "PIECE",
  "name": "string",
  "on_empty_sku_action": "DO_NOTHING"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the request body


```
{
  "properties": {
    "measurement_unit": {
      "$ref": "#/components/schemas/MeasurementUnit"
    },
    "name": {
      "title": "Name",
      "type": "string"
    },
    "on_empty_sku_action": {
      "$ref": "#/components/schemas/OnEmptySKUAction"
    }
  },
  "title": "SkuPatch",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity

application/json

```
{
  "carrier_utilization": 10.12,
  "created_at": "2022-04-13T15:42:05.901Z",
  "created_at_utc": "2022-04-13T15:42:05.901Z",
  "deleted_at": "2022-04-13T15:42:05.901Z",
  "deleted_at_utc": "2022-04-13T15:42:05.901Z",
  "fulfillable": 0,
  "id": "string",
  "image_url": "string",
  "load_count": 0,
  "max_quantity": 0,
  "measurement_unit": null,
  "name": "string",
  "on_empty_sku_action": null,
  "replenishable": 0,
  "total_negative_inventory_bookings": 0,
  "total_positive_inventory_bookings": 0,
  "total_quantity": 0,
  "updated_at": "2022-04-13T15:42:05.901Z",
  "updated_at_utc": "2022-04-13T15:42:05.901Z"
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.


Schema of the response body

```
{
  "description": "Base for API responses with Berlin-local naive datetimes and explicit UTC *_utc fields.\n\nJSON encoding: naive fields use Berlin local wall time; aware fields use UTC (ISO 8601, +00:00).",
  "properties": {
    "carrier_utilization": {
      "title": "Carrier Utilization",
      "type": "number"
    },
    "created_at": {
      "format": "date-time",
      "title": "Created At",
      "type": "string"
    },
    "created_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Created At Utc",
      "type": "string"
    },
    "deleted_at": {
      "format": "date-time",
      "title": "Deleted At",
      "type": "string"
    },
    "deleted_at_utc": {
      "description": "Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the *_utc suffix is kept for backward compatibility and will be deprecated in a future release.",
      "format": "date-time",
      "title": "Deleted At Utc",
      "type": "string"
    },
    "fulfillable": {
      "title": "Fulfillable",
      "type": "integer"
    },
    "id": {
      "title": "Id",
      "type": "string"
    },
    "image_url": {
      "title": "Image Url",
      "type": "string"
    },
    "load_count": {
      "title": "Load Count"
    }
  }
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

```
}  
  ]  
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{  
  "properties": {  
    "detail": {  
      "items": {  
        "$ref": "#/components/schemas/ValidationError"  
      },  
      "title": "Detail",  
      "type": "array"  
    }  
  },  
  "title": "HTTPValidationError",  
  "type": "object"  
}
```

GET /API/V2/SKUS

List SKUs

Description

Returns paginated SKU master data with optional filters, search, sorting, and inventory-derived quantities.

Timestamp responses include both legacy timestamp fields and matching fields with the `_utc` suffix. New integrations should use the `_utc` fields because they are ISO 8601 timestamps with an explicit UTC offset. Fields without the `_utc` suffix are kept for backward compatibility and will be deprecated in a future release.

Input parameters


Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.
<code>carrier_utilization__ilike</code>	query	number		No	Filter by partial, case-insensitive match on carrier utilization.
<code>created_at__gte</code>	query	string		No	Filter by created at greater than or equal to this value.
<code>created_at__lte</code>	query	string		No	Filter by created at less than or equal to this value.
<code>deleted_at__gte</code>	query	string		No	Filter by deleted at greater than or equal to this value.
<code>deleted_at__lte</code>	query	string		No	Filter by deleted at less than or equal to this value.
<code>fulfillable__ilike</code>	query	integer		No	Filter by partial, case-insensitive match on fulfillable.
<code>id_eq</code>	query	string		No	Filter by exact id.
<code>id__ilike</code>	query	string		No	Filter by partial, case-insensitive match on id.
<code>load_count_eq</code>	query	integer		No	Filter by exact load count.
<code>load_count__ilike</code>	query	integer		No	Filter by partial, case-insensitive match on load count.
<code>max_quantity_eq</code>	query	integer		No	Filter by exact max quantity.
<code>max_quantity__gte</code>	query	integer		No	Filter by max quantity greater than or equal to this value.
<code>max_quantity__ilike</code>	query	integer		No	Filter by partial, case-insensitive match on max quantity.
<code>max_quantity__lte</code>	query	integer		No	Filter by max quantity less than or equal to this value.
<code>measurement_unit__in</code>	query	array		No	Filter by one or more values.
<code>name__ilike</code>	query	string		No	Filter by partial, case-insensitive match on name.
<code>on_empty_sku_action__in</code>	query	array		No	Filter by one or more values.
<code>page</code>	query	integer	1	No	Result page to return. Pagination starts at page=1.
<code>replenishable__ilike</code>	query	integer		No	Filter by partial, case-insensitive match on replenishable.
<code>search</code>	query	string		No	Search expression using supported field:value terms for this endpoint. Matching is partial and case-insensitive. Combine terms with uppercase AND/OR and one level of parentheses. Use *:value to search all supported fields. Supported fields: id, name, carrier_utilization.
<code>size</code>	query	integer	50	No	Maximum number of items to return on one page.

Parameter	In	Type	Default	Nullable	Description
<code>sort_by</code>	query	string		No	Comma-separated sort fields. Use plain field or +field for ascending and -field for descending, for example created_at,-updated_at.
<code>total_quantity__eq</code>	query	integer		No	Filter by exact total quantity.
<code>total_quantity__gte</code>	query	integer		No	Filter by total quantity greater than or equal to this value.
<code>total_quantity__ilike</code>	query	integer		No	Filter by partial, case-insensitive match on total quantity.
<code>total_quantity__lte</code>	query	integer		No	Filter by total quantity less than or equal to this value.
<code>updated_at__gte</code>	query	string		No	Filter by updated at greater than or equal to this value.
<code>updated_at__lte</code>	query	string		No	Filter by updated at less than or equal to this value.

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "items": [
    {
      "carrier_utilization": 10.12,
      "created_at": "2022-04-13T15:42:05.901Z",
      "created_at_utc": "2022-04-13T15:42:05.901Z",
      "deleted_at": "2022-04-13T15:42:05.901Z",
      "deleted_at_utc": "2022-04-13T15:42:05.901Z",
      "fulfillable": 0,
      "id": "string",
      "image_url": "string",
      "load_count": 0,
      "max_quantity": 0,
      "measurement_unit": null,
      "name": "string",
      "on_empty_sku_action": null,
      "replenishable": 0,
      "total_negative_inventory_bookings": 0,
      "total_positive_inventory_bookings": 0,
      "total_quantity": 0,
      "updated_at": "2022-04-13T15:42:05.901Z",
      "updated_at_utc": "2022-04-13T15:42:05.901Z"
    }
  ],
  "page": 0,
  "pages": 0,
  "size": 0,
  "total": 0
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body


```
{
  "properties": {
    "items": {
      "items": {
        "$ref": "#/components/schemas/SkuResponse"
      },
      "title": "Items",
      "type": "array"
    },
    "page": {
      "minimum": 1.0,
      "title": "Page",
      "type": "integer"
    },
    "pages": {
      "minimum": 0.0,
      "title": "Pages",
      "type": "integer"
    },
    "size": {
      "minimum": 1.0,
      "title": "Size",
      "type": "integer"
    },
    "total": {
      "minimum": 0.0,
      "title": "Total",
      "type": "integer"
    }
  },
  "required": [
    "items",
    "total",
    "page",
    "size"
  ],
  "title": "Page[SkuResponse]",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],

```

```
    "msg": "string",
    "type": "string"
  }
]
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

Support

POST /API/V2/SUPPORT/CREATE_ISSUE

Create a support issue

Description

Creates a support issue from multipart form data, with a JSON payload and optional file attachments.

Input parameters

Parameter	In	Type	Default	Nullable	Description
<code>BearerAuth</code>	header	string	N/A	No	Bearer token authentication. Use the API key or token issued for your integration.

Request body

multipart/form-data

```
{
  "files": [
    "TG9yZW0gaXBzdW0gZG9sb3Igc2l0IGFtZXQ="
  ],
  "payload": "string"
}
```



This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.




Schema of the request body

```
{
  "properties": {
    "files": {
      "items": {
        "format": "binary",
        "type": "string"
      },
      "title": "Files",
      "type": "array"
    },
    "payload": {
      "description": "JSON stringified CreateIssuePayload",
      "title": "Payload",
      "type": "string"
    }
  },
  "required": [
    "payload"
  ],
  "title": "Body_create_issue_api_v2_support_create_issue_post",
  "type": "object"
}
```

Responses

200 OK 422 Unprocessable Entity
application/json

```
{
  "key": "string",
  "url": "string"
}
```


 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "description": "Response model for issue creation endpoint",
  "properties": {
    "key": {
      "title": "Key",
      "type": "string"
    },
    "url": {
      "title": "Url",
      "type": "string"
    }
  },
  "required": [
    "key",
    "url"
  ],
  "title": "CreateIssueResponse",
  "type": "object"
}
```

application/json

```
{
  "detail": [
    {
      "loc": [
        null
      ],
      "msg": "string",
      "type": "string"
    }
  ]
}
```

 This example has been generated automatically from the schema and it is not accurate. Refer to the schema for more information.

Schema of the response body

