Leitfaden zur Implementierung eines SOAP-Client für die Webservice API von Speed4Trade CONNECT

Version 1.94

Inhaltsverzeichnis

Inhaltsverzeichnis ... 2
1 Einleitung ... 6
2 Speed4Trade CONNECT ... 7
2.1 Speed4Trade CONNECT als Datendrehscheibe für den Online-Handel ... 7
2.2 Speed4Trade CONNECT als Middleware ... 8
2.2.1 Systemaufbau eines „Komplettsystems“ (Warenwirtschaft + Speed4Trade CONNECT) ... 8
3 Das Software-Entwicklungsprojekt SOAP-Client ... 10
3.1 Kommunikations-Protokoll ... 10
3.1.1 Integration Request-XML in SOAP-Envelope ... 12
3.2 Sicherheit ... 14
3.2.1 Verschlüsselte Kommunikation ... 14
3.2.2 Server-Authentifizierung ... 14
3.2.3 Client-Authentifizierung ... 14
3.3 Entwicklungsframework ... 15
3.4 Programm-Logik ... 15
3.5 Bedarfsanalyse ... 16
4 Funktionsübersicht ... 18
4.1 Grundeinstellungen, Stammdaten und Angebotslisting ... 18
4.2 Angebots- und Auftragsworkflow ... 19
5 Wichtige Grundeinstellungen - vorbereitende API Calls ... 20
5.1 Mandanten ... 22
5.1.1 WebService-Call: fetchMandators (Mandanten) ... 22
5.1.1.1 Sprachen ... 22
5.1.1.2 Währungen ... 23
5.2 Verkaufskanäle ... 23
5.3 SubChannels ... 23
5.3.1 Webservice-Call: fetchSubChannels (Sub-Channels) ... 23
5.4 Accounts ... 23
5.4.1 WebService-Call: fetchAccounts (Accounts) ... 24
5.5 Textbausteintypen ... 25
5.5.1 WebService-Call: fetchTextModuleTypes (Textbausteintypen) ... 26
5.6 Bildbausteintypen ... 27
5.6.1 Webservice-Call: fetchImageModuleTypes (Bildbausteintypen) ... 28
5.7 Medienverwaltung ... 28
5.8 Anwendung von Bild-, Text- und Medienbausteinen in Angeboten ... 29
5.9 Zentralisierte Attributverwaltung (ZAV) ... 30
5.9.1 WebService-Call: fetch / updateAttributes (zentrale Attribute und Werte) ... 31
5.9.2 Webservice-Call: fetch / updateAttributeSets (zentrale Attributgruppierungen) ... 31
5.10 Preislisten (updatePricelist) ... 32
5.11 Versandkostenkategorien ... 34
5.11.1 Webservice-Call: fetchDeliveryChargeCategories / updateDeliveryChargeCategories (Versandkostenkategorien) ... 34
5.11.1.1 Versandkosten ... 34
5.12 Lieferzeiten ... 35
5.12.1 Webservice-Call: fetchDeliveryTimes / updateDeliveryTimes (Lieferzeiten) ... 35

5.13 Hersteller ... 36
5.13.1 Webservice-Call: fetchManufacturers / updateManufacturers (Hersteller) ... 36
5.14 Lager ... 37
5.14.1 Webservice-Call: fetchWarehouses (Lager) ... 37
6 Warengruppen ... 37
6.1 Webservice-Call: updateCategories / fetchCategories (Warengruppen) ... 39
7 Artikelstammdaten ... 39
7.1 Artikel Grunddaten ... 39
7.1.1 Grunddaten - Allgemein ... 40
7.1.2 Grunddaten - Textverwaltung ... 41
7.1.3 Grunddaten - Bilderverwaltung ... 42
7.2 Webservice-Call: updateItems (Artikelstammdaten) ... 43
7.2.1 Artikel -> Grunddaten -> Allgemein ... 43
7.2.2 Artikel -> Grunddaten -> Zusatzartikel ... 45
7.2.3 Artikel -> Grunddaten -> Artikelset ... 45
7.2.4 Artikel-Varianten ... 45
7.2.5 Artikel-Attribute (Zentralisierte Attributverwaltung - ZAV) ... 46
7.3 Webservice-Call: fetchItems ... 46
8 Artikel - Verkaufskanal-spezifische Daten ... 47
8.1 Artikelbezeichnung und Preis ... 47
8.2 Marktplatz Amazon ... 48
8.2.1 Webservice-Call: updateItems (Amazon-spezifische Daten) ... 48
8.2.1.1 Artikel -> Amazon -> Allgemein ... 49
8.2.1.2 Artikel -> Amazon -> Bilder ... 49
8.2.1.3 Artikel -> Amazon -> Produktdaten ... 50
8.2.1.4 Artikel -> Amazon -> Kategoriedaten ... 51
8.3 Marktplatz eBay ... 52
8.3.1 Webservice-Call: updateItems (eBay-spezifische Daten) ... 53
9 Übersicht über marktplatzspezifische Eigenheiten ... 55
10 Bestandsverwaltung ... 56
10.1 Bestandsübergabe per Artikelstammdaten ... 57
10.1.1 Webservice-Call: updateItems (Artikelbestand) ... 58
10.1.2 Artikel -> Grunddaten -> Bestandsverwaltung ... 58
10.2 Bestandsübergabe per separater Bestandsaktualisierung ... 59
10.2.1 Webservice-Calls: updateStockCount (Artikelbestände) ... 59
11 Auftragsverwaltung ... 60
11.1 Auftragsworkflow ohne externes System ... 60
11.2 Auftragsworkflow mit externem System (ERP/WWS) ... 61
11.2.1 Standard-Auftragsworkflow ... 62
11.2.1.1 Dubletten-Prüfung ... 63
11.2.1.2 Aufträge mit Kunden-Kommentaren ... 64
11.2.2 Auftragsworkflow bei eBay-Aufträgen ohne abgeschlossenen eBay-Checkout ... 64
11.2.3 Auftragsworkflow bei Amazon-Aufträge - auf E-Mail warten ... 65
11.3 Webservice-Call: fetchOrders ... 66
11.3.1 Relevante Felder im Auftrag ... 66
11.3.1.1 Aufträge -> Grunddaten ... 66
11.3.1.2 Aufträge ->Pakete ... 67

