INTEGRATIONSDOKUMENT

mit Use Cases

Inhalt

1 Zweck dieser Dokumentation

Diese Dokumentation dient als technische Grundlage für Softwareanbieter, welche die Galexis XML-Schnittstelle PIS/POS V2.0 in ihre Applikationen (z.B. PIS/POS-, ERP- oder Warenwirtschaftssysteme) integrieren möchten. Für jede Funktion sind zwei Nachrichten definiert. Eine für die Anfrage (Request) und eine für die Antwort (Response). Die Nachrichten werden im XML-Format ausgetauscht, das durch eine Schemadefinition (XSD) festgelegt ist. Die Nachrichten werden über eine der folgenden Optionen ausgetauscht:
HTTPS (SSL)-Protokoll POST-Anfragen mit Inhaltstyp "text/xml". Der Inhalt der Anfrage ist die XML-Nachricht selbst, es werden keine Umschläge (Envelope-Element) verwendet
HTTPS (SSL)-Protokoll POST einer SOAP-Anfrage. Funktionen werden über WSDL-Definitionen beschrieben.

Ziel ist es, eine vollautomatisierte Kommunikation mit dem Galexis ERP-System zu ermöglichen und so den Bestell-, Verfügbarkeits- und Lieferprozess für Apotheken, Drogerien und andere Kunden effizienter zu gestalten.

Zielgruppe im Detail: Softwareanbieter und IT-Dienstleister im Gesundheitsmarkt – insbesondere Entwicklerteams für PIS/POS-, Warenwirtschafts- und ERP-Systeme sowie technische Projektleiter bei Apotheken-, Drogerie- und Softwareanbietern usw.
Link WSDL und Schemas: https://xml.e-galexis.com/V2/

2 Vorteile der XML-Schnittstelle PIS/POS V2.0

- Echtzeit-Abfragen für Preis, Verfügbarkeit und Lieferrückstände
- Integrierte Bestellprozess Abwicklung von der Abfrage, über Warenkorb zu Lieferschein der alle Charge- und Verfall-Daten, sowie Mengen und Preise der gelieferten Produkte beinhaltet
- Direkte Integration von kundenspezifischen Konditionen
- Flexible Kommunikation via HTTPS (text/xml oder SOAP)

3 Wie funktioniert der Bestellprozess über Galexis?

Die durchgängigen Methoden ermöglichen dem PIS/POS-Benutzer eine effiziente, transparente und sichere Abwicklung vom Produktcheck über Warenkorberstellung und -verwaltung bis hin zur Bestellung und Lagerverwaltung.

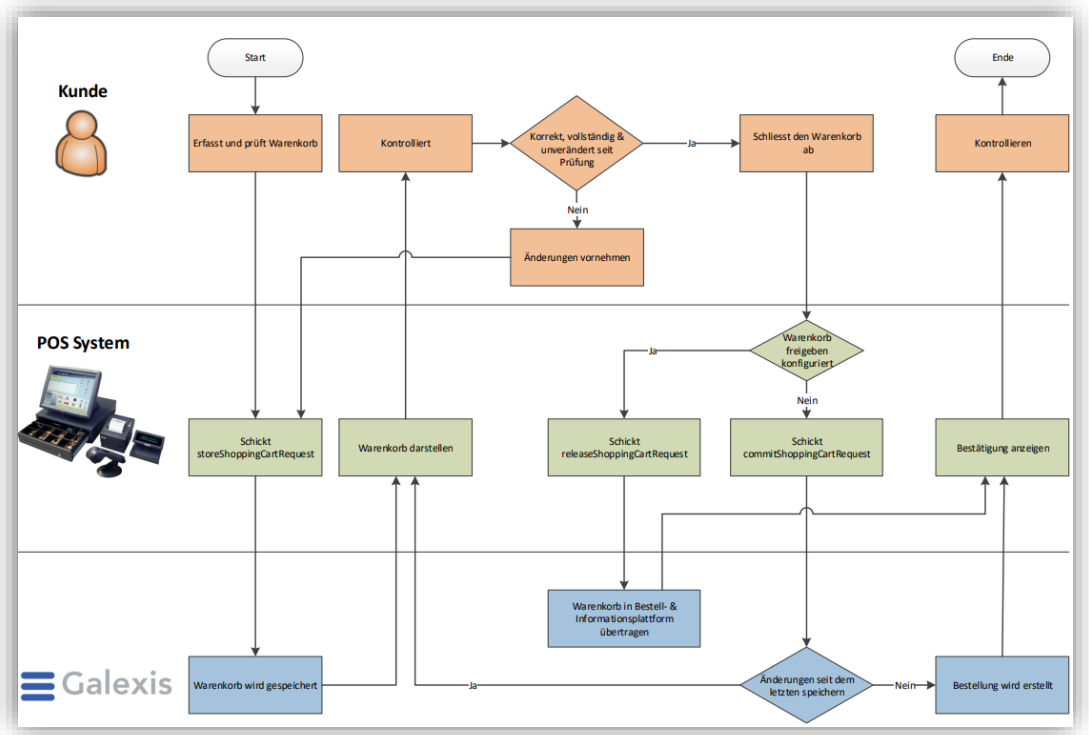


Abbildung 1: Workflow Bestellabwicklung

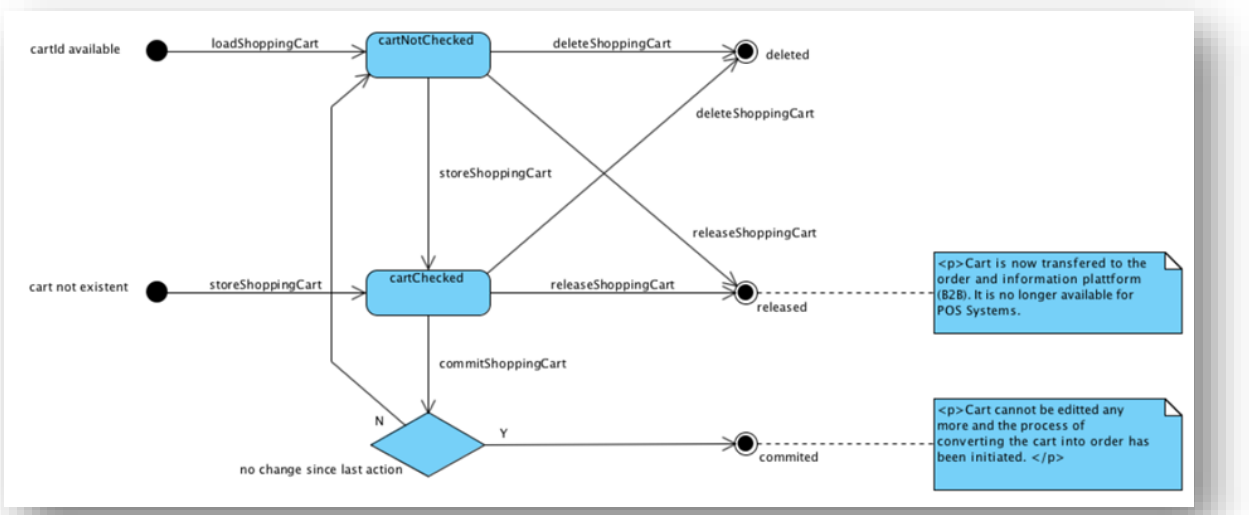


Abbildung 2: Zustände Warenkorb

3.1 Warenkorb im PIS/POS-System erfassen und prüfen

Der PIS/POS-Benutzer erfasst zunächst seinen Warenkorb im PIS/POS-System und prüft diesen auf Vollständigkeit und Korrektheit. Diese Funktionalität wird durch das PIS/POS-System zur Verfügung gestellt.

3.2 Warenkorb speichern und prüfen

Anschliessend sendet das PIS/POS-System den Warenkorb mittels der Methode storeShoppingCartRequest an Galexis. Dabei überprüft Galexis alle Positionen hinsichtlich Verfügbarkeit, Bezugsberechtigung und korrekter Preisbildung und liefert das Ergebnis über storeShoppingCartResponse zurück. So kann der PIS/POS-Benutzer sicher sein, dass der Warenkorb gültige und aktuelle Daten enthält.

3.3 Warenkorb laden und bearbeiten

Sollte der Warenkorb noch nicht endgültig abgeschlossen sein, kann der PIS/POS-Benutzer diesen jederzeit mit loadShoppingCartRequest wieder laden, um Anpassungen vorzunehmen oder den aktuellen Status zu prüfen. Die Antwort von Galexis erfolgt über loadShoppingCartResponse und zeigt den aktuellen Warenkorbinhalt.

3.4 Warenkorb löschen

Für den Warenkorb gibt es zusätzlich die Möglichkeit, einen gespeicherten (aber noch nicht übermittelt) Warenkorb zu löschen mittels DeleteShoppingCart, falls dieser nicht mehr benötigt wird. Diese Methode ermöglicht es dem PIS/POS-Benutzer, den Warenkorb aus dem System zu entfernen und somit für Klarheit und Ordnung in der Bestellung zu sorgen.

3.5 Warenkorb freigeben (ctSendX)

Für die weitere Bearbeitung im zentralen Kundenportal (e-Galexis) kann der PIS/POS-Benutzer den Warenkorb mit releaseShoppingCartRequest freigeben. Galexis bestätigt die Freigabe mit releaseShoppingCartResponse, wodurch der Warenkorb im Portal bearbeitbar wird.

Diese Funktion wurde speziell für Arztpraxen konzipiert, in denen mehrere MPAs über GaliPhone Warenkörbe erfassen. Der Arzt kann diese später in e-Galexis konsolidieren und daraus eine definitive Bestellung erstellen.

3.6 Warenkorb abschliessen und Bestellung auslösen

Ist der Warenkorb korrekt, vollständig und übermittelt, löst der PIS/POS-Benutzer mit commitShoppingCartRequest die Bestellung aus. Galexis bestätigt den erfolgreichen Abschluss mit commitShoppingCartResponse. Sollte die Bestellung nicht erfolgreich sein oder sich Konditionen geändert haben, erhält der PIS/POS-Benutzer eine entsprechende Rückmeldung, um Korrekturen vorzunehmen.