```
{
  "properties": {
    "detail": {
      "items": {
        "$ref": "#/components/schemas/ValidationError"
      },
      "title": "Detail",
      "type": "array"
    }
  },
  "title": "HTTPValidationError",
  "type": "object"
}
```

Schemas

APIKEYCREATEINPUT

Name	Type	Description
name	string	

APIKEYCREATERESPONSE

Name	Type	Description
api_key	string	

APIKEYDELETERESPONSE

Name	Type	Description
deleted_key	string	
deleted_key_id	integer	
deleted_key_name	string	
message	string	

APIKEYINFO

Name	Type	Description
api_key	string	
expires_at	string(date-time)	
expires_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
id	integer	
last_used	string(date-time)	
last_used_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
name	string	
role	RoleEnum	

AUTHENTICATEDSUBJECT

Name	Type	Description
email	string(email)	
id	integer	
role	string	
username	string	

AUTHENTICATERESPONSE

Name	Type	Description
settings		
user	AuthenticatedSubject	

BALCONYRESPONSE

Name	Type	Description
id	integer	

BODY_CREATE_ISSUE_API_V2_SUPPORT_CREATE_ISSUE_POST

Name	Type	Description
files	Array<string(binary)>	
payload	string	JSON stringified CreateIssuePayload

BODY_UPLOAD_IMAGE_API_V2_SKU_IMAGE_POST

Name	Type	Description
picture	string(binary)	
sku_id	string	The Sku ID of the SKU e.g. 'H000123'

BOTREMOVERESPONSE

Name	Type	Description
message	string	
success	boolean	

BOTRESPONSE

Name	Type	Description
battery_level	integer	
id	integer	
is_charging	boolean	
level_id	integer	
lifted_carrier	integer	

BOXCREATE

Name	Type	Description
section_code		Legacy section code stored as integer with binary digits (8-bit semantics). String inputs are accepted for compatibility: <=8 chars (legacy), exactly 16 chars, or exactly 32 chars.

BOXPATCH

Name	Type	Description
section_code		Legacy section code stored as integer with binary digits (8-bit semantics). String inputs are accepted for compatibility: <=8 chars (legacy), exactly 16 chars, or exactly 32 chars.

BOXREGIONSWAPREQUEST

Name	Type	Description
updates	Array<BoxSectionCodeUpdate>	

BOXREGIONSWAPRESPONSE

Name	Type	Description
boxes	Array<BoxResponse>	

BOXRESPONSE

Name	Type	Description
id	integer	
section_code	integer	Legacy section code stored as integer with binary digits (8-bit semantics). String inputs are accepted for compatibility: <=8 chars (legacy), exactly 16 chars, or exactly 32 chars.
section_code_32	integer	Canonical 32-bit section code stored as integer with binary digits. This field is the source of truth for overlap/placement logic.

BOXSECTIONCODEUPDATE

Name	Type	Description
box_id	integer	
section_code		

BOXSWAPRESPONSE

Name	Type	Description
source_box	BoxResponse	
target_box	BoxResponse	

CARRIERENTITYINPUT

Name	Type	Description
carrier_id	string	

CARRIERONBOARDINGREQUESTCREATEINPUT

Name	Type	Description
balcony_id	integer	
id	string	
type	string	

CHARGEALLBOTSREQUEST

Name	Type	Description
force	boolean	Force charge all bots.

CHARGINGREQUESTCREATEINPUT

Name	Type	Description
bot_id	integer	
force	boolean	
id	string	
level_id	integer	
type	string	

CREATEISSUERESPONSE

Name	Type	Description
key	string	
url	string	

ENTITYRESPONSE

Name	Type	Description
cancelled_quantity	integer	
completed_quantity	integer	
entity_type	EntityType	
id	string	
job_type	string	
succeeded_quantity	integer	
total_quantity	integer	

ENTITYTYPE

Type: string

EVENTSRESPONSE

Name	Type	Description
bot_id	string	
component	string	
created_at	string(datetime)	
created_at_utc	string(datetime)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
description	string	
hex_value	string	
level_id	integer	
request_id	string	
severity	integer	

FETCHREQUESTCREATEINPUT

Name	Type	Description
entities	Array<CarrierEntityInput>	
id	string	
priority	integer	
type	string	

FORGOTPASSWORDREQUEST

Name	Type	Description
email	string(email)	
locale		

FULLFILLMENTREQUESTCREATEINPUT

Name	Type	Description
entities	Array<SkuEntityInput>	
id	string	
priority	integer	
type	string	

HTTPVALIDATIONERROR

Name	Type	Description
detail	Array<ValidationError>	

INVENTORYVIEWRESPONSE

Name	Type	Description
box_height_string_max	integer	
box_id	integer	
box_size	integer	
box_size_string	string	Size label for this row (for empty-space rows this is derived from free-bit thresholds).
carrier_id	string	
content_codes	string	
depth	integer	
eta	number	
eta_int	integer	
id	string	
item_count	integer	
item_count_max	integer	
level_id	integer	
measurement_unit	string	
on_empty_sku_action	string	
position_x	integer	
position_y	integer	
section_code	string	Legacy 8-bit projection derived from canonical 32-bit, returned as string for backward compatibility.
section_code_32	integer	Canonical 32-bit section code stored as integer with binary digits.
sku_id	string	
sku_name	string	
system_quantity	integer	
updated_at	string(date-time)	
updated_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.

JOBRESPONSE

Name	Type	Description
balcony_id	integer	
bot_id	integer	
box_id	integer	
carrier_id	string	
created_at	string(date-time)	
created_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
id	string	
level_id	integer	
measurement_unit	string	
on_empty_sku_action	string	
quantity	integer	
request_id	string	
sku_id	string	
sku_name	string	
status	JobStatus	
type	JobType	
updated_at	string(date-time)	
updated_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.

JOBSTATUS

Type: string

JOBTYPE

Type: string

LEVELRESPONSE

Name	Type	Description
height	integer	
id	integer	

LOADCREATE

Name	Type	Description
item_count	integer	
item_count_max	integer	
sku_id	string	

LOADPATCH

Name	Type	Description
item_count	integer	
item_count_max	integer	
sku_id	string	

LOADRESPONSE

Name	Type	Description
item_count	integer	
item_count_max	integer	
name	string	
sku_id	string	

LOCALEENUM

Type: string

LOGINREQUEST

Name	Type	Description
email	string(email)	
locale		
password	string	

LOGINRESPONSE

Name	Type	Description
access_token	string	
refresh_token	string	
token_type	string	

MEASUREMENTUNIT

Type: string

OFFBOARDINGREQUESTCREATEINPUT

Name	Type	Description
bot_id	integer	
id	string	
type	string	

ONBOARDINGREQUESTCREATEINPUT

Name	Type	Description
id	string	
type	string	

ONEMPTYSKUACTION

Type: string

PAGE_APIKEYINFO_

Name	Type	Description
items	Array<ApiKeyInfo>	
page	integer	
pages	integer	
size	integer	
total	integer	

PAGE_EVENTSRESPONSE_

Name	Type	Description
items	Array<EventsResponse>	
page	integer	
pages	integer	
size	integer	
total	integer	

PAGE_INVENTORYVIEWRESPONSE_

Name	Type	Description
items	Array<InventoryViewResponse>	
page	integer	
pages	integer	
size	integer	
total	integer	

PAGE_JOBRESPONSE_

Name	Type	Description
items	Array<JobResponse>	
page	integer	
pages	integer	
size	integer	
total	integer	

PAGE_REQUESTRESPONSECOMPOSITE_

Name	Type	Description
items	Array<RequestResponseComposite>	
page	integer	
pages	integer	
size	integer	
total	integer	

PAGE_SKURESPONSE_

Name	Type	Description
items	Array<SkuResponse>	
page	integer	
pages	integer	
size	integer	
total	integer	

PAGE_USERSHEMA_

Name	Type	Description
items	Array<UserSchema>	
page	integer	
pages	integer	
size	integer	
total	integer	

PASSWORDPOLICYRESPONSE

Name	Type	Description
min_length	integer	
require_digit	boolean	
require_lowercase	boolean	
require_special	boolean	
require_uppercase	boolean	
special_characters	string	
special_characters_pattern	string	

PAUSEREQUESTCREATEINPUT

Name	Type	Description
id	string	
type	string	

REFRESHTOKENREQUEST

Name	Type	Description
refresh_token	string	

REPLENISHMENTREQUESTCREATEINPUT

Name	Type	Description
entities	Array<SkuEntityInput>	
id	string	
priority	integer	
type	string	

REPORTRESPONSE

Name	Type	Description
report	string	Deprecated: Use status instead
report_path	string	Path where the report is saved
status	string	Status of report generation
tar_filename	string	Name of the generated tar file

REQUESTRESPONSECOMPOSITE

Name	Type	Description
bot_id	integer	
counter	integer	
created_at	string(date-time)	
created_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
entities	Array<EntityResponse>	
id	string	
level_id	integer	
placing_pos	integer	
priority	integer	
status	RequestStatus	
type	RequestType	
updated_at	string(date-time)	