11.3.1.3 Auftragsstatus: umgruppierte Aufträge bei eBay ... 68
11.4 Webservice-Call: updateOrders ... 68
12 Weitere Funktionen ... 70
12.1 E-Mail Benachrichtigung an Käufer ... 70
12.2 Zahlungsabgleich ... 70
12.2.1 Webservice-Call: fetchPayments ... 70
12.3 Versand der Aufträge über FBA ... 70
12.4 Teillieferungen ... 71
12.5 Aftersales ... 71
12.5.1 eBay ... 71
12.5.2 Amazon ... 71
13 Fernsteuerung von Speed4Trade CONNECT ... 72
13.1 Allgemein ... 72
13.2 Workflow zum automatischen Einstellen ... 72
13.3 Abgleich von neuen Artikeldaten zu Marktplätzen ... 73
13.4 Masken von Speed4Trade CONNECT in einem Container anzeigen ... 73
13.4.1 Aufbau der Masken-Adresse für den automatischen Login ... 73
13.4.2 Informationen zur Authentifizierung ... 73
13.5 Ausblenden des Rahmens des Speed4Trade-CONNECT-WEB-Templates ... 74
13.6 Definition einer Callback-Adresse ... 75
13.6.1 Parameter callback nicht gesetzt ... 75
13.6.2 Parameter callback = no ... 75
13.6.3 Parameter callback auf eine andere Maske setzen ... 75
14 Zusammenfassung ... 76
14.1 Übersichtsdiagramm der Workflows ... 77

Abbildungsverzeichnis

Abbildung 1: Systemskizze Speed4Trade CONNECT für den Online-Handel 7
Abbildung 2: Systemaufbau eines "Komplettsystems" 9
Abbildung 3: HTTPS Kommunikation der XML-Daten mit Hilfe des SOAP-Protokolls..... 11
Abbildung 4: Funktionsübersicht - Grundeinstellungen, Stammdaten und Listing 18
Abbildung 5: Funktionsübersicht - Angebots- und Auftragsworkflow 19
Abbildung 6: Pflichtfelder sind mit einem "gelben" Balken hinterlegt 20
Abbildung 7: Weitere Optionen, bspw. Sprachauswahl im Speed4Trade Online Handbuch 21
Abbildung 8: Mandanten-Verwaltung..... 22
Abbildung 9: Accountverwaltung 24
Abbildung 10: Textbausteintypen 25
Abbildung 11: Bildbausteintypen 27
Abbildung 12: Anwendung von Bild-, Text- und Medienbausteinen in Angeboten 29
Abbildung 13: Anwendung von zentralen Attributen 30
Abbildung 14: Beispiel für Integration von zentralen Attributen..... 31
Abbildung 15: Basis- und abgeleitete Preislisten (BPL & APL) 32
Abbildung 16: Anwendungsbeispiel von Basispreislisten 33
Abbildung 17: Versandkostenkategorien 34
Abbildung 18: Lieferzeiten 35
Abbildung 19: Hersteller 36
Abbildung 20: Lagerverwaltung 37
Abbildung 21: Warengruppen 38
Abbildung 22: Artikel Grunddaten 40
Abbildung 23: Artikel Textverwaltung..... 41
Abbildung 24: Artikel Bildverwaltung..... 42
Abbildung 25: Artikeldaten - Amazon 48
Abbildung 26: eBay Kategorien 52
Abbildung 27: Bestandsverwaltung 56
Abbildung 28: Navigationsleiste..... 60
Abbildung 29: Angebotsworkflow..... 60
Abbildung 30: Auftragsworkflow 60
Abbildung 31: Auftragsdaten 61
Abbildung 32: Auftragsworkflow 63
Abbildung 33: Workflow zum automatischen Einstellen 72
Abbildung 34: Übersichtsdiagramm der Workflows 77

1 Einleitung

Der vorliegende Leitfaden zur Implementierung eines SOAP Clients für die Speed4Trade-CONNECT-Webservice-API gibt Ihnen einen umfassenden Überblick zur Realisierung einer eigenen Anbindung an den Webservice der Speed4Trade GmbH.

Ziel dieses Dokumentes ist es:

- Ihnen einen generellen Überblick über die Funktionsweise von Speed4Trade CONNECT zu geben.
- Sie zu sensibilisieren, dass die Implementierung eines SOAP Clients ein vollwertiges Software-Entwicklungsprojekt ist.
- Ihnen die grundsätzlichen Workflows der Speed4Trade CONNECT aufzuzeigen.
- Die zur Abbildung der Workflows notwendigen Webservice-Calls zu definieren.
- Die Anbindung Schritt für Schritt in der Integrationstiefe zu verfeinern.