3.7 Produktverfügbarkeit abfragen

Zusätzlich möchte der PIS/POS-Benutzer jederzeit die Produktverfügbarkeit, Preise und Details sicherstellen können. Dazu sendet er eine productAvailabilityRequest, die über eine Antwort von Galexis mittels productAvailabilityResponse verlässliche Informationen zur aktuellen Verfügbarkeit und Preisgestaltung liefert.

3.8 Konditionen einsehen

Um stets die individuellen Galexis-Konditionen (KuKo's) und Rabatte einzusehen, nutzt der PIS/POS-Benutzer die Methode customerSpecificConditionsRequest. Die Antwort customerSpecificConditionsResponse gibt ihm Einsicht in seine Preisgestaltung. Die Galexis-Kundenkonditionen müssen nicht von Hand gepflegt werden, sondern werden automatisch durch die Schnittstelle zur Verfügung gestellt.

3.9 Rückkäufe registrieren

Rückkäufe sind Verkäufe an Galexis, für die Ware ausserhalb des Retouren Prozesses (innerhalb 10 Tage[1]). Produkte können aufgrund eines Chargen-Rückruf, eines Lieferanten initiierten Rücknahme oder einer Lagerbereinigung beim Kunden, an Galexis zum Rückkauf angeboten werden. Rückkaufsdaten erfassen funktioniert analog der Warenkorb Funktionalität. Zuerst werden die Rückkäufe über storeReturnCart gespeichert und abschliessend mit commitReturnCartRequest abgeschlossen. Die von der Galexis nicht akzeptierten Rückkäufe können über deleteReturnCartRequest gelöscht werden. Ob und zu welchen Konditionen die Produkte zurückgekauft werden, ist jeweils in der Antwort storeReturnCartResponse ersichtlich.

[1] Es gilt die Swissmedic Verordnung : https://www.swissmedic.ch/swissmedic/fr/home/news/mitteilungen/neue-techn-interpretation-ti-i-smi-ti-28.html

3.10 Wareneingang automatisiert verarbeiten (Lieferschein & Box Download)

Lieferungen verarbeitet der PIS/POS-Benutzer automatisiert über deliveryNoteDownloadRequest. Diese Methode ermöglicht die Abfrage basierend auf Lieferscheinnummer oder Boxnummer. Die Antwort von Galexis mittels deliveryNoteDownloadResponse liefert umfassende Details wie Produkt-Charge, Menge, Verfallsdatum, Preisangaben, logistische Kosten, Gesamtsummen und mehr.

3.11 Offene Rückstände (Backlog) abrufen

Offene Rückstände sind bestellte Zeilen, für die Galexis noch eine Warenanlieferung erwartet um sie dem Kunden auszuliefern. Diese Rückstände können normale Lagerware betreffen (NOTA), oder auch die Ware die Galexis nicht am Lager hält, aber für den Kunden beim Lieferant besorgen kann (Besorger). Um offene Bestellrückstände abzurufen, nutzt der PIS/POS-Benutzer die Methode openBacklogRequest. Galexis gibt detaillierte Informationen zu ausstehenden Bestellrückstände zurück mittels openBacklogResponse.

3.12 Inventur durchführen

Um Inventur durchzuführen, startet PIS/POS-Benutzer eine Inventur mit getInventoryIdRequest und erhält eine Inventur-ID via getInventoryIdResponse. Zwischenergebnisse übermittelt er mit inventoryTransmissionRequest, bestätigt durch inventoryTransmissionResponse. Den Abschluss der Inventur meldet er mit closeInventoryRequest, bestätigt durch closeInventoryResponse. Der Status der Inventur wird mit inventoryStatusRequest abgefragt und durch inventoryStatusResponse zurückgemeldet. Für eine effiziente Erfassung unterstützt das System auch einen Offline-Barcodescanner mit Daten aus getProductBaseMasterDataDiff. Das heisst zuerst werden Offline Daten in die Scanner geladen und danach kann die Inventur offline durchgeführt werden.

Nach Abschluss der Inventur, erstellt Galexis die Liste der gezählten Produkte und Mengen ergänzt mit den Einkaufspreisen und stellt es in CSV und/oder PDF Format dem Kunden zu.

3.13 Pakete an andere Galexis-Kunden versenden

Abschliessend kann der PIS/POS-Benutzer Informationen abfragen um eine Parcel-Etikette zu erstellen. Mit dieser können Pakete über Galexis an andere Galexis-Kunden versendet werden.

4 Übersicht Abgeleitete Use Cases

Use Case	Beschreibung	Methoden
Warenkorb erfassen (PIS/POS seitig)	Der PIS/POS-Benutzer erfasst im PIS/POS-System seinen Warenkorb und überprüft ihn auf Vollständigkeit und Korrektheit. Diese Funktionalität liegt im PIS/POS-System.	*storeShoppingCartRequest* *storeShoppingCartResponse*
Warenkorb speichern	Das PIS/POS-System sendet den Warenkorb mittels storeShoppingCartRequest an Galexis. Galexis prüft Verfügbarkeit, Bezugsberechtigung und Preise und gibt das Ergebnis über storeShoppingCartResponse zurück. So erhält der PIS/POS-Benutzer gültige und aktuelle Daten zum Warenkorb.	*storeShoppingCartRequest* *storeShoppingCartResponse*
Warenkorb laden und bearbeiten	Wenn der Warenkorb noch nicht abgeschlossen ist, kann der PIS/POS-Benutzer ihn jederzeit mit loadShoppingCartRequest laden, um Anpassungen vorzunehmen oder den aktuellen Status zu prüfen. Die Antwort erfolgt über loadShoppingCartResponse.	*loadShoppingCartRequest* *loadShoppingCartResponse*
Warenkorb löschen	Der PIS/POS-Benutzer kann einen gespeicherten Warenkorb mit deleteShoppingCartRequest löschen, wenn dieser nicht mehr benötigt wird. Galexis bestätigt dies mit deleteShoppingCartResponse.	*deleteShoppingCartRequest* *deleteShoppingCartResponse*

Warenkorb freigeben (ctSendX)	Zur weiteren Bearbeitung im e-Galexis Kundenportal kann der Warenkorb mit releaseShoppingCartRequest freigegeben werden. Die Bestätigung erfolgt durch releaseShoppingCartResponse, wodurch der Warenkorb im Portal bearbeitbar wird.	*releaseShoppingCartRequest* *releaseShoppingCartResponse*
Warenkorb abschliessen (Bestellung auslösen)	Ist der Warenkorb vollständig, löst der PIS/POS-Benutzer die Bestellung mittels commitShoppingCartRequest aus. Galexis bestätigt den erfolgreichen Abschluss mit commitShoppingCartResponse. Bei Fehlern oder geänderten Konditionen wird direkt im Response eine Meldung mitgegeben sowie der Inhalt des Warenkorb damit dieser erneut gespeichert und abgeschlossen werden kann.	*commitShoppingCartRequest* *commitShoppingCartResponse*
Produktverfügbarkeit prüfen	Zur schnellen Auskunft über Produkte sendet der PIS/POS-Benutzer eine productAvailabilityRequest. Galexis liefert über productAvailabilityResponse aktuelle Informationen zu Verfügbarkeit, Preisen und Produktdetails.	*productAvailabilityRequest* *productAvailabilityResponse*
Konditionen einsehen	Der PIS/POS-Benutzer ruft seine individuellen Galexis-Konditionen und Rabatte mit customerSpecificConditionsRequest ab. Die Antwort customerSpecificConditionsResponse zeigt ihm die persönlichen Preisgestaltungen.	*customerSpecificConditionsRequest* *customerSpecificConditionsResponse*

Rückkäufe registrieren	Für Rückkäufe nutzt der PIS/POS-Benutzer die storeReturnCart Funktionalität. Diese funktioniert analog der Warenkorb Funktionalität. Zuerst werden die Rückkäufe über storeReturnCart gespeichert und abschliessend mit commitReturnCartRequest abgeschlossen. Fehlerhafte Rückkäufe können über deleteReturnCartRequest gelöscht werden.	**Rückkäufe eröffnen** *storeReturnCartRequest* *storeReturnCart-Response* **Rückkäufe abschliessen** *commitReturnCartRequest* *commitReturnCart-Response* **Rückkäufe löschen** *deleteReturnCartRequest* *deleteReturnCart-Response*
Wareneingang automatisiert verarbeiten	Der PIS/POS-Benutzer kann Lieferungen anhand von Lieferscheinnummer, Datum oder Boxnummer mit deliveryNoteDownloadRequest abfragen. Die Antwort deliveryNoteDownloadResponse enthält Details wie Charge, Menge, Verfallsdatum, Preise, logistische Kosten, Rabatte und Gesamtinformationen.	deliveryNoteDownloadRequest deliveryNoteDownloadResponse
Offene Rückstände (Backlog) abrufen	Um den Status offener Bestellrückstände zu überwachen, nutzt der PIS/POS-Benutzer openBacklogRequest. Galexis antwortet mit openBacklogResponse, die detaillierte Informationen zu offenen Bestellrückständen liefert.	openBacklogRequest openBacklogResponse

Inventur durchführen	Der PIS/POS-Benutzer startet eine Inventur mit getInventoryIdRequest und erhält die Inventur-ID via getInventoryIdResponse. Zwischenergebnisse übermittelt er mit inventoryTransmissionRequest (Bestätigung: inventoryTransmissionResponse). Den Abschluss meldet er mit closeInventoryRequest (closeInventoryResponse). Den aktuellen Status fragt er mit inventoryStatusRequest ab (Antwort: inventoryStatusResponse). Für eine effiziente Erfassung unterstützt das System einen Offline-Barcodescanner mit Daten aus getProductBaseMasterDataDiff. Offline-Daten werden zuerst in den Scanner geladen, danach erfolgt die Inventur offline.	**Inventur Starten** getInventoryIdRequest getInventoryIdResponse **Inventurzwischenstand senden** inventoryTransmissionRequest inventoryTransmissionResponse **Inventur abschliessen** closeInventoryRequest closeInventoryResponse **Inventurstatus abfragen** inventoryStatusRequest inventoryStatusResponse **Offline-Barcodescanner Daten** getProductBaseMasterDataDiff
Paket versenden (Parcel)	Der PIS/POS-Benutzer kann die Informationen abfragen um eine Parcel-Etikette zu erstellen. Mit dieser können Pakete über Galexis an andere Galexis-Kunden versendet werden.	getAddressLabelsRequest getAddressLabelsResponse

4.1 Warenkorb speichern

Der POS-Mitarbeiter erfasst im POS-System einen Warenkorb mit Produkten, der dann an Galexis übermittelt wird. Galexis prüft die Produkte im Warenkorb hinsichtlich Verfügbarkeit, Bezugsberechtigung und korrekter Preisbildung und gibt eine Rückmeldung. So wird sichergestellt, dass der Warenkorb valide ist, bevor er abgeschlossen wird.