updated_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
warnings	Array<>	

REQUESTSTATUS

Type: string

REQUESTTYPE

Type: string

RESETPASSWORDREQUEST

Name	Type	Description
email	string(email)	
locale	string	
new_password	string	
old_password	string	
otp	string	

RESOLVEDSETTINGSRESPONSE

Name	Type	Description
settings		

RESUMEREQUESTCREATEINPUT

Name	Type	Description
id	string	
type	string	

RFIDWRITEREQUESTCREATEINPUT

Name	Type	Description
action_type	string	
bot_id	integer	
id	string	
level_id	integer	
start_from_scratch	boolean	
type	string	

ROLEENUM

Type: string

SETTINGCREATE

Name	Type	Description
default_value_json		
key	string	
required_role	Array<string>	
restart_required	boolean	
value_json		

SETTINGLISTITEMRESPONSE

Name	Type	Description
default_value_json		
enum_values	Array<string>	
key	string	
required_role	Array<string>	
restart_required	boolean	
scope	string	
user_idx	integer	
value_json		
value_type	string	

SETTINGRESPONSE

Name	Type	Description
created_at	string(date-time)	
created_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
default_value_json		
id	string(uuid)	
key	string	
required_role	Array<string>	
restart_required	boolean	
updated_at	string(date-time)	
updated_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
user_idx	integer	
value_json		

SETTINGSLISTRESPONSE

Name	Type	Description
items	Array<SettingListItemResponse>	
total	integer	

SETTINGSPATCHREQUEST

Name	Type	Description
resets	Array<string>	
updates		

SETTINGUPDATE

Name	Type	Description
value_json		

SIGNUPREQUEST

Name	Type	Description
email	string(email)	
locale		
role		
username	string	

SIGNUPRESPONSE

Name	Type	Description
email	string(email)	
id	integer	
username	string	

SKU

Name	Type	Description
id	string	
measurement_unit		
name	string	
on_empty_sku_action		

SKUENTITYINPUT

Name	Type	Description
quantity	integer	
sku_id	string	

SKUPATCH

Name	Type	Description
measurement_unit	MeasurementUnit	
name	string	
on_empty_sku_action	OnEmptySKUAction	

SKURESPONSE

Name	Type	Description
carrier_utilization	number	
created_at	string(date-time)	
created_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
deleted_at	string(date-time)	
deleted_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
fulfillable	integer	
id	string	
image_url	string	
load_count	integer	
max_quantity	integer	
measurement_unit		
name	string	
on_empty_sku_action		
replenishable	integer	
total_negative_inventory_bookings	integer	
total_positive_inventory_bookings	integer	
total_quantity	integer	
updated_at	string(date-time)	
updated_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.

STORAGERESPONSE

Name	Type	Description
brain_status	string	Deprecated: Use status instead
details	string	
status	string	
testing_script	string	Deprecated: Use status instead

STORAGESTATUS

Type: string

STORAGESTATUSRESPONSE

Name	Type	Description
storage_status	StorageStatus	

TOKENRESPONSE

Name	Type	Description
access_token	string	
refresh_token	string	
token_type	string	

TRIGGERREQUEST

Name	Type	Description
TriggerStatus		

TRIGGERRESPONSE

TRIGGERSTATUS

Type: string

UNCHARGINGREQUESTCREATEINPUT

Name	Type	Description
bot_id	integer	
force	boolean	
id	string	
level_id	integer	
type	string	

USERSCHEMA

Name	Type	Description
email	string(email)	
id	integer	
otp	string	
otp_expires_at	string(date-time)	
otp_expires_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
password_expires_at	string(date-time)	
password_expires_at_utc	string(date-time)	Recommended timestamp field for new integrations. Uses ISO 8601 with an explicit UTC offset. The matching field without the `_utc` suffix is kept for backward compatibility and will be deprecated in a future release.
role	RoleEnum	
username	string	

USERUPDATEREQUEST

Name	Type	Description
email	string(email)	
role	RoleEnum	
user_id	integer	
username	string	

USERUPDATERESPONSE

Name	Type	Description
email	string(email)	
id	integer	
role	RoleEnum	
username	string	

VALIDATIONERROR

Name	Type	Description
loc	Array<>	
msg	string	
type	string	

Security schemes

Name	Type	Scheme	Description
BearerAuth	http	bearer	Bearer token authentication. Use the API key or token issued for your integration.





5. Support & Service

5.1 Fehlerbehebung

Gelegentlich können Probleme mit NoyesStorage auftreten. Um ein Problem effektiv zu lösen, ist es wichtig, zunächst dessen Ursache zu verstehen. Dies kann über die NoyesStorage Status- und Fehlerbehebungsseite oder durch Überprüfung des NoyesBot-Status erfolgen. Diese Ressourcen unterstützen bei der Fehlerdiagnose und ermöglichen eine schnelle Lösung — entweder eigenständig oder mit Unterstützung unseres [Support-Teams](#).

5.1.1 NoyesStorage Status

Der NoyesStorage-Status wird in der Kopfzeile der App angezeigt und ist somit stets sichtbar. Die folgenden Farben kennzeichnen den Status:

Farbe	Bedeutung	Symbol
Grün	NoyesStorage ist betriebsbereit.	
Rot	NoyesStorage hat einen Fehler.	
Gelb	NoyesStorage ist pausiert.	
Grau	NoyesStorage ist ausgeschaltet.	

Bei Auftreten eines Problems leuchtet die Statusanzeige dauerhaft rot, und der Benutzer wird auf die Troubleshooting-Seite weitergeleitet. Diese Seite bietet detaillierte Informationen zum Problem und unterstützt bei der Diagnose und Lösung.

5.1.2 Support Center nutzen

Das integrierte Support Center bündelt alle Kontaktwege und Selbsthilfe-Inhalte an einem Ort:

- **E-Mail Support:** Öffnet ein vorausgefülltes Mailfenster an support@noyes-tech.com mit typischer Antwortzeit (laut Hinweis im UI) innerhalb von 12 Stunden.
- **Telefon Support:** Zeigt die direkte Support-Hotline inkl. Erreichbarkeitszeitfenster.
- **Problem melden:** Erstellt ein Support-Ticket direkt aus der App, damit das Team Ihr Anliegen nachverfolgen kann.
- **FAQs:** Kurze Antworten zu häufigen Fragen, z. B. Vorgehen bei Systemstatus „Fehler“ (mit/ohne Roboterproblem) und Erkennen von Bot-Problemen.

Support Center
Holen Sie sich Hilfe von unserem Support-Team. Wir sind für Sie da.

E-Mail Support
Schreiben Sie uns eine E-Mail

support@noyes-tech.com
Wir antworten in der Regel innerhalb von 12 Stunden

Telefon Support
Rufen Sie uns direkt an

+49 158 88 64 77 23
Montag - Freitag, 08:00-16:00 MEZ (ausgenommen gesetzliche Feiertage)

Problem melden
Erstellen Sie ein Ticket. [Ticket erstellen](#)

Senden Sie Fehlerberichte, Funktionswünsche oder technische Probleme. Unser Team verfolgt und löst Ihr Ticket zeitnah.

Frequently Asked Questions
Find quick answers to common questions

- [Systemstatus „Fehler“ \(kein Roboterproblem\) – wie weiter kommissionieren?](#) ▾
- [Systemstatus „Fehler“ \(mit Roboterproblem\) – wie weiter kommissionieren?](#) ▾
- [Woran erkenne ich, ob ein NoyesBot ein Problem hat?](#) ▾

So gelangen Sie dorthin:

Hinweis

Benutzeranmeldung vorausgesetzt.




1. Öffnen Sie in der StorageUI das **Support Center** (Support-Icon oder Menüeintrag).
2. Wählen Sie den gewünschten Kanal:
3. Ticket oder E-Mail für strukturierte Fehlermeldungen.
4. Hotline bei dringenden Fällen innerhalb der Servicezeiten.
5. FAQ, wenn Sie eine schnelle Schritt-für-Schritt-Anleitung benötigen.
6. Nutzen Sie die angegebenen Kontaktinformationen bzw. klappen Sie die FAQ-Einträge auf, um die relevanten Schritte direkt zu sehen.

5.1.3 NoyesBot Status

Der NoyesBot ist mit einem Satz LEDs ausgestattet, die seinen Status visuell anzeigen und so die Fehlerbehebung erleichtern. Die Farben und Muster, die auf dem NoyesBot angezeigt werden, kommunizieren Status- und Fehlersignale, wobei jedes LED-Muster über drei Zeitintervalle wechselt, um den aktuellen Zustand des

Roboters schnell zu erkennen oder Probleme zu diagnostizieren. Die folgenden LED-Farben und -Muster werden verwendet:

LED-Farbe & Muster	Bedeutung	Zeit 1	Zeit 2	Zeit 3
Dauerhaft Grün	NoyesBot ist bereit; keine Aktion in Arbeit.		-	-
Blinkendes Grün	NoyesBot führt einen Befehl aus.			-
Dauerhaft Violett	RFID-Schreibmodus aktiviert.		-	-
Pulsierendes Orange	Batterie wird gerade geladen.			
Dauerhaft Gelb	<i>Schritt 1: Verbindung mit Wi-Fi herstellen.</i>		-	-
Dauerhaft Hellblau	<i>Schritt 2: Verbindung zum MQTT-Broker herstellen.</i>		-	-
Blinkendes Gelb	<i>Schritt 3: Warte auf Bestätigung von NoyesStorage.</i>			-
Abwechselnd blinkend Rot und Blau	Ungültige Startposition; RFID-Tag konnte nicht gelesen werden.			-