Der Leitfaden kann wie folgt gelesen werden. Zunächst erhalten Sie pro Kapitel:

- eine generelle Erklärung der Funktion anhand des Speed4Trade CONNECT Clients,
- dann werden die Webservice-Calls für eine minimale Standard-Anbindung erklärt
- und schließlich Hinweise auf eine erweiterte Anbindung mit zusätzlichen Optionen gegeben.

Sie sollten zunächst mit der Implementierung einer Standard-Anbindung beginnen. Dies hat den Vorteil, dass Sie dazu noch keine tiefergehenden Kenntnisse zur Funktionsweise der angebundenen Marktplätze (eBay, Amazon, etc.) oder Online-Shops benötigen. Die Standard-Anbindung beschränkt sich auf das vorbereitende Auslesen von Grund- und Stammdaten, das Übergeben von Artikel-Grunddaten, Bestandsänderungen sowie das Abholen von Aufträgen und Setzen neuer Auftragsstatus. Durch diese Vorgehensweis können Sie schnelle Projekterfolge erzielen und bekommen Zug um Zug mehr Sicherheit bei der Anwendung der Webservice-Calls. In einem zweiten Schritt können dann komplexere Calls und Spezialfunktionen realisiert werden.

Lesen Sie jetzt, was die Speed4Trade CONNECT ist, welche Funktionen und Leistungsmerkmale die Software besitzt und wie beispielhaft ein "Komplettsystem" mit ERP/Warenwirtschaft, Speed4Trade CONNECT als Middleware und den Online-Verkaufskanälen aufgebaut ist.

2 Speed4Trade CONNECT

Speed4Trade CONNECT ist eine Client-Server-basierte Anwendungssoftware für Unternehmen, die Internet-Handel betreiben. Sie ermöglicht es, Artikel und Waren aller Art hoch-automatisiert über eine Vielzahl von Online-Verkaufskanälen wie z.B. dem eigenen Online-Shop (Multi-Shop-fähig), eBay, Amazon, Yatego, Rakuten, Shopgate etc. zu verkaufen. Die Software vereint die Verkaufsabwicklung über die verschiedenen Online-Kanäle zentral in einem System und erlaubt somit die Erschließung zusätzlicher Absatzkanäle sowie effizienten Multi-Channel Vertrieb im Internet. Dabei steuert die Software alle Bereiche des professionellen Online-Handels: Artikelverwaltung, Aufbereitung für die Produktpräsentation auf den Online-Plattformen, Listing in den verschiedenen Benutzer-Accounts, Käufer-Kommunikation, Auftragsbearbeitung bis hin zur Unterstützung der Versandabwicklung und der After-Sales-Prozesse.