4.1.1 XML-Beispiel Request zu storeShoppingCartRequest

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/storeShoppingCartRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Der POS-Mitarbeiter übermittelt den Warenkorb mit allen Produktpositionen an Galexis. Jedes Produkt wird durch einen eindeutigen Pharmacode identifiziert, zusammen mit der gewünschten Menge, dem Positionstyp und der Lieferzone. Das gewünschte Lieferdatum wird im Header definiert, um die Auslieferung zu planen. So erhält Galexis alle nötigen Daten, um den Warenkorb zu prüfen.

Das Hauptelement `<storeShoppingCartRequest>` enthält nebst den technischen Attributen für die schemaLocation und den Namespace folgende Attribute:

- version: Version der Meldung: 2.0 ist zu verwenden
- language: Sprache in welcher z.B. die Produktbeschreibung zurückgeliefert wird. Wird keine Sprache übermittelt, wird die Sprache aus dem Kundenstamm verwendet.
- communicationSoftwareId: Identifikation der Software welche den Request übermittelt hat.

Das Unterelement `client` enthält die Authentifizierungsdaten (Webclient-Id und Passwort).
Das `cartIdentifier` Element enthält die Identifikation des Warenkorbes. Wird ein neuer Warenkorb erstellt, muss das Element nicht mitgeliefert werden. Wird ein bestehender Warenkorb verändert, muss die Identifikation des Warenkorbes mitgegeben werden.
Der `<cartHeader>` enthält Metadaten zum Warenkorb, hier z.B. das gewünschte Lieferdatum.

- <cartLines>: Die Liste der einzelnen Positionen im Warenkorb.
- <cartLine>: Eine einzelne Position im Warenkorb.
- cartLineNumber Positionsnummer
- orderQuantity bestellte Menge
- cartLineType Art der Position z.B. Ware, Etiketten oder beides. Bei Etiketten werden die in orderQuantity gewünschte Menge an Etiketten geliefert. Bei Ware und Etiketten wird die Ware in der gewünschten Menge geliefert und eine Etikette.
- deliveryZone Liefergebiet, erlaubt leer, CUSTOMER, FRDGE, FRIGO, LSAB, NARCOTIC, OTC, PERFUMERY, ROBOT, STUPEFIANTS, URGENT, VERIFY.
- <product>: Das Produkt, das bestellt wird. Dabei kann das gewünschte Produkt über die drei Typen <pharmaCode>, <EAN> oder <supplierProductNumber> identifiziert werden.

4.1.2 XML-Beispiel Response zu storeShoppingCartResponse

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/storeShoppingCartResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Galexis bestätigt den Empfang und validiert jede Position. Für jede Produktzeile wird die Verfügbarkeit, die tatsächlich lieferbare Menge und der gültige Preis zurückgemeldet. Zudem gibt es eine Gesamtsumme ohne MwSt. und geschätzte Lieferkosten. So bekommt der POS-Mitarbeiter eine klare Übersicht über den aktuellen Warenkorbstatus.

- <storeShoppingCartResponse>: Das Hauptelement der Antwort mit Namespace, Version und Sprache.
- <clientResponse>: Bestätigung der Kundendaten, die in der Anfrage übermittelt wurden.
- <shoppingCart>: Informationen zum verarbeiteten Warenkorb.
    - <cartIdentifier>: Die eindeutige Warenkorb-ID welche für weitere Aktionen, z.B. Warenkorb abschliessen, benötigt wird. Zudem eine für Menschen lesbare Nummer des Warenkorbs, die intern bei Galexis genutzt wird.

- <cartHeaderResponse>
  - deliveryDate: Bestätigtes Lieferdatum

— <cartLinesResponse>: Die Liste der Rückmeldungen zu jeder einzelnen Warenkorbposition.

- <cartLineResponse>: Für jede Position wird hier angezeigt:
  - lineAccepted: Ob die Position akzeptiert wurde (true/false).
  - deliveryZone: Bestätigte Lieferzone.
  - productResponse: Details zum Produkt wie Beschreibung, Basispreis, gelieferte Menge und fakturierte Menge.
  - <availability>: Verfügbarkeitsstatus des Produkts (z.B. "yes" = verfügbar).
- <cartTotals>: Zusammenfassung der Kosten im Warenkorb:
  - totalAmountExclVATExclExpLogServiceCosts: Gesamtbetrag ohne Mehrwertsteuer und ohne logistische Zusatzkosten.
  - totalExpectedLogServiceCosts: Erwartete logistische Zusatzkosten.

4.2 Warenkorb laden und bearbeiten

Der POS-Benutzer möchte einen zuvor gespeicherten Warenkorb aus Galexis laden, um dessen Inhalt einzusehen, zu prüfen oder bei Bedarf anzupassen. Das POS-System sendet eine Anfrage an Galexis, um die aktuelle Version des Warenkorbs abzurufen.

4.2.1 XML-Beispiel Request zu loadShoppingCartRequest

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/loadShoppingCartRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Der POS-Benutzer sendet die Anfrage mit Authentifizierungsdaten, um den aktuell gespeicherten Warenkorb zu laden. Optional kann angegeben werden, welcher Warenkorb geladen werden soll (meist über eine ID). Dabei fordert er den Inhalt des Warenkorbs inklusive aller Positionen an.

Das Hauptelement <loadShoppingCartRequest> enthält die Authentifizierungsinformationen des POS-Benutzers. Galexis identifiziert den Kunden und lädt den aktuellen Warenkorb aus dem System. Das <cartIdentifier> Element enthält die WarenkorbId welche geladen werden soll.

4.2.2 XML-Beispiel Response zu loadShoppingCartResponse (erfolgreich)

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/loadShoppingCartResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Galexis liefert eine Antwort mit den vollständigen Details des Warenkorbs, einschliesslich aller Positionen mit Produktinformationen, Preisen und Verfügbarkeitsstatus.

Das Hauptelement <loadShoppingCartResponse> bestätigt den Kunden und den geladenen Warenkorb. Innerhalb von <cartLinesResponse> sind alle Warenkorbpositionen mit Details zum Produkt, Preis, Menge und Verfügbarkeit aufgeführt. Die <cartTotals> geben eine Zusammenfassung der Kosten ohne MwSt. und erwarteter logistischer Zusatzkosten. So kann der POS-Benutzer den aktuellen Status seines Warenkorbs exakt nachvollziehen und bei Bedarf Änderungen vornehmen.

4.3 Warenkorb löschen

Der POS-Benutzer möchte einen gespeicherten Warenkorb löschen, wenn dieser nicht mehr benötigt wird. Dies schafft Klarheit im System und verhindert versehentliche Bestellungen. Das POS-System sendet dazu eine Löschanfrage an Galexis, welche den Warenkorb mit der angegebenen ID aus dem System entfernt.

4.3.1 XML-Beispiel Request zu deleteShoppingCartRequest

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/deleteShoppingCartRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Der POS-Benutzer identifiziert den zu löschenden Warenkorb über seine eindeutige ID und authentifiziert sich gegenüber Galexis.

Das Hauptelement `<deleteShoppingCartRequest>` enthält die Authentifizierungsdaten des POS-Benutzers sowie das `<cartIdentifier>` mit der eindeutigen ID des Warenkorbs, der gelöscht werden soll. Galexis verwendet diese Information, um den richtigen Warenkorb zu identifizieren und zu entfernen.

4.3.2 XML-Beispiel Response zu deleteShoppingCartResponse (erfolgreich)

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/deleteShoppingCartResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Galexis bestätigt die erfolgreiche Löschung des Warenkorbs.

Das Hauptelement `<deleteShoppingCartResponse>` bestätigt die Kunden- und Warenkorbdetails. Das Vorhandensein des Elements `<successful/>` zeigt an, dass der Warenkorb

erfolgreich gelöscht wurde. Der POS-Benutzer erhält somit eine klare Rückmeldung über den Abschluss der Löschaktion.

4.4 Warenkorb freigeben (ctSendX)

Der POS-Benutzer möchte einen gespeicherten Warenkorb für die weitere Bearbeitung im zentralen Kundenportal (e-Galexis) freigeben. Die Freigabe ermöglicht es, dass der Warenkorb dort sichtbar und editierbar wird, z.B. für andere Berechtigte oder im Rahmen von Freigabeprozessen.

4.4.1 XML-Beispiel Request zu releaseShoppingCartRequest

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/releaseShoppingCartRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Der POS-Benutzer identifiziert den freizugebenden Warenkorb per ID und authentifiziert sich gegenüber Galexis.

Das Hauptelement <releaseShoppingCartRequest> enthält die Authentifizierungsdaten und die eindeutige ID des Warenkorbs, der für das Kundenportal freigegeben werden soll. Galexis verwendet diese Information, um die Freigabe im System zu vermerken.