Blinkendes Rot	Kritisches Problem; Bewegung verhindert.			-
Abwechselnd blinkend Grün und Gelb	Problem mit dem Hubmechanismus erkannt.			-
Blau, Rot, Grün	Problem mit dem IMU-System erkannt.			
Rot, Blau, Rot	Keine Batterien im CAN-Bus erkannt.			
Rot, Rot, Blau	Ungültige Encoder-Kalibrierung.			

LED-Farbe & Muster	Bedeutung	Zeit 1	Zeit 2	Zeit 3
Rot, Blau, Blau	Ungültige Ladekalibrierung.			

5.1.4 Problembehebung

Probleme mit dem NoyesStorage

1. Schalten Sie den NoyesStorage aus.
2. Schalten Sie den NoyesStorage wieder ein.
3. Fahren Sie mit den Anfragen fort.
4. Das System sollte nun wieder funktionsfähig sein. Sollte das Problem weiterhin bestehen, kontaktieren Sie bitte den Support unter support@noyes-tech.com.

Probleme mit einem NoyesBot

HINWEIS

Lediglich der Maintainer darf nach erfolgreich absolviertem Training zur Fehlerbehebung – nach entsprechenden sicherheitsgerichteten Vorkehrungen – in den NoyesStorage klettern.

1. Schalten Sie den NoyesBot aus.
2. Schalten Sie den NoyesStorage aus.
3. Schalten Sie den NoyesStorage wieder ein.
4. Fahren Sie mit den Anfragen fort.
5. Schalten Sie den NoyesBot wieder ein.
6. Das System sollte nun funktionsfähig sein. Sollte das Problem weiterhin bestehen, kontaktieren Sie bitte den Support unter support@noyes-tech.com.

Bemerkung: Sollte es dabei zu falsch platzierten Carriern kommen, müssen diese auch aus dem System entfernt und später wieder manuell eingelagert werden.

Probleme mit Tablet und App

1. **Schließen Sie die NoyesStorage-App vollständig** (Wichtig: nicht nur die App verlassen!).
2. Starten Sie die NoyesStorage-App neu.
3. Löschen Sie den App-Cache.
4. Melden Sie sich erneut an.
5. Das System sollte nun funktionsfähig sein. Sollte das Problem weiterhin bestehen, kontaktieren Sie bitte den Support unter support@noyes-tech.com.

5.2 Reinigung, Inspektion, Wartung und Reparatur

5.2.1 Reinigung

Die erforderlichen Reinigungsaktivitäten und die entsprechenden Zeitintervalle sind von den Umgebungsbedingungen abhängig.

Folgende Aktivitäten und Intervalle sind empfohlen:

Intervall	Reinigungsarbeit
Monatlich	Teilreinigung: Reinigen Sie alle Balkone und Autobahnen gründlich mit einem sauberen, weichen, faserfreien Tuch, um das Eindringen von Schmutz in das System zu minimieren.
Vierteljährlich	Systemreinigung: Schalten Sie den NoyesStorage aus und nehmen Sie alle NoyesBots außer Betrieb (Offboarden). Reinigen Sie anschließend alle Moduloberflächen im ausgeschalteten NoyesStorage. Unsere Kunden empfehlen die automatisierte Reinigung mittels eines Staubsaugerroboters (z. B. SwitchBot K10+). Danach reinigen Sie die Räder und Sensoren der NoyesBots mit einem sauberen, weichen, faserfreien Tuch. Starten Sie abschließend das System neu und nehmen Sie die Roboter wieder in Betrieb (Onboarden) – das System ist dann wieder voll betriebsbereit.

5.2.2 Inspektion

Um eine sichere und ausfallfreie Nutzung sicherzustellen, sind folgende Inspektionsarbeiten vorgeschrieben:

Intervall	Inspektionsarbeit
Vor jedem Gebrauch	Stellen Sie sicher, dass alle Abdeckungen des NoyesStorage montiert sind. Falls Abdeckungen fehlen, informieren Sie Ihren Store Manager.
Vor jedem Gebrauch	Stellen Sie sicher, dass alle Kantenschutzteile (schwarzes Gummi) der Übergabestationen montiert sind. Falls Teile fehlen oder lose sind, informieren Sie Ihren Store Manager.
Vor jedem Gebrauch	Stellen Sie sicher, dass sich keine Fremdkörper in oder auf dem NoyesStorage befinden.
Vor jedem Gebrauch	Stellen Sie sicher, dass alle Laufwege frei von Stolperquellen sind und eine eventuell vorhandene Treppe sicher positioniert ist.

5.2.3 Wartung

Der NoyesStorage selbst bedarf kaum Wartung. Folgende Arbeiten sind jedoch empfohlen, um die Lebensdauer zu erhöhen und die Ausfallraten gering zu halten:

Intervall	Wartungsarbeit
24 Monate	Antriebseinheiten der NoyesBots warten. <i>(Bei größeren Verunreinigungen früher.)</i>
24 Monate	DGUV V3 Prüffristen einhalten
36 Monate	Hubsysteme der NoyesBots warten. <i>(Bei größeren Verunreinigungen früher.)</i>
48 Monate	Batterien der NoyesBots warten. <i>(Bei intensiver Nutzung und Mehrschichtbetrieb früher.)</i>
48 Monate	Ladestationen des NoyesStorage warten.

Wartungsarbeiten am NoyesBot müssen nicht vor Ort stattfinden. Die Roboter können zur Wartung einfach eingeschickt werden und kommen innerhalb weniger Tage gewartet zurück.

Wartungsarbeiten am NoyesStorage dürfen nur von qualifiziertem, trainiertem Personal durchgeführt werden.

Bei allen Wartungsarbeiten sind die zusätzlichen Sicherheitsanforderungen zu berücksichtigen.

5.2.4 Reparatur

Sollten trotz korrekter und pflichtbewusster Reinigung, Inspektion und Wartung Fehler auftreten, die nicht mithilfe eines Software-Updates behoben werden können, muss das System für die entsprechende Reparatur heruntergefahren werden.

GEFAHR

Verletzungsgefahr durch sich bewegende Teile!

Beim Hineinklettern in den NoyesStorage oder der Demontage von Abdeckungen besteht Verletzungsgefahr durch sich bewegende Teile.

- Unbefugte dürfen den NoyesStorage zu keiner Zeit betreten.
- Abdeckungen an den Seiten und der Decke des Systems dürfen von Unbefugten nicht demontiert werden.
- Der NoyesStorage ist in einen Ruhezustand zu bringen, bevor er von geschultem Personal betreten wird.

Reparaturarbeiten am NoyesBot müssen nicht vor Ort stattfinden. Die Roboter können zur Reparatur einfach eingeschickt werden und kommen innerhalb weniger Tage funktionstüchtig zurück.

Reparaturarbeiten am NoyesStorage dürfen nur von qualifiziertem, trainiertem Personal durchgeführt werden.

Bei allen Reparaturarbeiten sind die zusätzlichen Sicherheitsanforderungen zu berücksichtigen.

5.3 Transport und Lagerung

Informationen zur Handhabung

HINWEIS

Die Installation und Erstinbetriebnahme wird nur von Mitarbeitern der Noyes Robotics GmbH oder autorisiertem Personal durchgeführt. Während der Installationsarbeiten und der weiteren Nutzung kann es jedoch erforderlich sein, dass das Personal des Betreibers bei der Handhabung der Lieferung behilflich ist. Beachten Sie dabei unbedingt alle Hinweise.

Der NoyesStorage besteht aus mehreren Modulen mit den folgenden Abmessungen:

Breite	698 mm
Höhe	367 mm/509 mm + Füße (50 mm)
Tiefe	512 mm

5.3.1 Transport

Alle Teile des NoyesStorage passen auf EURO-Paletten. Die Teile werden unter Umständen auf mehrere Lieferungen aufgeteilt. Für den Versand auf dem Seeweg werden Seecontainer verwendet.

5.3.2 Verpackung

Die einzelnen Pakete sind entsprechend den zu erwartenden Transportbedingungen verpackt.

Die Verpackung soll die einzelnen Komponenten bis zum Einbau vor Transportschäden, Korrosion und anderen Beschädigungen schützen. Daher darf die Verpackung nicht zerstört und erst kurz vor dem Aufbau entfernt werden.

5.4 Demontage und Entsorgung

5.4.1 Nutzungsende

Sobald der NoyesStorage das Ende seiner Nutzung erreicht hat, muss er demontiert und umweltgerecht entsorgt werden.

5.4.2 Fehlerhafte Demontage

WARNUNG

Verletzungsgefahr durch unsachgemäße Demontage!

Fehler bei der Durchführung von Demontearbeiten können zu lebensgefährlichen Situationen und erheblichen Sachschäden führen.

- Die Demontage des Storage darf nur von Mitarbeitern der Noyes Robotics GmbH oder von ihr autorisiertem Personal durchgeführt werden.

6. Changelog

Hier finden Sie Information über die neuesten Updates und Verbesserungen des NoyesStorage

6.1 v4.38 (783fdef39) ab 10.03.2026

Anpassbare Startseite und flexible Shortcuts

Die Startseite der NoyesStorage App lässt sich jetzt deutlich stärker an die eigenen Arbeitsabläufe anpassen. Wichtige Karten und Shortcuts können direkt auf der Home-Seite platziert werden, sodass zentrale Prozesse und Informationen ohne zusätzliche Navigation verfügbar sind.

Zusätzlich stehen mehr interne Shortcuts auf vorhandene Seiten zur Verfügung. Shortcuts können auch mit Filtern verwendet werden, zum Beispiel um das Inventory direkt mit einer bestimmten Boxgröße oder einem bestimmten SKU-Typ zu öffnen. Das macht die Oberfläche flexibler und spart Zeit in wiederkehrenden Abläufen.

Neu ist außerdem, dass Shortcuts auch auf externe Seiten verweisen können. Dadurch lässt sich die Startseite besser an standortspezifische Prozesse und ergänzende Tools anpassen.