2.1 Speed4Trade CONNECT als Datendrehscheibe für den Online-Handel

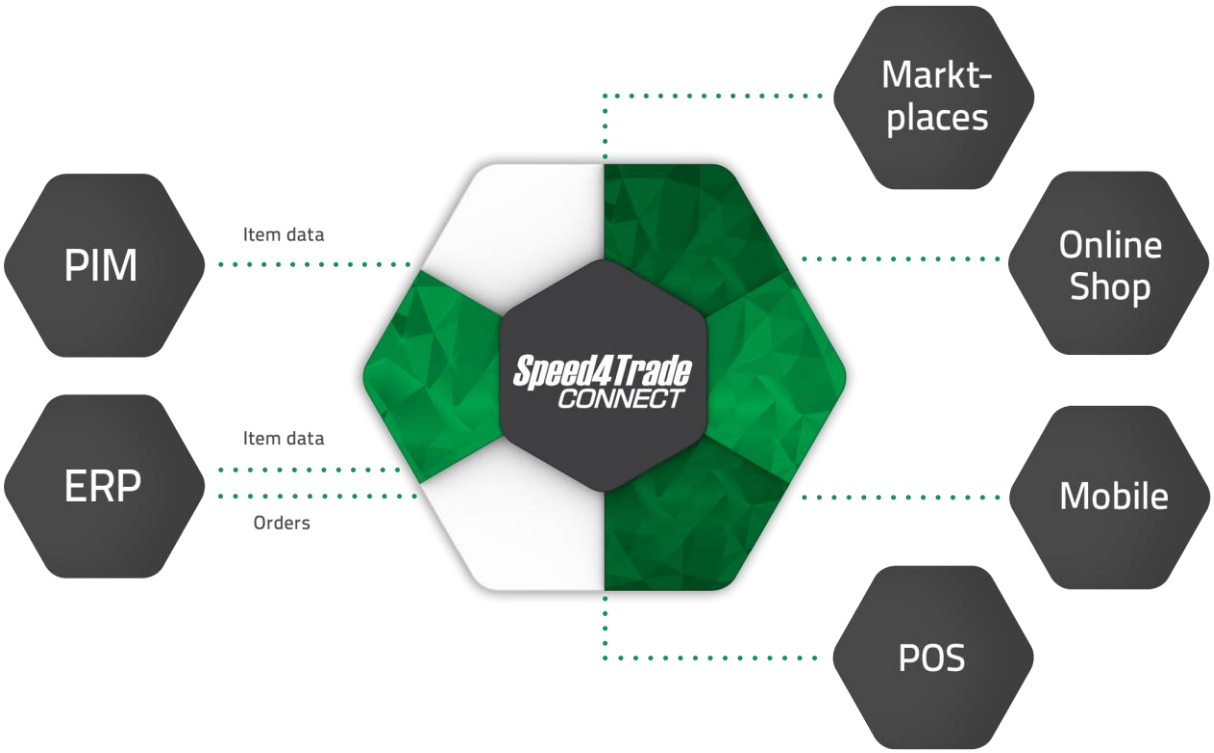

Abbildung 1: Systemskizze Speed4Trade CONNECT für den Online-Handel

Zu den Leistungsmerkmalen der Speed4Trade CONNECT zählen u.a. die ergonomische, benutzerfreundliche Oberfläche, eine echte Client-/Server-Technologie und performante Webservice Anbindungen. Die Benutzeroberfläche orientiert sich dabei an verbreitete Standards etablierter Business-Anwendungen und ist somit sehr intuitiv zu bedienen. Die Software ist skalierbar und bildet wichtige Komponenten wie Benutzer, Mandanten, Länder, Sprachen, Währungen in „Vielzahl“ ab,

wodurch das System bereits von Beginn an für Wachstum und den Ausbau des Online-Geschäfts ausgelegt ist.
Eine besondere Funktion ist der interaktive Bestandsabgleich (IABA). Dieser gleicht die Artikelbestände in den Online-Angeboten automatisch und Verkaufskanal-übergreifend entsprechend der Warenverfügbarkeit ab. Füllt somit Angebotsbestände wieder auf und steuert daher den reibungslosen Mehr-Kanal-Abverkauf. Zahlreiche Automatismen wie das zeitgesteuerte Einstellen, die Auftragsabholung, Käuferbenachrichtigungen oder auch Zahlungsabgleiche ersetzen den manuellen Eingriff in die Geschäftsprozesse, wodurch die Software immense Kosteneinsparungspotentiale bei gleichzeitiger Steigerung der Prozessqualität bietet.

Das System kann als Inhouse-Lösung in einem Unternehmensnetzwerk oder auch als gehostete Lösung von unterschiedlichen Standorten aus betrieben werden. Zudem ist die Lösung Kunden-individuell anpassbar und kann somit auf die verschiedensten Anforderungen und Geschäftsprozesse zugeschnitten werden (Customizing).

2.2 Speed4Trade CONNECT als Middleware

Die Speed4Trade CONNECT-Lösung kann – neben dem Einsatz als Stand-Alone-Version – als sogenannte Middleware zwischen einem bestehenden ERP- / Warenwirtschaftssystem und den verschiedenen „Online-Welten“ eingesetzt werden. Die Anbindung externer Systeme (z.B. ERP / Warenwirtschaft) erfolgt über die Speed4Trade-CONNECT-API und Webservice-Schnittstelle. Somit können Daten wie Artikel, Bestände und Aufträge automatisiert zwischen den Systemen ausgetauscht werden, was die Basis für durchgängige Verkaufsprozesse und niedrige Prozesskosten bildet.

2.2.1 Systemaufbau eines „Komplettsystems“ (Warenwirtschaft + Speed4Trade CONNECT)

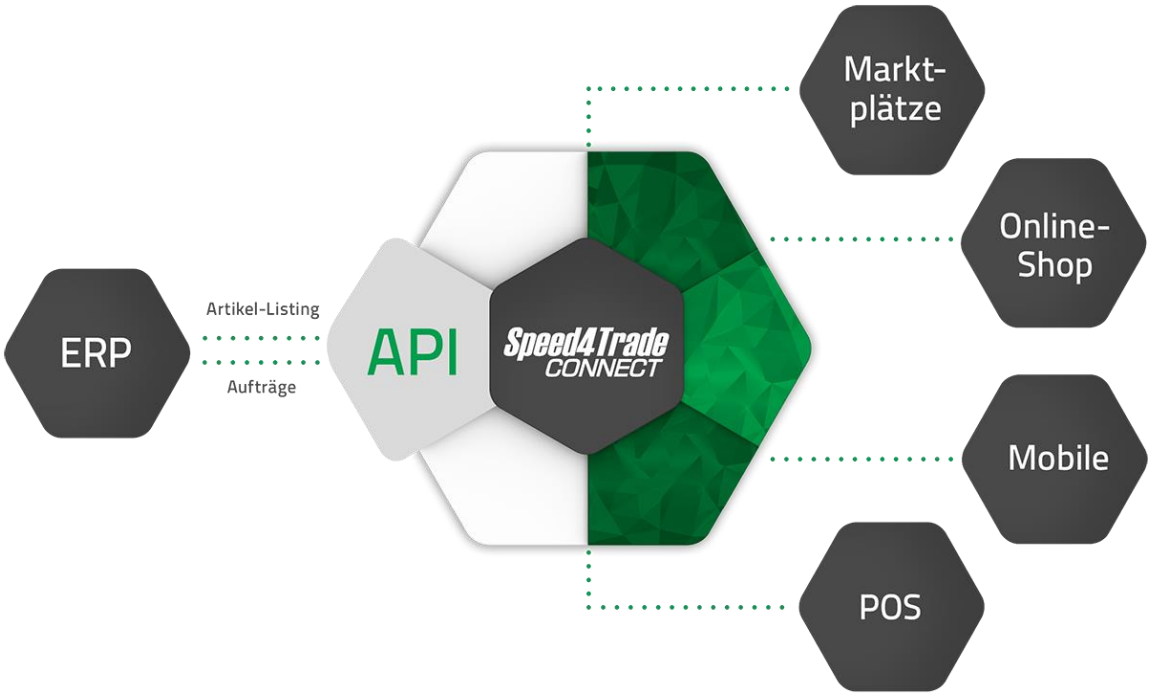

Abbildung 2: Systemaufbau eines "Komplettsystems"