4.4.2 XML-Beispiel Response zu releaseShoppingCartResponse (erfolgreich)

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/releaseShoppingCartResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Galexis bestätigt die erfolgreiche Freigabe des Warenkorbs.

Das Antwort-Element `<releaseShoppingCartResponse>` bestätigt die Freigabe mit den Kundendaten und Warenkorb-Identifikationen. Das Element `<successful>` mit einer entsprechenden Nachricht zeigt an, dass der Warenkorb nun im Kundenportal bearbeitbar ist.

4.5 Warenkorb abschliessen (Bestellung übermitteln)

Der POS-Benutzer möchte nach Prüfung und Freigabe den Warenkorb abschliessen und damit die Bestellung bei Galexis übermitteln. Dabei wird der komplette Warenkorb an Galexis übermittelt, um die Bestellung verbindlich zu platzieren.

4.5.1 XML-Beispiel Request zu commitShoppingCartRequest

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/commitShoppingCartRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Der POS-Benutzer sendet eine Anfrage zur Bestellung mit der eindeutigen Warenkorb-ID und Authentifizierungsdaten.

Das Hauptelement `<commitShoppingCartRequest>` beinhaltet neben dem Namespace, der Version Authentifizierungsinformationen und die ID des Warenkorbs, der bestellt werden soll. Die Attribute `productDescriptionDesired` und `compressionDesired` steuern, ob

Produktbeschreibungen in der Antwort erwünscht sind und ob die Daten komprimiert übertragen werden sollen.
Der PIS/POS-Kunde kann die Informationen *deliverZone* sowie *cartIdentifier* übermitteln, sofern er möchte, dass diese bei der Wareneingangsbuchung mit der *delivery-NoteDownloadResponse* zurückgesendet werden.

4.5.2 XML-Beispiel Response zu commitShoppingCartResponse (erfolgreich)

XSD:

https://xml.e-galexis.com/V2/schemas/shoppingCart/commitShoppingCartResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/shoppingCart/

Das Antwort-Element <commitShoppingCartResponse> enthält das Element clientResponse mit den bestätigten Kundendaten und dem cartIdentifier Element mit der Warenkorb-ID. Zudem enthält der Response entweder ein successful oder ein unsuccessful Element, welches über den Erfolg des Warenkorb-Abschluss informiert.

Das Antwort-Element <commitShoppingCartResponse> enthält das Element clientResponse mit den bestätigten Kundendaten und dem cartIdentifier Element mit der Warenkorb-ID. Enthält der Response das <successful> Element, wurde der Warenkorb erfolgreich abgeschlossen respektive eine Bestellung erstellt.
Enthält der Response das <unsuccessfull> Element, konnte der Warenkorb aufgrund z.B. geänderter Konditionen nicht abgeschlossen werden. In diesem Fall wird der Warenkorb als Unterelement <shoppingCart> zurückgegeben. Dieser muss erneut gespeichert und abgeschlossen werden.

4.6 Produktverfügbarkeit prüfen

Der POS-Benutzer möchte schnell und zuverlässig die Verfügbarkeit, Preise und weitere Produktdetails eines Produkts bei Galexis abfragen. Die Abfrage erfolgt anhand einer eindeutigen Produktidentifikation sowie der zu prüfende Menge.

4.6.1 XML-Beispiel Request zu productAvailabilityRequest

XSD:

https://xml.e-galexis.com/V2/schemas/productAvailability/productAvailabilityRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/productAvailability/

Der POS-Benutzer sendet eine Anfrage mit der ProduktIdentifikation und der gewünschten Menge.

Das Hauptelement <productAvailabilityRequest> enthält nebst den technischen Attributen für die schemaLocation und den Namespace folgende Attribute:

- version: Version der Meldung. 2.0 ist zu verwenden
- language: Sprache in welcher z.B. die Produktbeschreibung zurückgeliefert wird. Wird keine Sprache übermittelt, wird die Sprache aus dem Kundenstamm verwendet.
- communicationSoftwareId: Identifikation der Software welche den Request übermittelt hat.

Das Unterelement client enthält die Authentifizierungsdaten (Webclient-Id und Passwort). Authentifizierungsdaten sowie die Liste der Produkte, die abgefragt werden sollen. Jedes Produkt wird durch seinen pharmacode eindeutig identifiziert und die gewünschte Menge requestedQuantity angegeben, um die Verfügbarkeit zu prüfen.

4.6.2XML-Beispiel Response zu productAvailabilityResponse

XSD:

https://xml.e-galexis.com/V2/schemas/productAvailability/productAvailabilityResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/productAvailability/

Galexis liefert für jedes abgefragte Produkt Informationen zur Verfügbarkeit, Preisen und weiteren Attributen.

Die Antwort enthält pro angefragte Zeile ein productAvailabilityResponseLine Element. Dieses beinhaltet:

productAvailabilityLine: Im Request angefragte Informationen

productResponse: Produktinformationen inklusive Preise.

availability: Information ob das Produkt in der gewünschten Menge verfügbar ist.

genericsResponse: Listet Produkte derselben Generika-Gruppe

productConditionLevels: Konditionsinformationen

4.7 Konditionen einsehen (KuKo's)

Der POS-Benutzer möchte jederzeit seine individuellen Galexis-Konditionen und Rabatte für Produkte einsehen, um die Preisgestaltung transparent und nachvollziehbar zu halten. Da die Anzahl an Zeilen für eine einzelne Abfrage zu gross ist, ist die Abfrage in Seiten unterteilt. Die erste Abfrage wird ohne requestKey (eine Art Positionsmerkmal) gemacht. Im Response erhält man dann den requestKey zurück welcher bei der nächsten Abfrage wieder mitgegeben wird. Dies wird solange gemacht, bis der Response meldet, dass man am Ende ist.

! Um KuKo Daten abfragen zu können, muss die Galexis AG den Kunden in die Aufbereitung aufnehmen.

4.7.1 XML-Beispiel Request zu customerSpecificConditionsRequest

XSD:

https://xml.e-galexis.com/V2/schemas/customerSpecificConditions/customerSpecificConditionsRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/customerSpecificConditions/

ⓘ Der POS-Benutzer sendet eine Anfrage zur Abfrage seiner spezifischen Konditionen.

Das Hauptelement <customerSpecificConditionsRequest> enthält nebst den technischen Attributen für die schemaLocation und den Namespace folgende Attribute:

- version: Version der Meldung. 2.0 ist zu verwenden
- language: Sprache in welcher z.B. die Produktbeschreibung zurückgeliefert wird. Wird keine Sprache übermittelt, wird die Sprache aus dem Kundenstamm verwendet.

— communicationSoftwareId: Identifikation der Software welche den Request übermittelt hat.

Unterhalb des Hauptelements gibt es folgende Unterelemente:

— client enthält die Authentifizierungsdaten (Webclient-Id und Passwort).

— browseRequest: enthält die direction (vorwärts oder rückwärts) sowie den requestKey. Bei der ersten Abfrage muss dieser leer übermittelt werden.

— customerSpecificConditionsBrowseDefinition: enthält die pageSize mit welcher die Anzahl an Konditionen, welche pro Seite zurückgeliefert werden, steuern kann.

4.7.2 XML-Beispiel Response zu customerSpecificConditions-Response

XSD:

https://xml.e-galexis.com/V2/schemas/customerSpecificConditions/customerSpecificConditionsResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/customerSpecificConditions/

Galexis liefert eine Übersicht der individuellen Konditionen und Rabatte des Kunden.

Die Antwort <customerSpecificConditionsResponse> bestätigt die Kundendaten und listet für jedes Produkt, das eine individuelle Kondition hat, den Pharmacode, den Rabatt in Prozent sowie den Zeitraum der Gültigkeit auf. So hat der POS-Benutzer jederzeit volle Transparenz über seine aktuellen Preisvorteile.

4.8 Rückkäufe registrieren

4.8.1 XML-Beispiel Request zu storeReturnCartRequest (für Rückkäufe)

XSD:

https://xml.e-galexis.com/V2/schemas/returnCart/storeReturnCartRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/returnCart/

Sendet Rückgabeartikel mit Mengen und Rückgabegrund an Galexis.

Der POS-Benutzer übermittelt die Produkte, die zurückgekauft werden sollen, inklusive Pharmacode, Rückgabemenge und Rückgabegrund. Damit kann Galexis die Rückkäufe systematisch erfassen.

4.8.2 XML-Beispiel Response zu storeReturnCartResponse (für Rückkäufe)

XSD:

https://xml.e-galexis.com/V2/schemas/returnCart/storeReturnCartResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/returnCart/

Liefert Annahme- oder Ablehnungsstatus der Rückgaben inkl. Konditionen.

Galexis bestätigt für jedes Produkt, ob die Position für einen Rückkauf akzeptiert, wurde `<lineAccepted>` oder nicht `<lineNotAccepted>` und liefert eine Rückmeldung `message`. Das `<repurchaseCartLineResponse>` Element enthält zudem Informationen zu welchen Konditionen das Produkt zurückgekauft wird. So ist die Rückkaufabwicklung für den POS-Benutzer transparent und nachvollziehbar.

4.8.3XML-Beispiel Request zu commitReturnCartRequest (für Rückkäufe)

XSD:

https://xml.e-galexis.com/V2/schemas/returnCart/commitReturnCartRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/returnCart/

Übermittelt die Bestätigung der akzeptierten Konditionen.

Der POS-Benutzer ist mit den im storeReturnCartResponse mitgeteilten Konditionen einverstanden und möchte die Positionen definitiv erfassen/abschliessen.

4.8.4XML-Beispiel Response zu commitReturnCartResponse (für Rückkäufe)

XSD:

https://xml.e-galexis.com/V2/schemas/returnCart/commitReturnCartResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/returnCart/

Bestätigt die endgültige Erfassung der Rückgaben.

Das Antwort-Element <commitReturnCartResponse> enthält das Element clientResponse mit den bestätigten Kundendaten und dem cartIdentifier Element mit der Warenkorb-ID. Enthält der Response das <successful> Element, wurden die erfassten Positionen erfolgreich erfasst/abgeschlossen.