Mehr Struktur in Analytics

Die Analytics-Seite wurde klarer gruppiert. Relevante Kennzahlen und Widgets lassen sich damit schneller finden und gezielter nutzen.

Auch auf der Startseite profitieren Sie davon: Analytics-Karten können dort gezielt ergänzt werden, damit wichtige Informationen direkt beim Einstieg sichtbar sind.

Mehr Überblick bei Carrier und Packing

Die Carrier-Ansicht in den Job-Details zeigt jetzt mehr Ladungsinformationen direkt in der Übersicht. Viele Details sind dadurch ohne weiteres Tippen oder Öffnen zusätzlicher Ansichten ablesbar.

Zusätzlich wurde die Packing-Übersicht präzisiert: Angeforderte, gepickte und stormierte Mengen werden jetzt getrennt dargestellt. Dadurch ist schneller erkennbar, was tatsächlich bearbeitet wurde und welche Positionen abgebrochen wurden.

Flexiblere Carrier mit 16 Boxen

Carrier können jetzt feiner aufgeteilt werden. Durch die neue 16er-Struktur lassen sich Inhalte differenzierter darstellen und kleinere Einheiten besser abbilden.

Uhrzeit direkt in der oberen Leiste

In der oberen Leiste steht jetzt optional eine Uhr zur Verfügung. Sie kann je nach Bedarf ein- oder ausgeblendet werden und unterstützt sowohl das 12-Stunden- als auch das 24-Stunden-Format.

6.2 v4.37 (d2edd42e7) ab 24.02.2026

Kleine Verbesserungen im Request Dashboard

Im Request Dashboard der NoyesStorage App haben wir die technische Basis nachgebessert. Die Liste reagiert damit in manchen Situationen etwas schneller und aktualisiert sich verlässlicher.

Außerdem werden Requests jetzt klar als eigene Zeilen dargestellt. Das macht die Ansicht etwas übersichtlicher, gerade wenn mehrere Aufträge parallel laufen.

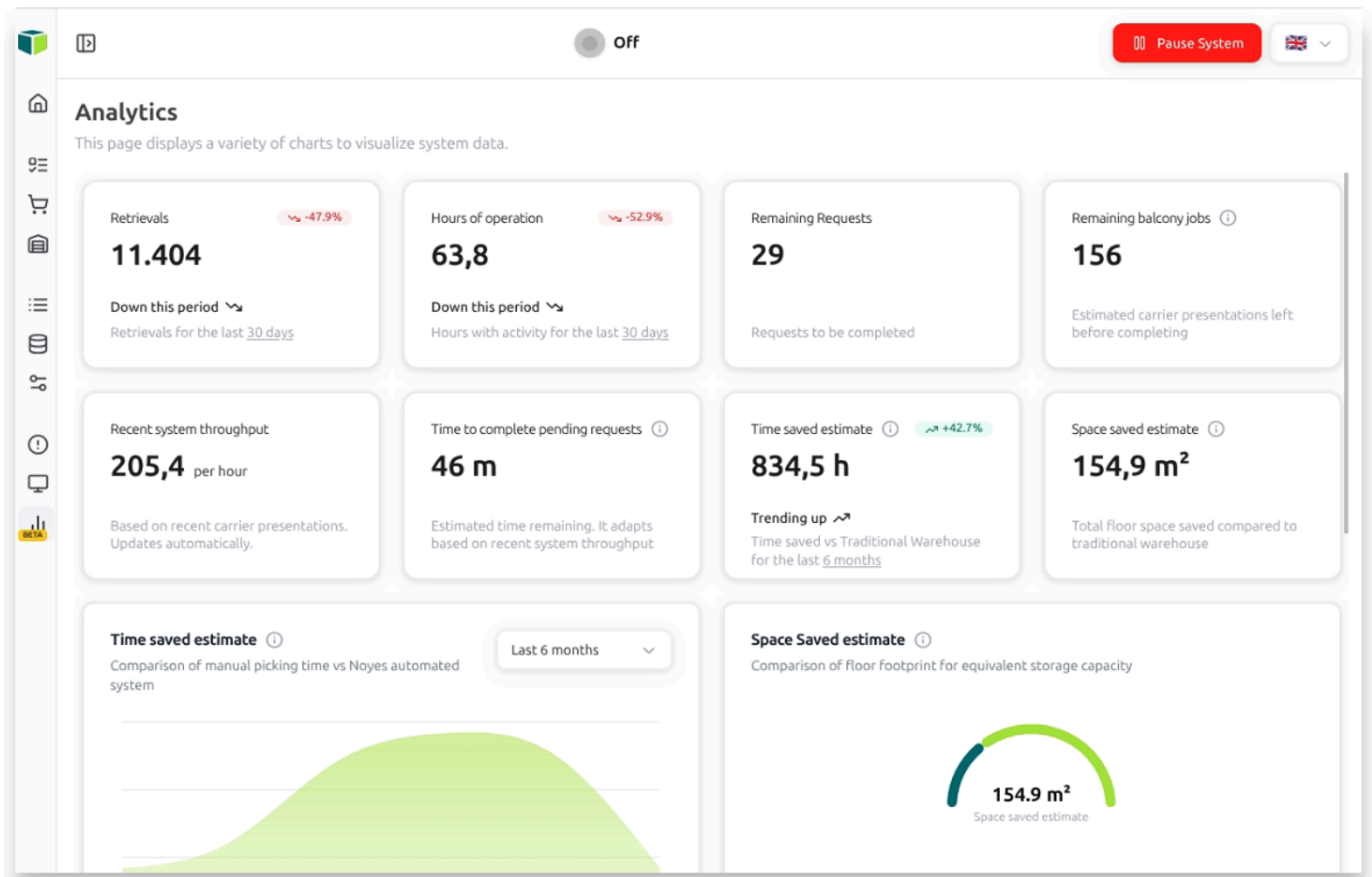
Kleines Stabilitätsupdate bei Bot-Störungen

Wenn ein NoyesBot auf einer Ebene als nicht fahrbereit erkannt wird, sendet das System dort keine weiteren Fahrbefehle. So lassen sich mögliche Folgefehler besser begrenzen.

6.3 v4.36 (8c7bdc6f7) ab 11.02.2026

Genauere Durchsatzstatistiken

Auf dem Analytics Dashboard steht ein neues Widget für den aktuellen Systemdurchsatz zur Verfügung. Die Schätzungen für die verbleibende Bearbeitungszeit und die Anzahl offener Requests basieren jetzt auf dem tatsächlichen Durchsatz der letzten Betriebsstunden. Damit erhalten Sie genauere Prognosen für die Planung Ihrer Arbeitsabläufe.



Fehlerbehebungen

Die Farbzunordnung im Radialdiagramm der Analytics-Seite wurde korrigiert: Die Tooltip-Farben stimmen jetzt mit den jeweiligen Diagrammsegmenten überein.

6.4 v4.35 (4a61c0c9e) ab 27.01.2026

Vorwärts-Laden und erweiterte Analytics-Metriken

Neuere Bot-Modelle unterstützen jetzt das Laden in Vorwärtsrichtung (Forward Charging). Da die rückwärtigen Ladekontakte bei neueren Generationen entfallen sind, stellt diese Funktion sicher, dass alle Kundenstandorte sowohl neue als auch ältere Robotergenerationen betreiben können. Die Aktivierung erfolgt schrittweise für alle Standorte.

Das Analytics Dashboard zeigt jetzt zusätzlich die eingesparte Zeit und den eingesparten Platz durch Ihren NoyesStorage an. Diese Metriken helfen Ihnen, den Mehrwert des Systems zu quantifizieren.

Fehlerbehebungen

Ein Fehler im Exportdialog wurde behoben, der die Auswahl eines anderen Empfängers verhinderte. Zusätzlich wurde das Layout der Support Center Seite für eine bessere Übersichtlichkeit angepasst.

6.5 v4.34 (8bfbf673c) ab 12.01.2026

Auftragsimport per Einzelscan und Status-LED

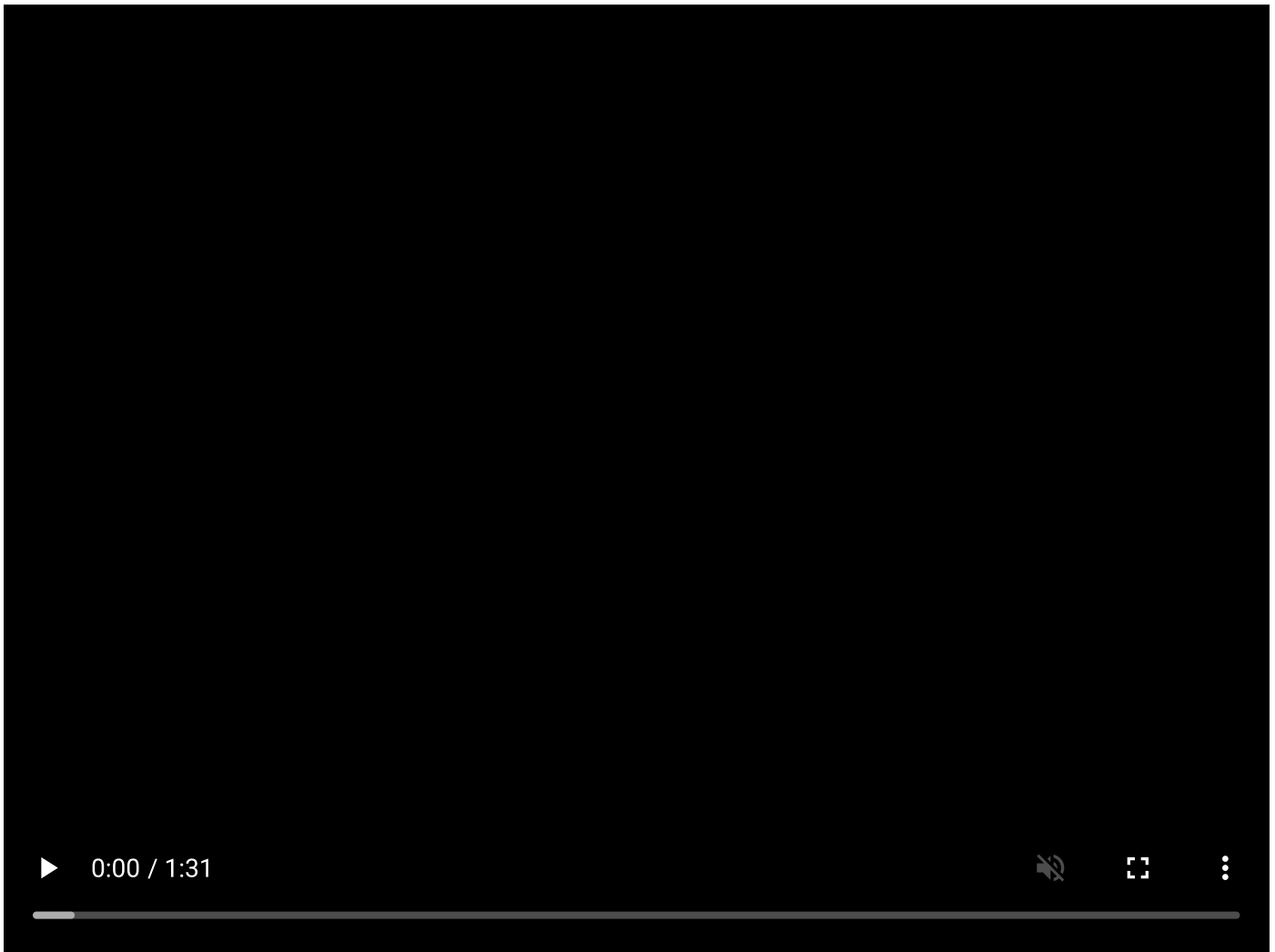
Ab sofort können komplette Aufträge durch das Scannen eines einzigen QR-Codes oder Barcodes übermittelt werden. Statt jeden Artikel einzeln zu erfassen, genügt ein Scan für den gesamten Auftrag. Dies vereinfacht die Auftragsübergabe und reduziert den manuellen Aufwand am Arbeitsplatz.

Neu ist eine physische Status-LED am System. Diese zeigt den aktuellen Systemstatus direkt am Gerät an, ohne dass die Benutzeroberfläche geöffnet werden muss. Bediener erkennen so den Betriebszustand auf einen Blick, auch wenn sie nicht direkt vor dem Bildschirm stehen.

6.6 v4.33 (44da13c9d) ab 19.12.2025

Neues Analytics Dashboard für operative Einblicke

Mit diesem Release führen wir ein neues Analytics Dashboard ein, das Ihnen detaillierte Einblicke in den Betrieb Ihres NoyesStorage bietet. Das Dashboard ist direkt in der Benutzeroberfläche verfügbar und zeigt wichtige Kennzahlen in übersichtlichen Widgets.



Ein zentrales Widget zeigt die geschätzte Bearbeitungszeit für alle offenen Requests. Diese Information hilft Bedienern zu verstehen, wie lange sie noch mit den verbleibenden Aufträgen beschäftigt sein werden, und ermöglicht eine bessere Planung der Arbeitsabläufe.

Ein weiteres Widget identifiziert "Top Runner" und "Low Runner" – also Produkte, die häufig oder selten bewegt werden. Diese Übersicht unterstützt Sie bei Bestandsentscheidungen: Häufig genutzte Artikel können strategisch platziert werden, während selten genutzte Artikel möglicherweise ausgelagert oder konsolidiert werden sollten. So können Sie Ihre Lageroptimierung datenbasiert durchführen.

Das Analytics Dashboard wird kontinuierlich erweitert. Basierend auf Ihrem Feedback entwickeln wir weitere Funktionen, die Ihnen helfen, den Betrieb Ihres NoyesStorage noch effizienter zu gestalten.

Schnellerer Systemneustart über die Benutzeroberfläche

Ein neuer Neustart-Button in der Systemsteuerung ermöglicht einen deutlich schnelleren Systemneustart. Im Gegensatz zum manuellen Neustart werden die Roboter nicht mehr zur Ladestation geschickt, bevor das System herunterfährt. Dies spart wertvolle Minuten im Betriebsablauf, besonders wenn ein schneller Neustart nach einem Fehler erforderlich ist.

Der Neustart erfolgt weiterhin sicher und vollständig, nur ohne die zusätzliche Wartezeit bis die Roboter zum Laden fahren. Dies ist besonders vorteilhaft, wenn das System nach einem Fehler schnell wieder in Betrieb genommen werden muss.

Verbesserte Systemüberwachung und Fehlererkennung

Das System erkennt jetzt automatisch, wenn die Firmware-Version eines Bots nicht mit der erwarteten Version übereinstimmt. In diesem Fall wird ein Warn-Event generiert, das Sie frühzeitig auf mögliche Probleme hinweist. So können Bot-Firmware-Probleme erkannt werden, bevor sie zu Betriebsstörungen führen.

Zusätzlich werden neue Events für fehlerhafte Bot-Bewegungen und falsche Bot-Feedback-Meldungen protokolliert. Diese Events enthalten präzise Zeitstempel und Ursachenhinweise, was die Fehleranalyse deutlich erleichtert. Bei der Weiterleitung von Problemen an den Support können Sie nun konkrete Informationen zu Zeitpunkt und Art des Fehlers bereitstellen.

Nachvollziehbarkeit von API- und Benutzer-Aktionen

Das System unterscheidet jetzt zwischen Requests und Jobs, die über API-Keys erstellt wurden, und solchen, die von Benutzern über die Benutzeroberfläche ausgelöst wurden. Diese Information ist in der Request- und Job-Übersicht sichtbar und erhöht die Transparenz darüber, welche Aktionen automatisiert über die API und welche manuell durch Bediener durchgeführt wurden.

Performance-Verbesserungen

Die API-Antwortzeiten wurden optimiert, sodass alle API-Aufrufe nun schneller verarbeitet werden. Dies verbessert die Reaktionszeit über die Benutzeroberfläche, bei der Integration mit externen Systemen und macht die Nutzung der API insgesamt flüssiger.

Verbesserte Benutzerfreundlichkeit auf iPad

Bei der Nutzung auf iPad-Geräten scrollen Dialoge und Tabellen jetzt korrekt, wenn die Tastatur geöffnet ist. Bisher konnte es vorkommen, dass durch Scrollen bei geöffneter Tastatur wichtige Bereiche der Benutzeroberfläche verschwunden sind. Dies ist jetzt behoben, sodass alle Funktionen auch bei geöffneter Tastatur vollständig zugänglich sind.

Die Tabellensuche wurde ebenfalls optimiert: Bei schneller Eingabe werden veraltete Suchergebnisse automatisch abgebrochen, sodass immer die aktuellsten Ergebnisse angezeigt werden. Dies verhindert Verwirrung durch überholte Suchergebnisse und macht die Suche zuverlässiger.

Verbesserte Stabilität

Ein seltener Fehler in der Bot-Kommunikation, der zu Abstürzen führen konnte, wurde behoben. Dies erhöht die allgemeine Stabilität des Systems, insbesondere bei längeren Betriebszeiten.