Bei der Implementierung sind grob folgende Prozesse zu realisieren:

- Aus dem externen System werden Grunddaten wie Bilder, Beschreibung, Preislisten eines Artikels usw. an Speed4Trade CONNECT übergeben.
- In Speed4Trade CONNECT erfolgt im einfachen Integrationsmodell die weitere Verkaufskanal-spezifische Aufbereitung der Daten, bspw. Zuweisung von Kategorien und Attributen.
- Das externe System liefert regelmäßig aktuelle Bestandswerte und ggf. Preisänderungen zu den Artikeln.
- Nach der finalen Aufbereitung der Artikel in Speed4Trade CONNECT werden diese in den Verkaufskanälen als „laufende Angebote“ eingestellt.
- Neue Aufträge werden durch Speed4Trade CONNECT aus den Verkaufskanälen abgeholt.
- Das externe System holt diese Aufträge in einem definierten Status ab und aktualisiert wiederum den Auftragsstatus in Speed4Trade CONNECT nach bestimmten Ereignissen (Zahleingang, Versand der Ware).
- Speed4Trade CONNECT synchronisiert den jeweiligen Auftragsstatus in den Verkaufskanälen und informiert den Kunden über den aktuellen Stand.

Die Aufzählung macht bereits deutlich, dass es sich bei der Entwicklung eines SOAP-Clients um mehr handelt, als nur um ein paar Scripte für den Datenimport und -export. Ein SOAP-Client ist ein vollwertiges Programm mit eigener Programm-Logik und muss daher auch mit professionellen Entwicklungswerkzeugen und -methoden realisiert werden.

3 Das Software-Entwicklungsprojekt SOAP-Client

Um den Speed4Trade CONNECT Webservice nutzen zu können, sollten Entwickler ein gutes Verständnis von und Erfahrungen mit SOAP (Simple Object Access Protocol, siehe dazu http://de.wikipedia.org/wiki/SOAP) mitbringen. Es ist lediglich ein Internetzugang nötig, über den der Webservice via TCP/IP angesprochen werden kann. Neben dem Verbindungs-Management mit dem Speed4Trade-CONNECT-Webservice sollten SOAP- und XML-Tools vorhanden sein, um Aufruf-Dokumente aus der Anwendungslogik heraus zu erstellen und die Informationen aus dem Antwort-Dokument auszulesen (‚parsen'). Zudem werden die URLs der Webservice Aufrufpunkte (Endpunkte), sowie 'Username' und 'Password' und eine Freischaltung benötigt.

Eine WSDL Datei (Web Services Description Language) mit den relevanten Webservice-Definitionen wird über die URL https://IHR-SPEED4TRADE-CONNECT-SERVER/services/ecs-ws?wsdl zur Verfügung gestellt. Diese wird benötigt, um Applikationen für diesen Webservice zu entwickeln. Sie spezifiziert die erforderliche Struktur der SOAP Botschaften, die Funktionssignaturen, die ‚API Service Bindings' sowie die ‚Service Endpoints'.

Mit dieser Schnittstellenbeschreibungsdatei können Entwickler einen SOAP-Client entwickeln, um Business-Object Interfaces sowie Network Stubs daraus zu generieren und sich die Arbeit deutlich zu erleichtern. Der Request wird schließlich als SOAP Methoden-Aufruf formuliert und an den SOAP Service-Endpoint übermittelt.

Es wird die Benutzung des aktuellsten Bindings (ecs-wsSoap12Binding) und Endpoints (ecs-wsHttpsSoap12Endpoint) für SOAP 1.2 empfohlen. Die Unterstützung für SOAP 1.1 ist obsolet und nur noch aus Gründen der Abwärtskompatibilität vorhanden.

3.1 Kommunikations-Protokoll

Zur Kommunikation mit dem Webservice wird eine Kombination aus Anfrage- und Antwortdaten im XML-Format und dem Standard-Protokoll SOAP genutzt. Dieses bietet von sich aus keine Möglichkeit zur Versionierung von Anfragen und Antworten, kann jedoch auf vielen Systemen sehr einfach angesprochen werden.

Aus diesem Grund wird XML als flexibles, versionierbares Datenformat gewählt, und SOAP als einfach anzusprechendes Container-Protokoll zur Übertragung von Anfragen und Antworten des Web-Service genutzt.

Bitte stellen Sie sicher, dass eine aktuelle und sichere SSL-Version verwendet wird. Sobald ein Verfahren unsicher wird, wird dieses nicht mehr unterstützt. Eine separate Ankündigung der nicht mehr unterstützten Formate findet nicht statt (Cipher Verfahren).