4.9 Wareneingang automatisiert verarbeiten

Der POS-Benutzer möchte eingehende Lieferungen automatisiert verarbeiten, indem er Lieferdaten von Galexis abruft. Die Abfrage kann anhand von Lieferscheinnummer, Datum oder Boxnummer erfolgen. Ziel ist es, alle wichtigen Informationen wie Charge, Menge, Verfallsdatum, Preise, logistische Kosten und Rabatte systemseitig zu erhalten und im Lagerverwaltungssystem korrekt zu verbuchen.

⚠ Das Unterelement unterscheidet sich je nach eingegebenen Informationen: Lieferscheinnummer, Boxnummer oder Datum.

Untenstehend werden drei Varianten gezeigt, wie der Wareneingang technisch gehandhabt werden kann. Als Standardvariante ist der Wareneingang mit Box empfehlenswert. Welche der drei Varianten für Sie schlussendlich am meisten Sinn macht, können Sie selbst entscheiden. Bei Unsicherheiten hilft Ihnen ihr Galexis Berater.

Wareneingang mit Lieferschein: Im POS-System wird für jeden gelieferten Lieferschein ein Wareneingang durch den Benutzer erfasst. Der POS-Benutzer muss anschliessend die Behälter nach der übermittelten Bestellung sortieren. Dies erfordert die Durchführung mehrere Wareneingänge pro Lieferung.

Wareneingang mit Box: Im POS-System wird vom Benutzer pro Lieferung nur ein einziger Wareneingang erfasst. Durch das Scannen einer Boxnummer kann der Benutzer alle Daten der aktuellen Lieferung in das POS-System importieren. Dadurch lässt sich die gelieferte Ware in einem einzigen Schritt einbuchen – unabhängig von der Anzahl der Bestellungen, die in der Lieferung enthalten sind.

Die Lieferdaten werden von Galexis über die Tourennummer (Liefernummer) übermittelt, welche der aktuellen Lieferung aller Bestellungen – NOTA, BESO usw. – für den Endkunden entspricht. Da die Nummern aller zu einem bestimmten Zeitpunkt gelieferten Behälter bei Galexis registriert werden, um eine bessere Sendungsverfolgung zu gewährleisten, können diese Daten ebenfalls vom POS-Benutzer übernommen werden, um einen gruppierten Wareneingang durchzuführen.

Wareneingang mit Datum: Pro Datum kann nur ein einziger Wareneingang erfasst werden. Das bedeutet, dass der POS-Kunde, selbst wenn er mehrere Lieferungen pro Tag erhält, diese in einem einzigen Schritt einbuchen kann, sofern er abwartet, bis alle Lieferungen eingetroffen sind.

4.9.1 XML-Beispiel Request zu deliveryNoteDownloadRequest

XSD:

https://xml.e-galexis.com/V2/schemas/deliveryNote/deliveryNoteDownloadRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/deliveryNote/

ℹ Mit diesem Request fordert der POS-Benutzer Informationen zu einem Lieferschein oder einer Lieferung an.

Der POS-Benutzer gibt die spezifische Lieferscheinnummer, alternativ ein Lieferdatum oder eine Boxnummer an, um die zugehörigen Lieferdaten von Galexis abzurufen.
Das Hauptelement <deliveryNoteDownloadRequest> enthält nebst den technischen Attributen für die schemaLocation und den Namespace folgende Attribute:

- version: Version der Meldung. 2.0 ist zu verwenden
- language: Sprache in welcher z.B. die Produktbeschreibung zurückgeliefert wird. Wird keine Sprache übermittelt, wird die Sprache aus dem Kundenstamm verwendet.
- communicationSoftwareId: Identifikation der Software (Webclient-Id & Passwort).

Unterhalb des Hauptelements gibt es folgende Unterelemente:

- requestedDeliveryNotes enthält die zu übermittelnden Informationen, basierend auf den Daten, die für die Durchführung des Wareneingangs verwendet werden.
  - deliveryBox barcode: enthält die gescannte Behälternummer, um einen einmaligen Wareneingang für die aktuelle Lieferung durchzuführen.

    ⚠ Die Barcodenummer der Box enthält 7 Ziffern. Wenn die Box gescannt wird, müssen im PIS/POS-System jedoch 8 Ziffern erscheinen, da der Barcode eine achte Ziffer als Prüfsumme enthält (Die Prüfziffer wird als Response in Feld deliveryBoxNumber zurückgegeben). Es können auch 8 Ziffern inklusive Prüfziffer gesendet werden.

  - deliveryNote number : enthält die gescannte Lieferscheinnummer, um einen Wareneingang pro Lieferschein durchzuführen.

- deliveryDay date: enthält das Lieferdatum, um einen einzigen Wareneingang für das angegebene Datum durchzuführen

Auf dem Testsystem sind keine Lieferschein-Daten vorhanden. Aus diesem Grund kann das Attribute useStaticTestResponse auf dem Hauptelement deliveryNoteDownloadRequest auf "true" gesetzt werden. Ist das Attribute auf «true», können die Testlieferscheine mit den Nummern 100001, 100002, 200001, 200002, 300001, 400001, 500001 abgefragt werden.

4.9.2 XML-Beispiel Response zu deliveryNoteDownloadResponse

XSD:

https://xml.e-galexis.com/V2/schemas/deliveryNote/deliveryNoteDownloadResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/deliveryNote/

Galexis liefert detaillierte Informationen zur Lieferung zurück, inklusive aller wichtigen Positionsdaten.

Galexis stellt alle relevanten Lieferdetails zur Verfügung, wie zum Beispiel im Hauptelement <deliveryNoteDownloadResponse> der Antwort, welches *schemaLocation*, *Version* und *Sprache* enthält. Die Antwortelemente hängen von der ursprünglich übermittelten Anfrage ab. Für das Scannen eines Lieferscheins *deliveryNote number* :

- <clientResponse> Bestätigung der in der Anfrage übermittelten Kundendaten
- <deliveryNoteData> enthält das Leiferdatum und die Referenz der Lieferung
- <order>informationen zur bestellung
    - <ordreLine> Daten zur Bestellposition wie die Artikelbeschreibung und der Pharmacode des Artikels, die Preisinformationen sowie die Menge.
    - <deliveryBoxInformation>liefert die Informationen zur Boxnummer physicalBoxNumber und zur gelieferten Menge lineDeliveryQuantity.
        - <expiryAndBatchInformation>liefert die Informationen zum Verfalldatum (für Medikamente der Listen A bis E, gegebenenfalls auch für weiter Artikel) sowie chargennummer.
- <orderLine>enthält das Feld deliveryZone, sofern dieses übermittelt mit *commitShoppingCartReques* wurde. Liefergebiet, erlaubt leer, CUSTOMER, FRDGE, FRIGO, LSAB, NARCOTIC, OTC, PERFUMERY, ROBOT, STUPEFIANTS, URGENT, VERIFY.

Beim Scannen einer Box oder bei der Suche nach Datum werden zusätzlich die folgenden ergänzenden Informationen bereitgestellt:

- enthält die Informationen zum Lieferdatum (*deliveryDate*) sowie zur Tourennummer (*tourId*).
- *referenceNumber*, sofern dieses mit dem shoppingCart übermittelt wurde, wird die Bestellnummer des PIS/POS-System zurückgegeben.

⚠ Die 8 Ziffern des Barcodes (die 7 auf dem Behälter sichtbaren Ziffern plus die Prüfziffer) werden in der Antwort systematisch zurückgegeben, unabhängig von der Anzahl der in der Anfrage übermittelten Ziffern.

Dadurch kann der POS-Benutzer die Lieferung automatisch in sein System übernehmen und den Wareneingang effizient verwalten.

4.9.3 Praxisbeispiel Kunde mit SOAP Integration

Das folgende Beispiel zeigt den vollständigen Bestellprozess eines realen Kunden (Kunde 224946) über die POS v2 SOAP-Schnittstelle. Die Integration erfolgt über die Software «XY» und umfasst die Schritte von der Warenkorbübermittlung bis zum Lieferscheinabruf.
Der Kunde übermittelt seinen Warenkorb. Im Request wird eine abweichende Lieferadresse angegeben, die ausschliesslich für die Lieferdokumente gilt und den effektiven Lieferort in den Stammdaten nicht verändert. Die Kommunikation erfolgt in französischer Sprache mit der Software-ID des PIS/POS Herstellers. Der Request ist in *storeShoppingCartRequest.xml* dokumentiert, die Antwort in *storeShoppingCartResponse.xml* bestätigt die erfolgreiche Speicherung des Warenkorbs mit der ID 623837.

storeShoppingCartRequest.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
                   xmlns:galexis="http://www.galexis.com/XMLSchema">
  <SOAP-ENV:Body>
    <galexis:storeShoppingCartRequest version="2" language="fr"
        communicationSoftwareId="XY">
      <galexis:client number="007457" password="********"/>
      <galexis:customerSetupOverride>
        <galexis:deliveryAddress line1="PHARMACIE ***"
            line4="CHEMIN DE ***" line5PostalCode="1206" line5City="GENEVE"/>
      </galexis:customerSetupOverride>