Zusätzlich wurde eine Race Condition beim Abmelden (Offboarding) von Robotern behoben, die zu Abstürzen des Bot-Drivers führen konnte. Dies stellt sicher, dass Roboter zuverlässig aus dem System entfernt und wieder hinzugefügt werden können.



Hinweis

Dieses Release enthält Sicherheitsverbesserungen für CIS-Compliance (ClamAV Anti-Malware-Scanning). Diese Änderungen laufen im Hintergrund und sind für Sie als Nutzer nicht sichtbar, verbessern aber die System-Sicherheit und erfüllen wichtige Compliance-Anforderungen.

6.7 v4.32 (f0f86cc27) ab 01.12.2025

Klare Alarme & detaillierte Ereignisse

Wenn 10 Minuten lang keine Batteriemeldung eingeht, springt der Systemstatus automatisch auf ERROR. Ausfälle werden damit sofort sichtbar, und Bediener können eingreifen, bevor der Betrieb zum Stillstand kommt.

Neue Log-Events für fehlerhafte Moves und Host-Neustarts zeichnen den Verlauf präzise nach. Das erleichtert es, Weiterleitungen an Support/Technik mit klaren Zeitstempeln und Ursachenhinweisen zu versehen.

Fällt ein Bot aus, wird der Problemstatus klar hervorgehoben und in den Events abgebildet. So ist unmittelbar nachvollziehbar, wann und warum ein Bot ausgefallen ist, und Folgeschritte können gezielt eingeleitet werden.

6.8 v4.31 (0d4cd4c3f) ab 17.11.2025

Batterie-Monitoring

Neue Events protokollieren Lade- und Entladevorgänge – von Start bis Abschluss – und erfassen dabei auch ungewöhnliche Zustände. So entsteht eine klare Historie für Betrieb und Support, um Ladeprobleme schneller einzugrenzen.

Formulare markieren Pflichtfelder und Fehler deutlicher (rote Sternchen). Das senkt Fehleingaben und beschleunigt den Ablauf im Tagesgeschäft, weil Rückfragen entfallen.

Optimierte Kommunikationsstruktur zwischen lokalen Noyes Komponenten: Die Änderung setzt jüngste Empfehlungen zur Datensicherheit um. Nach Durchführung der Änderung müssen Sie die URL für ihre StorageUI anpassen. Sehen Sie [hier](#), wie Sie die Änderungen innerhalb weniger Minuten selbst vornehmen können.

Fehlerbehebung in der Deutschen Benutzeroberfläche: Bei Anpassung der Boxgröße kam es teilweise zu fehlerhafter Darstellung eines Buttons.

6.9 v4.30 (04b4ec69b) ab 03.11.2025

Login robuster, Feedback klarer

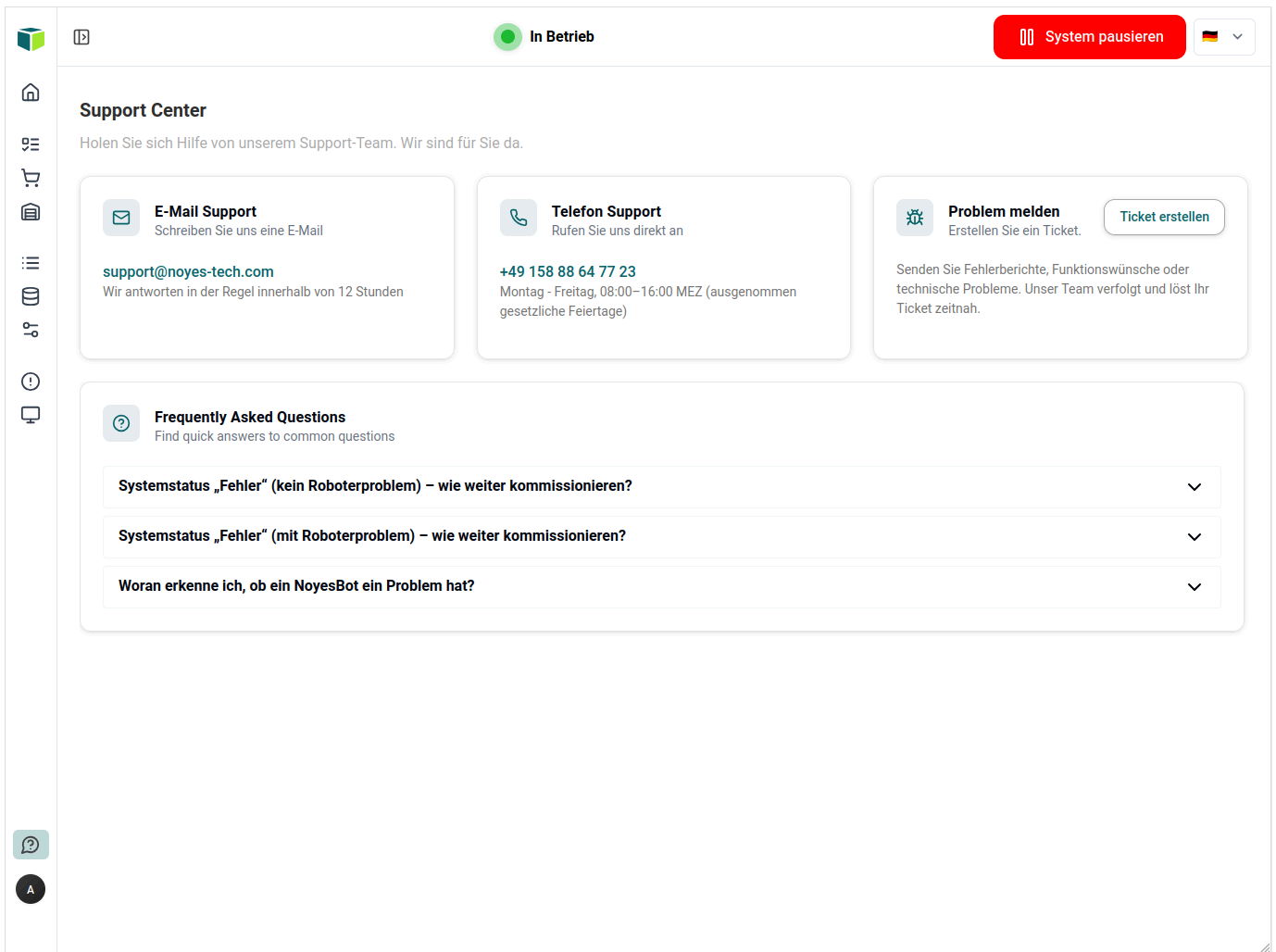
Logins sind nun schreibungsunabhängig, d.h. Groß-/Kleinschreibung spielt keine Rolle mehr. Fehleingaben und Support-Rückfragen rund um Passworteingaben werden reduziert.

Bei doppelten Request-IDs gibt die API eindeutiges Feedback; im Troubleshooting hebt ein Farbcode kritische Severity hervor, damit Prioritäten sofort erkennbar sind.

6.10 v4.29 (5a42740fa) ab 20.10.2025

Support-Seite und zentrales Logging

Einführung des Noyes Support Centers in der UI: Support-Kontakte, FAQs und ein direkter Ticket-Button stehen in einem eigenen Fenster bereit – Hilfe ist ohne Medienbruch erreichbar.



Logdaten werden lokal gespeichert, sodass Support schneller reagieren kann und Diagnosen konsistent vorliegen.

6.11 v4.28 (00a0b2853) ab 06.10.2025

Global App Access & transparenter Fehler-Workflow

Global App Access (Beta) ermöglicht nahtlose Bedienung bereichsübergreifend und – wo freigeschaltet – auch remote über das Internet.

Auf der Fehlerbehebungsseite wird der Status (Start/Pause/Shutdown) protokolliert; ein neues Pop-up weist direkt in der UI auf Fehler hin und verlinkt die Fehlerbehebungsseite, damit Bediener sofort die passende Anleitung sehen.

Verbessertes Verhalten beim Herunterfahren verkürzt in vielen Fällen die Wartezeit.

Troubleshooting- und Trigger-Validierung wurden optisch und fachlich nachgeschärft, damit Fehler schneller gefunden und korrekte Job-Typen genutzt werden.

6.12 v4.27 (da4af2fb5) ab 19.09.2025

Rollen, API-Keys und mehr Transparenz

User-Management mit Rollen User/Admin direkt in der UI: Admins können User anlegen, bearbeiten oder löschen und API-Keys verwalten (für Integrationen). Das reduziert Rückfragen an den Support und ermöglicht Self-Service.



In Betrieb

System pausieren



Benutzerverwaltung

Suchen Sie nach bestehenden Benutzern oder erstellen Sie neue.



Suchen...

Alle Filter löschen



+ Benutzer hinzufügen

Benutzer bearbeiten

<input type="checkbox"/>	ID	Benutzername	E-Mail	Rolle	Passwort läuft ab am	OTP	
<input type="checkbox"/>	2	Noyes	test@gmail.com	Admin	N/A	888279	
<input type="checkbox"/>	3	Noyes Hotmail	test@hotmail.com	Super Admin	19 May 2026 21:09	-	
<input type="checkbox"/>	1	Noyes Support	support@noyes-tech.com	Super Admin	29 Mar 2027 10:43	-	

Angezeigte Elemente: 3 / 3 - Ausgewählt: 0 Zeile(n)




ID
In Betrieb
System pausieren 


API-Key-Verwaltung

Verwalten Sie API-Key für Systemzugriff und Authentifizierung.



Suchen...

Alle Filter löschen

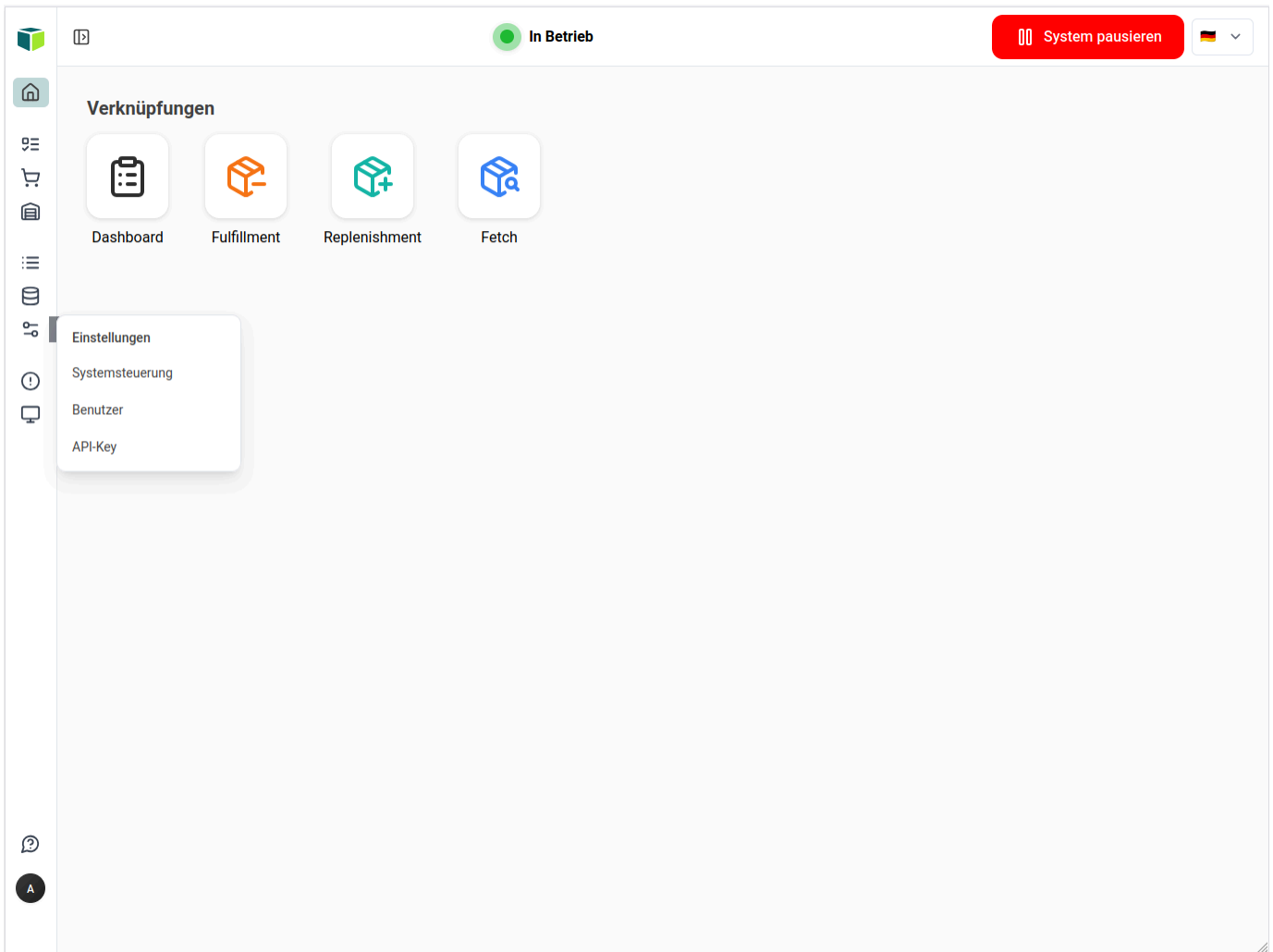
[+ API-Key hinzufügen](#)
[API-Key löschen](#)

<input type="checkbox"/>	ID	Name	API-Key	Läuft ab am	Zuletzt verwendet	
<input type="checkbox"/>	3	ERP Integration	4ca0... 	31 Dec 9999 23:59	N/A	

Angezeigte Elemente: 1 / 1 - Ausgewählt: 0 Zeile(n)

Das überarbeitete Menü führt ohne Umwege zur User- und API-Key-Verwaltung.



Inventarseite mit neuer Spalte „Aktualisiert am“: Sortierbar, um selten genutzte Ladungen schnell zu identifizieren und Lagerplätze gezielt freizumachen.

Requests & Jobs mit Spalten „Erstellt von“ und „Bestätigt von“ erhöhen die Nachvollziehbarkeit. Die Flyout-Sidebar hält die Navigation kompakt sichtbar.

In Betrieb
System pausieren
🇩🇪

Anfragen & Jobs

Diese Tabelle bietet einen Echtzeit-Überblick über alle Requests und Jobs. Sie können den Fortschritt laufender Jobs verfolgen, abgeschlossene Requests überprüfen und nach bestimmten Kriterien filtern, um schnell relevante Informationen zu finden.

Job Typ: REFILLING, PICKING, CHECKING ✕
Tabelle exportieren
Requests abbrechen

created_at__gte: 2025-11-08T14:55:04.544Z ✕
Alle Filter löschen ▾*

<input type="checkbox"/>	Request Typ ▾	Request ID ▾	Job Typ ▾	Job Status ▾	Aktualisiert ▾ ▾	Erstellt von ▾	Bestätigt von ▾	Levi	
<input type="checkbox"/>	Fetch	CO-1208-15:53-50128 fa0-a48c-4342-8549- d6402df2996b	Checking	Cancelled	08 Dec 2025 15:53	admin1	-	3	
<input type="checkbox"/>	Fetch	CO-1208-15:52- f266878e-f964-4cdf- adba-597e4972be8e	Checking	Executing	08 Dec 2025 15:53	admin1	-	2	
<input type="checkbox"/>	Fetch	CO-1208-15:53- e6475a05-21c2-4f1c- bd8e-424f261dd85c	Checking	Accepted	08 Dec 2025 15:53	admin1	-	2	
<input type="checkbox"/>	Fetch	CO-1208-15:52- a1699f15-49e5-455c- b0fd-289059e39304	Checking	Cancelled	08 Dec 2025 15:52	admin1	-	2	
<input type="checkbox"/>	Fulfillment	FO-1205-11:14-81402 d78-8e08-45b7-94f9-5 43f4d4c1b19	Picking	Succeeded	05 Dec 2025 12:09	admin1	admin1	2	
<input type="checkbox"/>	Fetch	CO-1205-11:14-13624 812-5be1-4ed9- a0a1-95c40d00194e	Checking	Succeeded	05 Dec 2025 12:09	admin1	admin1	4	
<input type="checkbox"/>	Fetch	CO-1205-11:14- a785e6ae-	Checking	Succeeded	05 Dec 2025	admin1	admin1	1	

Angezeigte Elemente: 33 / 33 - Ausgewählt: 0 Zeile(n)

6.13 v4.26 (8ed961f77) ab 05.09.2025

Performance-Upgrade im Orchestrator

Der Store Orchestrator verarbeitet viele und große Requests deutlich effizienter und startet nach Fehlern schneller neu – spürbar bei großen Sammelaufträgen.

Bot-Logger-Events sind konsolidiert, Trigger-Verhalten vereinheitlicht und API-Filter (iLike) behandeln leere Werte korrekt. Das reduziert Fehlalarme und verbessert die Analysierbarkeit.

6.14 v4.25 (b0e02f1e7) ab 12.08.2025

Troubleshooting-Events statt Websocket

Troubleshooting-Daten laufen jetzt über eine Events-Tabelle und einen Endpoint, wodurch Fehleranalysen stabiler werden.

Die NoyesStorage API erhält mehrere Verbesserungen und neue Funktionen; API-Filter (iLike) verarbeiten nun auch leere Werte zuverlässig, sodass Suchergebnisse konsistent bleiben.

6.15 v4.24 (7173867ca) ab 25.07.2025

NoyesStorage API Alpha

Early-Access-Kunden erhalten Alpha-Zugang zur neuen NoyesStorage API. API-Keys und Kern-Endpunkte stehen bereit, um erste Integrationen aufzusetzen und Abläufe extern anzubinden.

6.16 v4.23 (cfb64c84f) ab 16.07.2025

Sicherer Umgang mit Inaktivität

Nach 60 Minuten Inaktivität wird die UI verschwommen; ein Touch reaktiviert die Anzeige, danach ist erneut eine Anmeldung nötig.

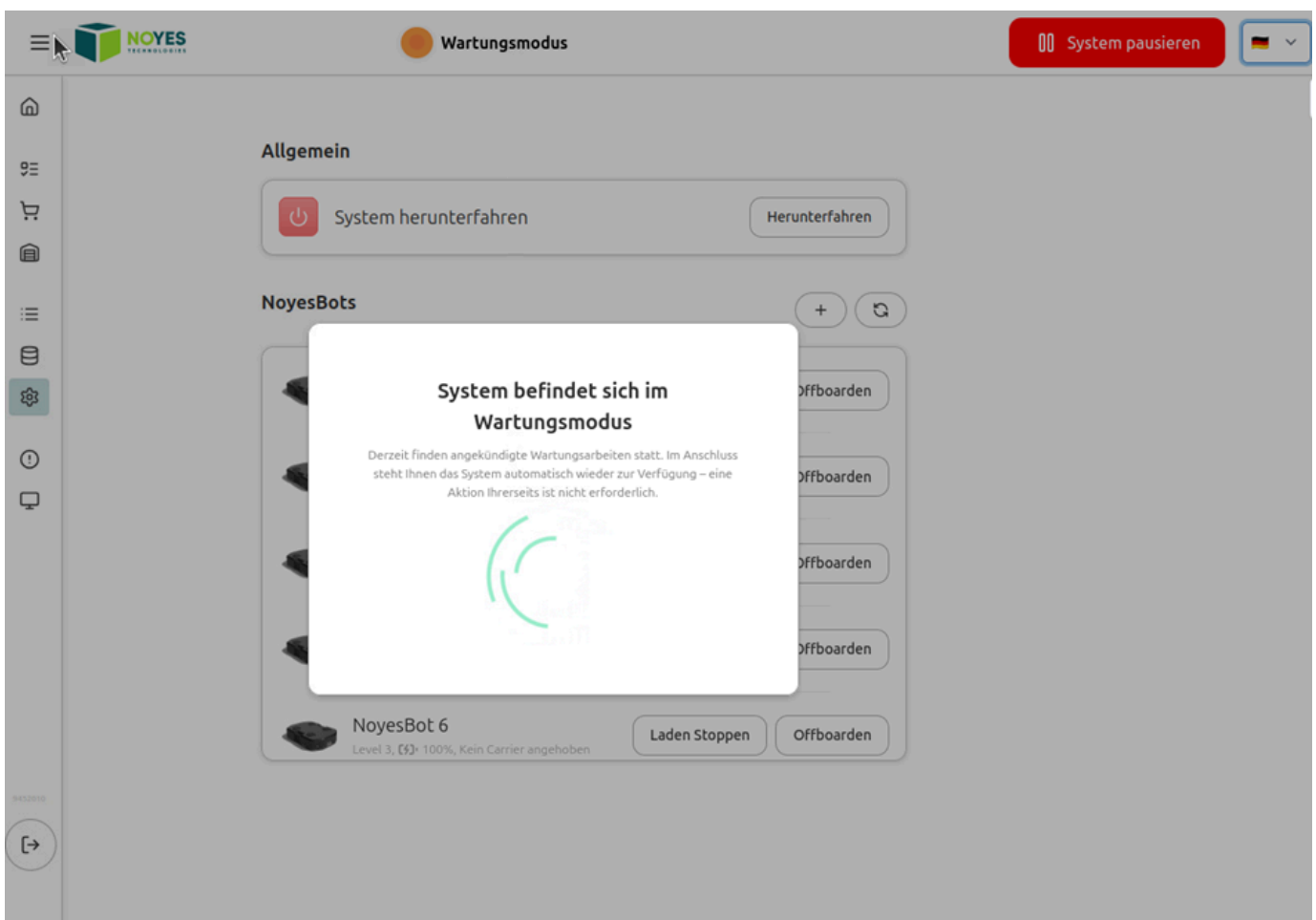
Schnellerer Neustart bei Bot-Fehlern (Bots im Fehlerzustand werden nicht zum Laden geschickt) und mehr Stabilität bei Requests mit vielen Positionen.