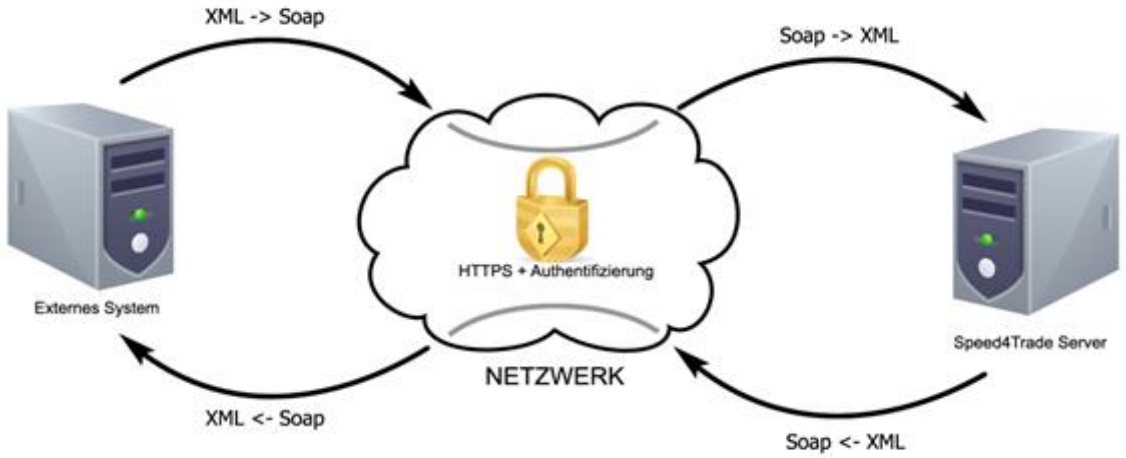

Abbildung 3: HTTPS Kommunikation der XML-Daten mit Hilfe des SOAP-Protokolls

Ein beispielhafter Aufbau der XML-Daten für die Kommunikation ist wie folgt spezifiziert:

Beispiel Request: Auslesen der im Speed4Trade CONNECT System hinterlegten Accounts

```
<request method="fetchAccounts" version="1.0.0">
   <filter>
      <mandator_id filter_method="=" filter_value="1" />
      <last_changed from_date="2000-01-01T00:00:00.000+01:00"
     to_date="2012-12-31T00:00:00.000+01:00" />
      <created_at from_date="2000-01-01T00:00:00.000+01:00"
     to_date="2012-12-31T00:00:00.000+01:00" />
   </filter>
</request>
```

Beispiel Response: Hinterlegte Accounts

```
<response method="fetchAccounts" version="1.0.0">
   <report return_code="0"/>
   <account>
      <account_id>25217</account_id>
      <mandator_id>1</mandator_id>
      <account_identifier>eBay-Account</account_identifier>
      <channel>
         <channel_id>2000</channel_id>
         <channel_identifier>eBay</channel_identifier>
      </channel>
      <sub_channel>
         <sub_channel_id>2100</sub_channel_id>
         <sub_channel_identifier>eBay.de</sub_channel_identifier>
      </sub_channel>
      <email_account>
         <email_account_id>24594</email_account_id>
```

```
<email_account_identifier>Speed4Trade</email_account_identifier>
         <email_address>contact@speed4trade.com</email_address>
      </email_account>
      <active>true</active>
      <order_number_prefix>Bay</order_number_prefix>
   </account>
</response>
```

Als Datentypen werden ausschließlich standardisierte, durch die XML-Schema-Datentypdefinition festgelegte Typen verwendet (siehe dazu auch die API Dokumentation unter: https://partner.speed4trade.com/entwickler/api-webservice/ -> Einleitung -> Erläuterung der Feldlisten).

Hinweis:
Falls der Webservice erhaltene Eingabedaten nicht interpretieren kann, wird durch den Server eine Exception ausgelöst, die vom Client abgefangen und **behandelt werden muss**. Dabei sind zwei Fälle zu unterscheiden. Zum einen wenn der SOAP Request strukturell falsch ist und nicht verarbeitet werden kann und zum anderen wenn der Request zwar strukturell richtig, inhaltlich aber fehlerhaft ist. Das heißt, wenn zum Beispiel ein Auftrag mit falschem Account angefragt wird.

3.1.1 Integration Request-XML in SOAP-Envelope

Wie bereits in Abbildung 3 erwähnt, werden die Requestdaten in XML-Format innerhalb des SOAP-Envelope verschickt. Dazu werden die XML-Daten entweder in einen CData-Block eingefügt oder als reine XML-Daten, allerdings muss in diesem Fall eine Maskierung der Daten erfolgen. Die Maskierung kann mittels einer in vielen Programmiersprachen vorhandenen EncodeXml-Funktion passieren. Nachfolgend jeweils ein Beispiel für SOAP-Envelope, Request-Daten im CData-Block und maskierte Daten.

Beispiel SOAP-Envelope: vereinfachtes Konstrukt

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
   xmlns:web="http://webservice.web.speed4trade.com">
   <soap:Header/>
   <soap:Body>
      <web:fetchMandators>
         <web:requestData>
            <!-- Request-XML -->
         </web:requestData>
      </web:fetchMandators>
   </soap:Body>
</soap:Envelope>
```

In diesen Envelope können nun Daten als CData-Block eingebunden werden. Dazu hier ein kleines Beispiel anhand des Calls fetchMandators.

Beispiel CData-Block: Auslesen der Mandanten (I)

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
   xmlns:web="http://webservice.web.speed4trade.com">
   <soap:Header/>
   <soap:Body>
      <web:fetchMandators>
         <web:requestData>
            <![CDATA[<?xml version="1.0" encoding="UTF-8"?>
               <request method="fetchMandators" version="1.0.0" />
            ]]>
         </web:requestData>
      </web:fetchMandators>
   </soap:Body>
</soap:Envelope>
```

Beispiel maskierter Request: Auslesen der Mandanten (II)