      <galexis:cartHeader referenceNumber="116064"/>
      <galexis:cartLines>
        <galexis:cartLine cartLineNumber="1" orderQuantity="1">
          <galexis:product><galexis:pharmaCode id="2474044"/></galexis:product>
        </galexis:cartLine>
        <galexis:cartLine cartLineNumber="2" orderQuantity="1">
          <galexis:product><galexis:pharmaCode id="7771215"/></galexis:product>
        </galexis:cartLine>
        <!-- weitere cartLine-Einträge gekürzt -->
        <galexis:cartLine cartLineNumber="7" orderQuantity="1">
          <galexis:product><galexis:pharmaCode id="7798958"/></galexis:product>
        </galexis:cartLine>
      </galexis:cartLines>
    </galexis:storeShoppingCartRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

storeShoppingCartResponse.xml

```
<?xml version="1.0" encoding="utf-8"?>

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

  <soapenv:Body>

    <storeShoppingCartResponse xmlns="http://xml.e-galexis.com/V2/schemas/" language="fr"
version="2.0">

      <clientResponse customerNumber="224946" number="007457"/>

      <shoppingCart>

        <overriddenCustomerSettings>

          <deliveryAddress line1="PHARMACIE ***" line4="CHEMIN DE ***" line5City="GENEVE"
line5PostalCode="1206"/>

        </overriddenCustomerSettings>


        <cartIdentifier humanReadableCartNumber="623837" id="y2N+eQjPH9CqmwHs1fdkRQ=="/>

        <cartHeaderResponse deliveryDate="2025-10-14" referenceNumber="116064"/>


        <cartLinesResponse>

          <cartLineResponse lineAccepted="true" deliveryDate="2025-10-14" deliveryZone="CUSTOMER"
originalLineNumberList="1">

            <productResponse deliveryQuantity="1" invoiceQuantity="1" lineAmount="00.01"
wholesalerProductCode="2474044" description="BURGERSTEIN Coenzyme Q10 caps 30 mg 180">

              <EAN id="7640121570377"/>

            </productResponse>

            <availability status="yes"/>

          </cartLineResponse>


          <!-- weitere cartLineResponse-Einträge gekürzt -->


          <cartLineResponse backLogLine="true" lineAccepted="true" deliveryDate="2025-10-15"
deliveryZone="CUSTOMER" originalLineNumberList="7">

            <productResponse deliveryQuantity="1" invoiceQuantity="1" lineAmount="00.01"
wholesalerProductCode="7798958" description="PADMED CIRCOSAN N caps 200 pce">

              <EAN id="7680675410026"/>

            </productResponse>

            <availability status="yes"/>

          </cartLineResponse>

        </cartLinesResponse>


        <cartTotals totalAmountExclVATExclExpLogServiceCosts="00.01"
totalExpectedLogServiceCosts="0.01"/>

      </shoppingCart>

    </storeShoppingCartResponse>

  </soapenv:Body>

</soapenv:Envelope>
```

Im nächsten Schritt wird der Warenkorb über *commitShoppingCartRequest.xml* finalisiert, wodurch der Auftrag im Galexis-System erstellt wird.

commitShoppingCartRequest.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:galexis="http://www.galexis.com/XMLSchema">
    <SOAP-ENV:Body>
        <galexis:commitShoppingCartRequest phoneCallDesired="false" version="2" language="fr"
            communicationSoftwareId="XY" productDescriptionDesired="false"
            compressionDesired="false">
            <galexis:client number="007457" password="******"></galexis:client>
            <galexis:cartIdentifier id="y2N+eQjPH9CqmwHs1fdkRQ=="></galexis:cartIdentifier>
        </galexis:commitShoppingCartRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

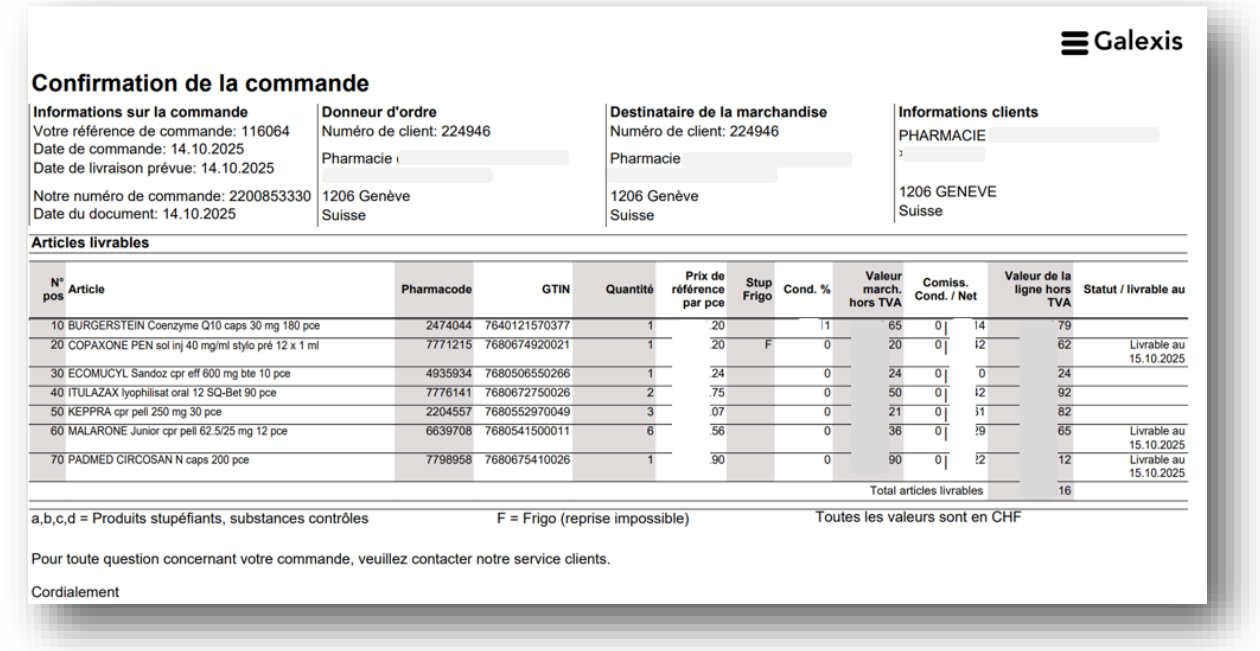
Die Antwort (*commitShoppingCartResponse.xml*) enthält die Bestätigung „G: Ordre ELE 2200853330 sauvegardé(e).“, womit der Auftrag erfolgreich angelegt ist.

commitShoppingCartResponse.xml

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <commitShoppingCartResponse xmlns="http://xml.e-galexis.com/V2/schemas/"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" language="de" version="2.0"
            xsi:schemaLocation="http://xml.e-
galexis.com/V2/schemas/shoppingCart/commitShoppingCartResponse.xsd">
            <clientResponse customerNumber="224946" number="007457" />
            <cartIdentifier humanReadableCartNumber="623837" id="y2N+eQjPH9CqmwHs1fdkRQ==" />
            <successful message="G: Ordre ELE 2200853330 sauvegardé(e)." />
        </commitShoppingCartResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

Die zugehörige Auftragsbestätigung zeigt alle bestellten Artikel, Mengen, Preise und Lieferdaten (14. und 15. Oktober 2025).

Auftragsbestätigung



Galexis

Confirmation de la commande

Informations sur la commande	Donneur d'ordre	Destinataire de la marchandise	Informations clients
Votre référence de commande: 116064 Date de commande: 14.10.2025 Date de livraison prévue: 14.10.2025	Numéro de client: 224946 Pharmacie	Numéro de client: 224946 Pharmacie	PHARMACIE
Notre numéro de commande: 2200853330 Date du document: 14.10.2025	1206 Genève Suisse	1206 Genève Suisse	1206 GENEVE Suisse

Articles livrables

N° pos	Article	Pharmacode	GTIN	Quantité	Prix de référence par pce	Stup Frigo	Cond. %	Valeur march. hors TVA	Comiss. Cond. / Net		Valeur de la ligne hors TVA	Statut / livrable au
10	BURGERSTEIN Coenzyme Q10 caps 30 mg 180 pce	2474044	7640121570377	1	.20		1	65	0	14	79	
20	COPAXONE PEN sol inj 40 mg/ml stylo pré 12 x 1 ml	7771215	7680674920021	1	.20	F	0	20	0	42	62	Livrable au 15.10.2025
30	ECOMUCYL Sandoz cpr eff 600 mg bte 10 pce	4935934	7680506550266	1	.24		0	24	0	0	24	
40	ITULAZAX lyophilisat oral 12 SQ-Bet 90 pce	7776141	7680672750026	2	.75		0	50	0	42	92	
50	KEPPRA cpr pell 250 mg 30 pce	2204557	7680552970049	3	.07		0	21	0	61	82	
60	MALARONE Junior cpr pell 62.5/25 mg 12 pce	6639708	7680541500011	6	.56		0	36	0	29	65	Livrable au 15.10.2025
70	PADMED CIRCOSAN N caps 200 pce	7798958	7680675410026	1	.90		0	90	0	72	12	Livrable au 15.10.2025
										Total articles livrables	16	

a,b,c,d = Produits stupéfiants, substances contrôles F = Frigo (reprise impossible) Toutes les valeurs sont en CHF

Pour toute question concernant votre commande, veuillez contacter notre service clients.