Dashboard-Sortierung und Job-Auswahl funktionieren jetzt auch mit Balkon-Filtern.

6.17 v4.22 (94520107f) ab 10.07.2025

Einführung des Wartungsmodes

Mit diesem Update führen wir einen neuen Systemstatus ein: den „Wartungsmodus“. Neben den bereits bestehenden Statusmeldungen „Aus“, „Pause“, „In Betrieb“, „Wird heruntergefahren“ und „Wird gestartet“, informiert der neue Wartungsmodus den Nutzer künftig direkt über laufende Wartungsarbeiten am System.



Der aktuelle Status wird wie gewohnt in der StorageUI angezeigt. Während sich das System im Wartungsmodus befindet, ist die Nutzung vorübergehend blockiert. So werden alle Nutzer – auch diejenigen, die nicht per E-Mail über geplante Wartungen informiert wurden – unmittelbar über den Systemstillstand informiert.

Sobald die Wartungsarbeiten abgeschlossen sind und der Status wieder auf „In Betrieb“ wechselt, steht das System wieder uneingeschränkt zur Verfügung.

6.18 v4.21 (9208edcc1) ab 12.06.2025

Anzeige der Carrier-Auslastung pro SKU

In den SKU-Stammdaten finden Sie nun die neue Spalte `Carrier Auslastung`. Diese zeigt, wie viele Carrier eine SKU konsolidiert belegt.

NOYES TECHNOLOGIES In Betrieb System pausieren 🇩🇪

SKU Stammdaten

Eine Stock Keeping Unit (SKU) ist ein eindeutiger Artikel in Ihrem Inventar. Sie kann benutzt werden um Lagerbestände über Ein- und Auslagerungen zu verfolgen. Hier können SKU Stammdaten verwaltet werden. Sie können SKUs erstellen, aktualisieren und löschen. Nutzen Sie Filter, um bestimmte SKUs schnell zu finden.

Suchen... Alle Filter löschen Tabelle exportieren + Hinzufügen Bearbeiten Löschen

<input type="checkbox"/>	ID	Name	Einheit	Auf Lager	Nachfüllbar	Bei Leerstand	Carrier Auslastung
<input type="checkbox"/>	chris-3	Stones	Stück (Stk.)	99	7	Zuweisung aufheben	7
<input type="checkbox"/>	MAR-02	MERCH - Coffee cup big (togo)	Stück (Stk.)	799	151	Nichts tun	3
<input type="checkbox"/>	H000031 [A]	Noyes Storage - Levelling Foot	Stück (Stk.)	212	9	Zuweisung aufheben	2
<input type="checkbox"/>	H001035 [A]	ARCHIVED - Charging Station assembly	Stück (Stk.)	200	0	Zuweisung aufheben	2
<input type="checkbox"/>	OPS-24	Installation - DEWALT Tool box	Stück (Stk.)	1	1	Nichts tun	2
<input type="checkbox"/>	H000016 [A]	Noyes Storage - Srew M6x25 (Floorplates)	Stück (Stk.)	5000	1000	Zuweisung aufheben	1.5
<input type="checkbox"/>	OPS-03	Installation - Festool Akkuschauber	Stück (Stk.)	2	1	Nichts tun	1.5
<input type="checkbox"/>	124	Merch - Turnbeutel	Stück (Stk.)	152	48	Zuweisung aufheben	1
<input type="checkbox"/>	Cleaning bot	cleaning bot	Stück (Stk.)	1	0	Zuweisung aufheben	1

Angezeigte Elemente: 100 / 1577 - Ausgewählt: 0 Zeile(n)

Beispiel: Eine Auslastung von 1,5 bedeutet, dass die SKU in Summe 1,5 Stellplätze benötigt – je nach Boxgröße können das in diesem Beispiel 2-12 Boxen sein.

Anzeige der Lagerplatzgüte auf der Inventarseite

Auf der Inventarseite wird eine neue Spalte **Lagerplatzgüte** eingeführt. Diese zeigt an, wie schnell ein Carrier ausgelagert werden kann – sortiert wird automatisch nach der niedrigsten Güte (d. h. dem schnellstmöglichen Carrier). Die Güte basiert auf einer internen Bewertung und ermöglicht eine deutlich bessere Einschätzung um z.B. den nächsten leeren Carrier anzufordern.

NOYES TECHNOLOGIES In Betrieb System pausieren DE

Inventar

Diese Tabelle bietet eine Echtzeitübersicht über Boxen, Träger und Lagerbestände im NoyesStorage. Sie können Träger für SKU identifizieren, die nachgefüllt werden müssen, oder solche mit Boxen ohne zugewiesene Ladungen und einen Request erstellen, um diese Träger abzurufen.

Suchen... Alle Filter löschen Tabelle exportieren Carrier anfordern

<input type="checkbox"/>	Carrier ID	Box ID	SKU ID	SKU Name	Einheit	Level	Tiefe	Lagerplatzgüte
<input type="checkbox"/>	340	947	H001184 [A]	Tablet power adapter (POE)	Stück (Stk.)	1	1	0
<input type="checkbox"/>	348	928	H001166- [A]	Top cap assembly	Stück (Stk.)	1	1	0
<input type="checkbox"/>	348	949	INT-15	Extension Cord (Verlängerungskabel)	Stück (Stk.)	1	1	0
<input type="checkbox"/>	348	-	-	-	-	1	1	0
<input type="checkbox"/>	376	1013	NBC043 [A]	Noyes Bot - Red internal DC cable	Stück (Stk.)	2	1	0
<input type="checkbox"/>	376	1032	H000080 [B]	Top Cap (old revision)	Stück (Stk.)	2	1	0
<input type="checkbox"/>	376	1012	NBC042 [A]	Noyes Bot - Black internal DC cable	Stück (Stk.)	2	1	0
<input type="checkbox"/>	376	897	H000080 [C]	Noyes Storage - Top cap	Stück (Stk.)	2	1	0

Angezeigte Elemente: 100 / 152 - Ausgewählt: 0 Zeile(n)

Über die API ist die Lagerplatzgüte des Carriers über das Feld "ETA" abrufbar.

Schnellere Neustarts auch im Fehlerfall

Bots im Fehlerstatus werden beim Herunterfahren nicht mehr zur Ladestation geschickt. Das beschleunigt den Neustartprozess deutlich. *Bitte beachten Sie:* Ein Bot im Fehlerstatus muss weiterhin manuell neu gestartet werden, bevor das System wieder erfolgreich hochfahren kann.

Empfohlene Vorgehensweise: Bot ausschalten → System herunterfahren → Bot einschalten → System starten

Verbesserte Anzeige des Systemstatus

Zwei neue Systemstatus sorgen für mehr Transparenz:

- "Wird heruntergefahren"
- "Wird gestartet"

Diese sind in allen Benutzeroberflächen sichtbar. Der bisherige Status „Initialisierung“ entfällt und wird durch „Wird gestartet“ ersetzt. Die bekannten Status „Aus“, „Pause“ und „In Betrieb“ bleiben unverändert.

Fix einer bisherigen Funktionalität

Ein automatischer Logout der Benutzeroberfläche erfolgt wieder erst nach 24 Stunden Inaktivität.

Hinweis

Außerdem kleinere Optimierungen wie: Version im UI sichtbar, Carrier-Filter angepasst (Wildcard-Suche), API-Logging wurde erweitert.

6.19 v4.20 (00337a922) ab 15.05.2025

Verbessertes Verhalten im Request Dashboard

Im Request Dashboard wird der nächste fertige Job ab jetzt immer automatisch geöffnet. Das bislang initial notwendige Selektieren auf einen Job beim Öffnen des Dashboards entfällt.

Der aktuell bearbeitete Request bleibt ab jetzt in der Liste der fertigen Aufträge ganz oben, auch wenn ein neuer Request mit höherer Priorität hinzukommt. Dies verhindert für die Fertigstellung des aktuell bearbeiteten Requests nochmals scrollen zu müssen.

Sofern bei mehreren fertigen Jobs ein weiter unten stehender Job zur Bearbeitung gewählt wird, wird dieser automatisch an die erste Stelle der Liste gesetzt. Auch dies verringert nochmals scrollen zu müssen.

Weitere Verbesserungen

- Schnelleres Herunterfahren des Systems - Wartezeiten, sofern das System länger nicht heruntergefahren wurde, werden deutlich verringert.
- Verbesserte Tabellenfilter - Das exakt dem Suchbegriff entsprechende Ergebnis wird in der Auswahlliste ganz oben angezeigt.

API-Schnittstelle

Ab jetzt können neue SKUs ebenfalls automatisiert über die API-Schnittstelle angelegt werden. Bislang war dies nur über die Benutzeroberfläche möglich.

6.20 v4.19 (d338a651c) ab 29.04.2025

Verbesserte Funktionalitäten

- In der Spalte JobTyp der Request & Job Tabelle stehen ab jetzt zusätzliche Filtermöglichkeiten für eine verbesserte Suche zur Verfügung.
- Der Button um alle Tabelleneinträge der Inventarseite auf einmal auszuwählen, hat bisher nur Einträge auf vorgeladenen Seiten selektiert. Ab jetzt werden automatisch alle Tabelleneinträge berücksichtigt.

Einführung von Lagerstrategien

Ab jetzt stehen für ihren NoyesStorage zwei verschiedene Lagerstrategien zur Verfügung, die entscheiden, welche Box für die Bearbeitung eines Auftrags verwendet wird, sofern für eine SKU mehrere Boxen vorhanden sind:

- Bei der Strategie SPEED (bisheriger Standard) wird immer die am schnellsten erreichbare Box gewählt um einen maximalen Durchsatz zu erreichen.
- Bei der Strategie FIFO (First in first out) wird die Box gewählt, die zuerst mit dem jeweiligen Material gefüllt wurde.

Sofern sie ihre Lagerstrategie von SPEED zu FIFO wechseln möchten, wenden Sie sich bitte an ihre Noyes Ansprechpartner. Ansonsten ändert sich für sie nichts.

Zukünftig können weitere Lagerstrategien folgen. Wir informieren sie, sobald diese zur Verfügung stehen.

6.21 v4.18 (2f2ae4c89) ab 03.04.2025

Neue Funktionalitäten

SKU Stammdaten im Prozessverlauf ändern