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
   xmlns:web="http://webservice.web.speed4trade.com">
   <soap:Header/>
   <soap:Body>
      <web:fetchMandators>
         <web:requestData>
            &lt;?xml version=&quot;1.0&quot; encoding=&quot;UTF-
8&quot;?&gt;
            &lt;request method=&quot;fetchMandators&quot;
version=&quot;1.0.0&quot; /&gt;
         </web:requestData>
      </web:fetchMandators>
   </soap:Body>
</soap:Envelope>
```

3.2 Sicherheit

3.2.1 Verschlüsselte Kommunikation

Die Nutzung des Webservice ist ausschließlich über eine mittels TLS-Verschlüsselung gesicherte HTTPS-Verbindung möglich. Dadurch wird die Abhörsicherheit und Integrität der gesendeten Daten gewährleistet.

Des Weiteren ist die Kommunikation dadurch auch vor verschiedenen anderen Attacken geschützt. Beispielhaft sei hier die Replay-Attacke genannt, bei der ein Angreifer eine abgefangene Nachricht erneut an den Empfänger schickt. Dies wird durch die Verwendung von Sequenznummern erkannt und ein derartiger Angriff abgewehrt. Es sei dabei erwähnt, dass der Angreifer die wiederholt gesendeten Pakete nicht entschlüsseln und damit verstehen kann.

3.2.2 Server-Authentifizierung

Um zu gewährleisten, dass tatsächlich mit dem richtigen Server und nicht einer unberechtigten Stelle kommuniziert wird, wird das Zertifikat des Servers herangezogen, um diesen eindeutig zu authentifizieren (siehe dazu die Dokumentation zur Testumgebung unter: https://partner.speed4trade.com/entwickler/tools/ssl-zertifikat-download/ -> Sichere Verbindung zur Speed4Trade-CONNECT-API).

Achtung: Die Testumgebung nutzt ein anderes Zertifikat als die Kundenumgebungen.

3.2.3 Client-Authentifizierung

Um sicherzustellen, dass lediglich berechtigte Clients mit dem Webservice kommunizieren können, müssen sich diese per HTTP-Basic-Authentifizierung am Server anmelden.

Obwohl die Zugangsdaten bei diesem Verfahren offen im Header der Nachricht übertragen werden, ist es für die Verwendung in diesem Fall gut geeignet, da es auf der Client-Seite einfach zu implementieren ist, und die übertragenen Zugangsdaten aufgrund der Kommunikation über HTTPS ohnehin nicht abgehört werden können.

3.3 Entwicklungsframework

Insbesondere wenn eine ERP-eigene Entwicklungsumgebung verwendet werden soll, sind entwicklungstechnisch vorab folgende Fragen zu klären:

- steht eine vollwertige Entwicklungsumgebung zur Verfügung:
  - Unterstützung von SOAP/XML Nachrichten
  - Automatische Code-Generierung für Datenobjekte aus WSDL Dateien
  - Versionierung und Debugging für Pflege und Wartung
- können Daten in einer Datenbank zwischengespeichert werden (Grunddaten, Zwischenergebnisse, temporäre Abfragedaten, etc.)
- ist das Protokollieren der Webservice-Aufrufe über eine Log-Datei möglich
- kann auf Übertragungsfehler, z.B. durch Neusenden bei nicht Erreichbarkeit reagiert werden
- kann eine Programm-Logik über mehrere API-Calls hinweg aufgebaut werden
- steht neben dem Produktiv- auch ein Testsystem zur Verfügung.

Bedenken Sie, dass eine Webservice-API für den automatischen Datenaustausch großer Datenmengen gedacht ist und somit stabil, performant und zuverlässig funktionieren muss. Planen Sie daher einen modularen Aufbau des SOAP-Clients mit standardisierter Controller/Adapter Systematik. Dies erleichtert Ihnen auch die Aktualisierung des SOAP-Clients bei Erweiterungen der Webservice API, denn auch diese entwickelt sich permanent weiter.

3.4 Programm-Logik

Neben den Routinen zur reinen Datenabfrage, sollten Sie auch noch mindestens Programm-Module für nachfolgende Aufgaben vorsehen:

- Code-Generator, der aus der WSDL Datei Basisklassen in Ihrer Programmiersprache erzeugt.
- Modul zur Authentifizierung des SOAP-Clients am Speed4Trade CONNECT Webservice.
- Programm zum Auslesen statischer Daten (z.B. Mandanten-ID). Es reicht hier in der Regel die Daten bei Änderungen, sprich bei Bedarf oder einmal z.B. wöchentlich zu aktualisieren. Diese Daten sollten im ERP oder SOAP-Client zwischengespeichert werden, damit sie nicht erst beim dynamischen Datenaustausch (z.B. beim Abholen der Aufträge) zusätzlich über den Webservice geladen werden müssen.
- Generelle Routinen zum Speichern von Ergebnissen oder Zwischenergebnissen in einer Datenbank.
- Routine zur Synchronisierung von Daten, welche für weitere Abfragen benötigt werden.
- Mapping-Routinen zum Matchen von Datensatz IDs aus Speed4Trade CONNECT, zu den Datensatznummern der eigenen Lösung, um die korrekte eine Zuordnung sicherzustellen.
- Controller zum Verarbeiten von Ergebnissen, Ereignissen und Fehlern mit der Möglichkeit diese kontrolliert zu verarbeiten und den API Call ggf. erneut auszuführen.