Cordialement

Nach der Auftragsverarbeitung kann der Kunde über *deliveryNoteDownloadRequest.xml* den elektronischen Lieferschein mit der Sendungsnummer 1001087188 abrufen.

deliveryNoteDownloadRequest.xml

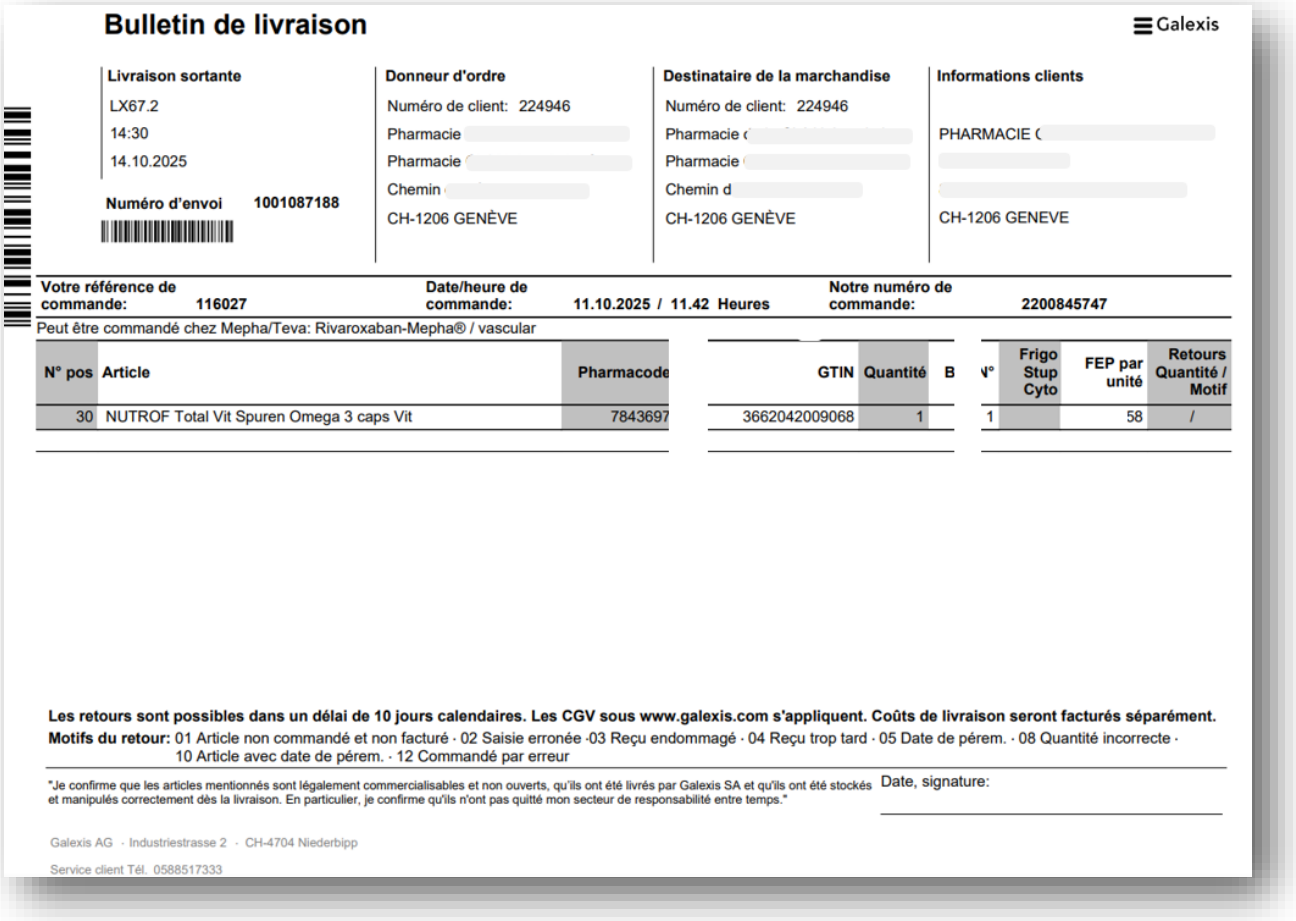
```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:galexis="http://www.galexis.com/XMLSchema">
    <SOAP-ENV:Body>
        <galexis:deliveryNoteDownloadRequest useStaticTestResponse="false" version="2" language="fr"
            communicationSoftwareId="ProPharma Systems AG" productDescriptionDesired="false"
            compressionDesired="false">
            <galexis:client number="007457" password="********"></galexis:client>
            <galexis:requestedDeliveryNotes>
                <galexis:deliveryNote number="1001087188"></galexis:deliveryNote>
            </galexis:requestedDeliveryNotes>
        </galexis:deliveryNoteDownloadRequest>
    </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Die Antwort (*deliveryNoteDownloadResponse.xml*) enthält die gelieferten Artikel, Chargen- und Ablaufdaten sowie die Boxnummern.

deliveryNoteDownloadResponse.xml

```
<?xml version="1.0" encoding="utf-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <deliveryNoteDownloadResponse xmlns="http://xml.e-galexis.com/V2/schemas/" language="fr"
version="2">
      <clientResponse customerNumber="224946" number="007457"/>

      <deliveryNoteData deliveryDate="2025-10-14" number="1001087188" tourId="LX67.2">

        <!-- Beispiel: vollständiger Order-Block -->
        <order number="2200845747" orderDate="2025-10-11" referenceNumber="116027">
          <orderLine pharmaCode="7843697"
                     description="NUTROF Total Vit Spuren Omega 3 caps Vit D3 90 pce"
                     invoicedQuantity="1" lineValue="00.01">
            <deliveryBoxInformation physicalBoxNumber="41086818" lineDeliveryQuantity="1">
              <expiryAndBatchInformation batchId="EXP20270531" expiryDate="2027-05-31"
partialQuantity="1"/>
            </deliveryBoxInformation>
          </orderLine>
          <!-- ggf. weitere orderLine-Zeilen dieses Orders ... -->
          <orderTotal totalBaseLinePriceNet="00.01" totalLineValue="00.01"/>
        </order>

        <!-- Hier kommen weitere <order> ... </order> Blöcke (z. B. Order 2, Order 3) -->

        <deliveryNoteTotal totalLines="11" totalLogisticServiceCost="00.01"
                           totalPieces="32" totalPrice="00.01"
                           totalPriceNet="00.01" totalVolumeL="13.03" totalWeightKg="3.68"/>
      </deliveryNoteData>
    </deliveryNoteDownloadResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Der entsprechende Lieferschein ist verfügbar.



Bulletin de livraison

Galexis

Livraison sortante	Donneur d'ordre	Destinataire de la marchandise	Informations clients
LX67.2	Numéro de client: 224946	Numéro de client: 224946	
14:30	Pharmacie	Pharmacie	PHARMACIE
14.10.2025	Pharmacie	Pharmacie	
Numéro d'envoi 1001087188	Chemin	Chemin d	
	CH-1206 GENÈVE	CH-1206 GENÈVE	CH-1206 GENEVE

Votre référence de commande: 116027 **Date/heure de commande:** 11.10.2025 / 11.42 Heures **Notre numéro de commande:** 2200845747

Peut être commandé chez Mepha/Teva: Rivaroxaban-Mepha® / vascular

N° pos	Article	Pharmacode	GTIN	Quantité	B	N°	Frigo Stup Cyto	FEP par unité	Retours Quantité / Motif
30	NUTROF Total Vit Spuren Omega 3 caps Vit	7843697	3662042009068	1		1		58	/

Les retours sont possibles dans un délai de 10 jours calendaires. Les CGV sous www.galexis.com s'appliquent. Coûts de livraison seront facturés séparément.

Motifs du retour: 01 Article non commandé et non facturé · 02 Saisie erronée ·03 Reçu endommagé · 04 Reçu trop tard · 05 Date de périem. · 08 Quantité incorrecte · 10 Article avec date de périem. · 12 Commandé par erreur

"Je confirme que les articles mentionnés sont légalement commercialisables et non ouverts, qu'ils ont été livrés par Galexis SA et qu'ils ont été stockés et manipulés correctement dès la livraison. En particulier, je confirme qu'ils n'ont pas quitté mon secteur de responsabilité entre temps."

Date, signature:

Galexis AG · Industriestrasse 2 · CH-4704 Niederbipp

Service client Tél. 0588517333

Ergänzend zeigt das Foto unten die physisch ausgelieferte Kiste mit der Boxnummer 41086818.

4.10 Offene Rückstände (Backlog) abrufen

Offene Rückstände sind bestellte Zeilen, für die Galexis noch eine Warenanlieferung erwartet um sie dem Kunden auszuliefern. Diese Rückstände können normale Lagerware betreffen (NOTA), oder auch die Ware die Galexis nicht am Lager hält, aber für den Kunden beim Lieferanten besorgen kann (Besorger).
Um offene Bestellrückstände abzurufen, nutzt der PIS/POS-Benutzer die Methode openBacklogRequest. Galexis gibt detaillierte Informationen zu ausstehenden Bestellrückstände zurück mittels openBacklogResponse.

4.10.1 XML-Beispiel Request zu openBacklogRequest

XSD:
https://xml.e-galexis.com/V2/schemas/openBacklog/openBacklogRequest.xsd
HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:
https://xml.e-galexis.com/V2/schemas/openBacklog/

Der POS-Benutzer sendet eine Anfrage, um alle offenen Rückstände abzurufen.

Der Request enthält den Webclient und das Passwort zur Authentifizierung, damit Galexis die offenen Rückstände für den entsprechenden Kunden korrekt identifizieren kann.

4.10.2 XML-Beispiel Response zu openBacklogResponse

XSD:
https://xml.e-galexis.com/V2/schemas/openBacklog/openBacklogResponse.xsd
HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:
https://xml.e-galexis.com/V2/schemas/openBacklog/

Galexis liefert eine Liste der offenen Rückstände inklusive Details zu jeder offenen Bestellposition.

Die Antwort enthält für jede offene Position den Pharmacode, die noch offene Menge, das Bestell-Eingangsdatum sowie die zugehörige Bestellnummer. Damit hat der POS-Benutzer jederzeit einen Überblick über alle nicht erfüllten Bestellungen.

4.11 Inventur durchführen

Der POS-Benutzer möchte eine Inventur starten, Zwischenergebnisse übermitteln, die Inventur abschliessen und den aktuellen Status der Inventur abfragen können. Das System unterstützt dabei auch die Nutzung eines Offline-Barcodescanners, der mit den notwendigen Produktdaten versorgt wird. In den untenstehenden Kapiteln werden die einzelnen Use Cases der Inventur mit Codebeispielen aufgezeigt.

Übersicht:

1. Inventur Starten

- XML-Beispiel Request: getInventoryIdRequest
- XML-Beispiel Response: getInventoryIdResponse

2. Inventurzwischenstand übermitteln

- XML-Beispiel Request: inventoryTransmissionRequest
- XML-Beispiel Response: inventoryTransmissionResponse

3. Inventur abschliessen

- XML-Beispiel Request: closeInventoryRequest
- XML-Beispiel Response: closeInventoryResponse

4. Inventurstatus abfragen

- XML-Beispiel Request: inventoryStatusRequest
- XML-Beispiel Response: inventoryStatusResponse

5. Offline Barcodescanner

- XML-Beispiel Request: getProductBaseMasterDataDiffRequest
- XML-Beispiel Response: getProductBaseMasterDataDiffResponse

4.11.1 Inventur Starten

4.11.1.1 XML-Beispiel Request: getInventoryIdRequest

XSD:

https://xml.e-galexis.com/V2/schemas/inventory/getInventoryIdRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/inventory/

Mit diesem Request fordert der POS-Benutzer eine neue Inventur-ID an, die für die laufende Inventur verwendet wird.