Bei Picking und Refilling Jobs können ab jetzt SKU-Name, Maßeinheit und Verhalten bei Leerstand angepasst werden, wenn der Carrier auf dem Balkon steht.



Bestandsmengen in SKU Stammdaten

In der Tabelle SKU Stammdaten wurden zwei zusätzliche Spalten hinzugefügt. Damit ist auf einen Blick je SKU ersichtlich, wie viele Einheiten sich im NoyesStorage befinden und wie viel in die bestehenden Boxen nachgelagert werden kann.

Datenexport per Mail

Die Daten jeder Tabelle auf der Benutzeroberfläche ihres NoyesStorage können ab jetzt per Mail exportiert werden. Der Export kann im CSV oder XLSX Format an die Mailadresse des angemeldeten Benutzers oder jede andere Mailadresse erfolgen.

Automatischer Logout nach 24 Stunden Inaktivität

Nach 24 Stunden ohne Interaktion mit dem NoyesStorage wird der angemeldete Benutzer automatisch abgemeldet. Sie benötigen Benutzernamen und Passwort für die erneute Anmeldung.

Neue Benutzeraccounts

Mit diesem Release findet eine grundlegende Überarbeitung der Nutzerdatenverwaltung statt, weshalb die Zugangsdaten aller Nutzer einmalig neu erstellt werden.

Nach dem erfolgten Release ihres NoyesStorage, erhält jeder bestehende Nutzer eine Mail mit einem Einmalpasswort (das bisherige Passwort verliert seine Gültigkeit).

Bitte melden sie sich mit ihrem Benutzernamen (bleibt gleich) und dem Einmalpasswort an ihrem NoyesStorage an und vergeben Sie ein Passwort ihrer Wahl.

6.22 v4.16 (eeb87589d) ab 06.03.2025

Erweitertes Stammdatenmanagement über die Benutzeroberfläche

- Löschen von Materialstammdaten
- Nachträgliche Anpassung von Materialstammdaten
- Auswahl verschiedener Maßeinheiten (UoM) bei Erstellung oder nachträglicher Anpassung eines Materialstammdatensatzes
- Definition je SKU, ob nach Entnahme der letzten Einheit die SKU aus der Box gelöscht wird oder die SKU weiterhin der Box zugeordnet bleibt

Zielgerichtete Informationen in der Request & Jobs Liste

- Standardfilter auf kundenrelevante Transaktionen (FULFILLMENT, REPLENISHMENT und FETCH)
- Anzeige der Kunden-Auftragsnummer in der Spalte Request-ID sofern der Auftrag über die Schnittstelle des NoyesStorage geschickt wurde

Zusätzliche Spalte Content Code auf der Inventar Seite

Diese Information wird in bestimmten Fällen bei einer Integration des NoyesStorage über eine Schnittstelle benötigt. Sofern die Content Codes in ihrem bisherigen Prozess nicht verwendet werden, bleibt diese Spalte leer und muss auch zukünftig nicht gefüllt werden.

Fehlerbehebungen

Seit dem letzten Release wurden auf der Troubleshooting Seite keine Log-Daten mehr aufgeführt. Dies ist jetzt behoben.

Wichtige Information

Mit diesem Release wird die alte Benutzeroberfläche abgeschaltet. Es steht ab jetzt nur noch die neue Version zur Verfügung die sie bereits seit einigen Wochen kennen. Es ist sichergestellt, dass ihnen alle Funktionalitäten der alten Version zur Verfügung stehen.

Durch diese Änderung muss einmalig die verwendete URL der Tablet App geändert werden. Dies muss zwingend nach dem Release durchgeführt werden. Ansonsten ist der NoyesStorage nicht mehr betriebsbereit.

6.23 v4.15 (e77b9e6e4) ab 20.02.2025

Das Release beinhaltet kleinere Anpassungen und Fehlerbehebungen an der neuen Benutzeroberfläche.

Bitte fokussieren Sie weiterhin die Nutzung der neuen Benutzeroberfläche. In den kommenden Wochen ist geplant, die parallele Verfügbarkeit der alten Benutzeroberfläche zu beenden.

6.24 v4.14 (2d929340f) ab 23.01.2025

Neue Benutzeroberfläche

Erstmalig steht Ihnen die Beta-Version einer komplett überarbeiteten Benutzeroberfläche zur Verfügung. Die bestehenden Funktionalitäten ihres NoyesStorage bleiben erhalten, jedoch erwarten wir eine wesentlich intuitivere Bedienung als bislang.

Für einen beschränkten Zeitraum stehen die bisherige und die neue Benutzeroberfläche parallel zur Verfügung. Sie können jederzeit mit einem Knopfdruck zwischen beiden Versionen wechseln. In einigen Wochen wird die bisherige Benutzeroberfläche vollständig abgeschaltet.

Sie erhalten eine initiale Schulung sowie eine vollständige Dokumentation der neuen Benutzeroberfläche. Wir kontaktieren Sie hierfür separat.

Ein Wechsel auf die neue Version wird erst nach erfolgter Schulung empfohlen, um Störungen des Betriebsablaufs zu vermeiden.

Definierbares Verhalten nach vollständiger Entnahme einer SKU

Bislang wurde die Zuordnung einer SKU zu einer bestimmten Box automatisch aufgelöst, sobald die letzte Einheit entnommen wurde. Ab jetzt kann auf Ebene des Materials festgelegt werden, ob die Zuordnung zu einer definierten Box bestehen bleibt oder bei Entnahme der letzten Einheit aufgelöst wird.

Mit diesem Release steht die Funktion nur über eine gesonderte Website an einem PC im Netzwerk Ihres NoyesStorage zur Verfügung. Auf der Benutzeroberfläche des NoyesStorage wird dies erst mit einem der folgenden Releases möglich sein.

Nachträgliche Bearbeitung der Materialstammdaten

Perspektivisch ist geplant SKU-ID, SKU-Name und Einheit vollständig über die Benutzeroberfläche bearbeiten zu können. Mit dem jetzigen Release können Sie die Felder SKU-ID und SKU-Name über eine gesonderte Website an einem PC im Netzwerk Ihres NoyesStorage aktualisieren.

Sobald mit einem der nächsten Releases die vollständige Funktionalität zur Verfügung steht, informieren wir Sie erneut.

Verbesserte Funktionalitäten

Vollständig automatisierter Neustart des NoyesStorage nach einem ungeplanten Stromausfall.

6.25 v4.13 (7fb944a28) ab 21.11.2024

Sortierung fertiger Aufträge

Fertige Aufträge werden in der Ansicht der Sammelaufträge jetzt nach dem Erstelldatum sortiert. Wenn der Bediener immer den obersten Auftrag auswählt, werden die Ebenen im Falle einer Auftragsanhäufung optimal ausgelastet, wodurch der Systemdurchsatz effektiv gesteigert wird.

Entwicklungen ohne unmittelbare Auswirkungen für sie

- Verbesserter Deployment-Prozess
- Ermöglicht eine schnellere und zuverlässigere Implementierung zukünftiger Updates.

Fehlerbehebungen

- Bearbeitung von Boxen mit "None" als SKU ID
- Anzeige der Ebenenhöhe für Carrier ohne Ladung
- Korrekte Anzeige und Bearbeitungsmöglichkeit aller Boxgrößen
- Schnellerer Aufruf von fertigen Aufträgen in der Ansicht "Einzelauftrag"

7. About

7.0.1 Versionierung

Version	Datum	Wesentliche Änderung
V 1.0.0	24.03.2023	Erste Version Betriebsanleitung
V 2.0.0	24.06.2024	Aktualisierung der Betriebsanleitung auf neueste Bot-Generation und Softwareversion v4.7.0
V 2.1.0	27.08.2024	Aktualisierung der Betriebsanleitung auf Softwareversion v4.8.0
V 2.1.1	02.09.2024	Aktualisierung der Betriebsanleitung auf Softwareversion v4.8.1
V 2.2.0	21.01.2025	Grundlegende Überarbeitung Abschnitt 2 und Abschnitt 6, Neuer Abschnitt 5 zur Integration, Aktualisierung der Betriebsanleitung auf NoyesStorage v4.11 und NoyesBot v0.6
V 2.3.0	06.02.2025	Grundlegende Überarbeitung des Abschnitts 4 aufgrund Einführung neuer NoyesStorage App mit NoyesStorage v4.15

7.0.2 Haftungsausschluss

Dieses Handbuch wird von Noyes Robotics GmbH nur zu Referenzzwecken veröffentlicht. Dieses Dokument wird kontinuierlich aktualisiert, das heißt alle Produktangebote und Spezifikationen können ohne vorherige Ankündigung geändert werden. Noyes Robotics GmbH ist rechtlich nicht an die Details dieses Dokuments gebunden. Noyes Robotics GmbH haftet nicht für direkte, indirekte, besondere, zufällige oder Folgeschäden, die sich aus der Verwendung dieses Dokuments, des NoyesStorage oder aus der Verletzung von Rechten Dritter ergeben, die sich aus einer solchen Verwendung ergeben können.

7.0.3 Kommentare, Vorschläge und Korrekturen

Die Abteilung für Benutzerdokumentation würde gerne Ihre Meinung zu diesem Handbuch erfahren. Ihr Feedback hilft uns, unsere Dokumentation für Ihre individuellen Bedürfnisse zu optimieren. Schicken Sie uns Ihre Kommentare per E-Mail an: support@noyes-tech.com

7.0.4 Copyright und Handelsmarken

Dieses Dokument enthält urheberrechtlich geschützte Informationen. Alle Rechte vorbehalten.

Kein Teil dieser Publikation darf in irgendeiner Form oder mit irgendwelchen Mitteln, sei es elektrisch, durch Fotokopie, Aufzeichnung oder auf andere Weise, ohne vorherige schriftliche Genehmigung durch Noyes Robotics GmbH vervielfältigt werden. Die Rechte an allen Markennamen, Produktnamen und Warenzeichen liegen bei ihren jeweiligen Eigentümern.