Der POS-Benutzer oder das POS-System initiiert eine neue Inventur, indem es eine Anfrage an Galexis sendet. Dabei übermittelt es die Kunden-Authentifizierungsdaten. Ziel ist es, eine neue Inventur-ID zu erhalten, die als eindeutige Kennung für den laufenden Inventurprozess dient.

4.11.1.2 XML-Beispiel Response: getInventoryIdResponse

XSD:

https://xml.e-galexis.com/V2/schemas/inventory/getInventoryIdResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/inventory/

Galexis liefert eine eindeutige Inventur-ID, die der POS-Benutzer für die weitere Kommunikation bei dieser Inventur nutzt.

Response:

Galexis bestätigt die Erstellung der Inventur und liefert eine eindeutige Inventur-ID zurück. Diese ID wird für alle weiteren Inventur-Operationen verwendet, um die Daten der spezifischen Inventur zuzuordnen.

4.11.2 Inventurzwischenstand übermitteln

4.11.2.1 XML-Beispiel Request: inventoryTransmissionRequest

XSD:

https://xml.e-galexis.com/V2/schemas/inventory/inventoryTransmissionRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/inventory/

Der POS-Benutzer übermittelt Zwischenergebnisse der gezählten Bestände je Produkt (identifiziert durch Pharmacode).

Während der Inventur sendet der POS-Benutzer (bzw. das System) Zwischenstände an Galexis. Die Anfrage enthält die Inventur-ID, eine Liste der gescannten Produkte inklusive Mengenangaben und weitere relevante Informationen. Dies ermöglicht eine laufende Dokumentation des aktuellen Bestands.

4.11.2.2 XML-Beispiel Response: inventoryTransmissionResponse

XSD:

https://xml.e-galexis.com/V2/schemas/inventory/inventoryTransmissionResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/inventory/

Galexis bestätigt den erfolgreichen Empfang der Zwischenergebnisse.

Galexis bestätigt den Empfang der übermittelten Daten und gibt eine Statusmeldung zurück, die signalisiert, ob die Daten korrekt verarbeitet wurden oder ob es Fehler gibt.

4.11.3 Inventur abschliessen

4.11.3.1 XML-Beispiel Request: closeInventoryRequest

XSD:

https://xml.e-galexis.com/V2/schemas/inventory/closeInventoryRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/inventory/

Der POS-Benutzer signalisiert mit diesem Request, dass die Inventur abgeschlossen ist.

Nach Abschluss der Erfassung sendet der POS-Benutzer eine Anfrage zur Beendigung der Inventur an Galexis. Die Anfrage enthält die Inventur-ID, um die Inventur eindeutig zu identifizieren und abzuschliessen.

4.11.4 Inventurstatus abfragen

4.11.4.1 XML-Beispiel Request: inventoryStatusRequest

XSD:

https://xml.e-galexis.com/V2/schemas/inventory/inventoryStatusRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/inventory/

Mit diesem Request fragt der POS-Benutzer den aktuellen Status der Inventur ab.

Der POS-Benutzer oder das System fragt den aktuellen Status einer laufenden Inventur ab, indem es die Inventur-ID mitsendet. Dies ermöglicht eine Überwachung des Fortschritts oder die Kontrolle des Abschlussstatus.

4.11.4.2 XML-Beispiel Response: inventoryStatusResponse

XSD:

https://xml.e-galexis.com/V2/schemas/inventory/inventoryStatusResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/inventory/

Galexis gibt den aktuellen Status der Inventur zurück (z.B. "inProgress", "completed").

Galexis liefert eine Statusmeldung zurück, die Informationen zum aktuellen Stand der Inventur enthält, z.B. ob sie läuft, abgeschlossen oder fehlerhaft ist.

4.11.5 Offline Barcodescanner

Der POS-Benutzer kann einen Offline-Barcodescanner nutzen, um bei der Inventur schnell und unabhängig vom Netz Barcodes von Produkten zu erfassen. Dies erhöht die Effizienz und ermöglicht die Inventur auch an Orten ohne Online-Verbindung.

Funktionsweise

- **Vorbereitung:** Das POS-System lädt per `getProductBaseMasterDataDiff` (Master-Daten-Differenz) die aktuellen Produktstammdaten herunter, die dann in den Offline-Barcodescanner übertragen werden.
- **Offline-Erfassung:** Der Scanner nutzt diese Daten lokal, um Barcodes zu erkennen und Produktinformationen anzuzeigen, ohne eine Live-Verbindung zu Galexis zu benötigen.
- **Synchronisation:** Nach der Inventurerfassung werden die gesammelten Daten vom Scanner zurück an das POS-System übermittelt und dann mittels der regulären Inventur-Methoden (`inventoryTransmissionRequest`) an Galexis gesendet.

4.11.5.1 XML-Beispiel Request: getProductBaseMasterDataDiffRequest

XSD:

https://xml.e-galexis.com/V2/schemas/getProductBaseMasterDataDiff/getProductBaseMasterDataDiffRequest.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/getProductBaseMasterDataDiff/

Der POS-Benutzer fordert die aktuellen Änderungen der Produktstammdaten seit dem letzten Synchronisationszeitpunkt an, um den Offline-Barcodescanner auf den neuesten Stand zu bringen.

Der POS-Benutzer (bzw. das POS-System) initiiert die Anfrage an Galexis, um die aktuellen Änderungen der Produktstammdaten seit dem letzten Synchronisationszeitpunkt abzurufen. Dabei übermittelt das System die Kundennummer, Authentifizierungsdaten sowie den Zeitstempel des letzten Updates. Dieser Request sorgt dafür, dass nur die geänderten Daten übertragen werden, was Bandbreite und Zeit spart.

4.11.5.2 XML-Beispiel Response: getProductBaseMasterDataDiffResponse

XSD:

https://xml.e-galexis.com/V2/schemas/getProductBaseMasterDataDiff/getProductBaseMasterDataDiffResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/getProductBaseMasterDataDiff/

Galexis liefert die Änderungen an den Produktstammdaten, die der Offline-Barcodescanner benötigt, um seine lokale Datenbank zu aktualisieren.

Galexis sendet als Antwort eine Liste aller Produktänderungen (neue, aktualisierte oder gelöschte Produkte) zurück. Diese enthalten alle wichtigen Informationen wie Pharmacode,

Beschreibung und Preise. Das POS-System nutzt diese Daten, um den Offline-Barcodescanner aktuell zu halten, sodass der Scanner auch ohne Online-Verbindung korrekt arbeitet.

4.12 Paket versenden

Der POS-Mitarbeiter möchte über Galexis Pakete an andere Galexis-Kunden versenden. Dazu übermittelt er die gewünschten Empfänger. Pro Empfänger kann zudem eine Referenz mitgegeben werden. Der Response beinhaltet dann den Wert des Barcodes mit welchem der POS-Mitarbeiter die Parcel-Etikette erstellen kann.

Beispiel einer Parceletikette:
https://xml.e-galexis.com/V2/schemas/parcelService/doc/ParcelEtikette.pdf

4.12.1 XML-Beispiel Request zu getAddressLabelsRequest

XSD:
https://xml.e-galexis.com/V2/schemas/parcelService/getAddressLabelsRequest.xsd
HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:
https://xml.e-galexis.com/V2/schemas/parcelService/

Der POS-Mitarbeiter sendet an Galexis eine Anfrage mit den gewünschten Empfänger sowie einer optionalen Referenz pro Empfänger.

Das Hauptelement <getAddressLabelsRequest> enthält nebst den technischen Attributen für die schemaLocation und den Namespace folgende Attribute:

- version: Version der Meldung. 2.0 ist zu verwenden
- language: Sprache von Texten welche zurückgeliefert werden. Wird keine Sprache übermittelt, wird die Sprache aus dem Kundenstamm verwendet.
- communicationSoftwareId: Identifikation der Software welche den Request übermittelt hat.

Im Unterelement addressLabels gibt es pro Empfänger ein addressLabel Element. Pro addressLabel gibt es die Unterelemente:

- recipient: Gewünschter Empfänger identifiziert mittels Galexis-Kundennummer (customerNumber) oder GLN (customerGLN).
- sendersReference: Referenz pro Absender

4.12.2 XML-Beispiel Response zu getAddressLabelsResponse

XSD:

https://xml.e-galexis.com/V2/schemas/parcelService/getAddressLabelsResponse.xsd

HTML-Dokumentation mit Codebeispielen und weiterführenden Unterlagen:

https://xml.e-galexis.com/V2/schemas/parcelService/

Galexis bestätigt die erfolgreiche Verarbeitung der Anfrage. Für jeden gültigen Empfänger wird ein Barcode-Wert zurückgeliefert, mit welchem der POS-Mitarbeiter die ParcelService Adresse erstellen kann. So ist ein reibungsloser Versand gewährleistet.

Pro Empfänger wir ein addressLabelResponse Element zurückgeliefert. Dieses beinhaltet die folgenden Elemente:

- addressLabel angefragte Informationen für Empfänger und Referenz
- senderAddress Adresse des Absender
- recipient Galexis Kundennummer des Empfänger
- recipientAddress Adresse des Empfängers
- parcelServiceBarcode Wert welcher als Barcode auf der Parceletikette aufgedruckt wird.

5 Tests & Produktivsetzung

Testumgebung:

- URL: https://test.e-galexis.com/testPOS
- Zugang via Test-Login (nach Kontaktaufnahme)

Produktivstellung:

- Nach erfolgreichen Tests wird Zugriff zum Live-System erteilt
- Kontakt: xml@galexis.com
- Beschaffung simulieren und testen mit der Swagger UI