- Zentrale Logging-Routine zur Protokollierung aller Funktionsaufrufe, Fehler und Debug-Meldungen.

Ein ausgereifter Logging-Mechanismus ist umso wichtiger, wenn der SOAP-Client permanent im Hintergrund läuft. Nur dadurch können fehlerhafte Calls umgehend erkannt und verbessert werden.

Dabei muss ein Call nicht unbedingt komplett vom Webservice abgewiesen werden. Denken Sie nur z.B. daran, dass Sie 20 Artikel übergeben und einer fehlerhafte Daten enthält. Ohne detailliertes Logging haben Sie keine Chance diese Fehler zeitnah zu ermitteln.

Hinweise!
Zu Vereinfachung der Schnittstellenentwicklung, stehen ihnen optional im Partnerprotal XSD Dateien zur Verfügung.

Achtung! Zur Sicherstellung der Abwärtskompatibilität Ihrer Schnittstelle sind die bereitgestellten XSDs nicht zur Validierung zu verwenden!

Für die Entwicklung und Fehleranalyse kann Speed4Trade CONNECT-seitig ein Logging der SOAP Nachrichten über den Webservice eingeschaltet werden. Wenden Sie sich dazu bitte an unseren Support. Achtung! Schalten Sie das Logging in Live-Betrieb beim Kunden unbedingt wieder aus, da das Logging die Geschwindigkeit des Gesamtsystems beeinflusst.

Zu bedenken ist auch, dass über manche Webservice Calls nicht nur Daten übertragen, sondern in der Folge auch weitere Funktionen in der Speed4Trade CONNECT ausgelöst werden. Wenn Sie zum Beispiel einen Versandstatus im Auftrag setzen, wird dieser sofort über einen Abgleich mit dem Marktplatz auch automatisch an den Marktplatz übertragen. Ohne qualitativ hochwertigen SOAP-Client und einem professionellen Logging ist hier auf Dauer keine stabile Lösung zu erzielen.

Neben dem rein technischen Teil, gilt es auch im Vorfeld der Entwicklung eine fachliche Bedarfsanalyse durchzuführen. Welche Daten und Funktionen werden überhaupt benötigt und auf welche Webservice Calls kann man sich daraus konzentrieren?

3.5 Bedarfsanalyse

Bevor Sie mit der Entwicklung eines SOAP-Clients beginnen, sollten Sie, wie bei jedem Projekt, eine genaue Planung und Bedarfsanalyse durchführen. In der Standardkonfiguration gehen wir davon aus, dass das ERP-System/die Warenwirtschaft das führende System ist. Das heißt, das ERP übergibt aktiv die Artikeldaten (inkl. Bilder) und aktuelle Lagerbestände. In Speed4Trade CONNECT werden danach die Plattform-spezifischen Daten ergänzt und als Angebot in die Verkaufskanäle eingestellt. Das ERP wiederum holt aktiv die eingegangenen Bestellungen von Speed4Trade CONNECT ab und setzt je nach Auftragsworkflow den Status für geliefert und bezahlt weiter. Dieser wird dann durch Speed4Trade CONNECT wieder an die Verkaufskanäle (Marktplätze und Shops) weitergereicht.

Folgende Fragen helfen Ihnen bei der Bedarfsanalyse weiter:

- Wie viele Mandanten sind vorhanden und sollen bedient werden?
- Welche Verkaufskanäle (eBay.de, Amazon.de, Shops, etc.) sollen angebunden werden?
- Welche Accounts gibt es pro Verkaufskanal?
- Welche Artikelstammdaten sind bereits vorhanden, welche müssen ergänzt werden?
- Gibt es erweiterte Beschreibungen zu den Artikeln?
- Sind im ERP-System bereits Plattform-spezifische Daten (z.B. Artikelattribute) vorhanden?
- Wo und wie sind die Bilder zu den Artikel gespeichert (lokal, per URL erreichbar)?

- Wird mit Varianten-Artikeln gearbeitet und sind diese im ERP bereits vollständig angelegt?
- Soll mit Sets, Cross-Selling und Zubehör gearbeitet werden?
- Welche Preise, Sonderpreise, UVP sollen angeboten werden; werden Grundpreise benötigt?
- Wie wird der aktuelle Lagerbestand ermittelt, gibt es ein oder mehrere Lger?
- Wie sind die Versandkosten und Versandkostenkategorien strukturiert?
- Sind die Artikel bereits in Warengruppen einsortiert und können diese für die Zuordnung von Kategorien auf den Verkaufskanälen verwendet werden?
- Welche Auftragsworkflows sind definiert ("Vorkasse-Workflow" -> erst bezahlen dann liefern und/oder "Rechnungs-Workflow"-> erst liefern dann bezahlen)?
- Welche Auftragsdaten und Statusänderungen können verarbeitet und gesetzt werden?
- Welche Zahlungsarten werden verwendet und wo findet der Zahlungsabgleich statt?
- Wo findet die Übergabe der Auftragsdaten an die Versanddienstleister statt und wie wird die Tracking-ID zur Sendungsverfolgung zurück geliefert?
- Wie werden eingehende Kundendaten (Duplikatsprüfung) aus den Aufträgen zugeordnet?

Durch die Beantwortung obiger Fragen können Sie Ihren Bedarf gezielt ermitteln und in Anwendungsfällen dokumentieren. Darauf aufbauend können dann leichter die dafür notwendigen API-Calls ermittelt werden. Beginnen wir mit den "Vorbereitenden API Calls" wie z.B. das Auslesen der Mandanten ID, die bei jeden Folge-Call fast immer benötigt wird.

4 Funktionsübersicht

4.1 Grundeinstellungen, Stammdaten und Angebotslisting

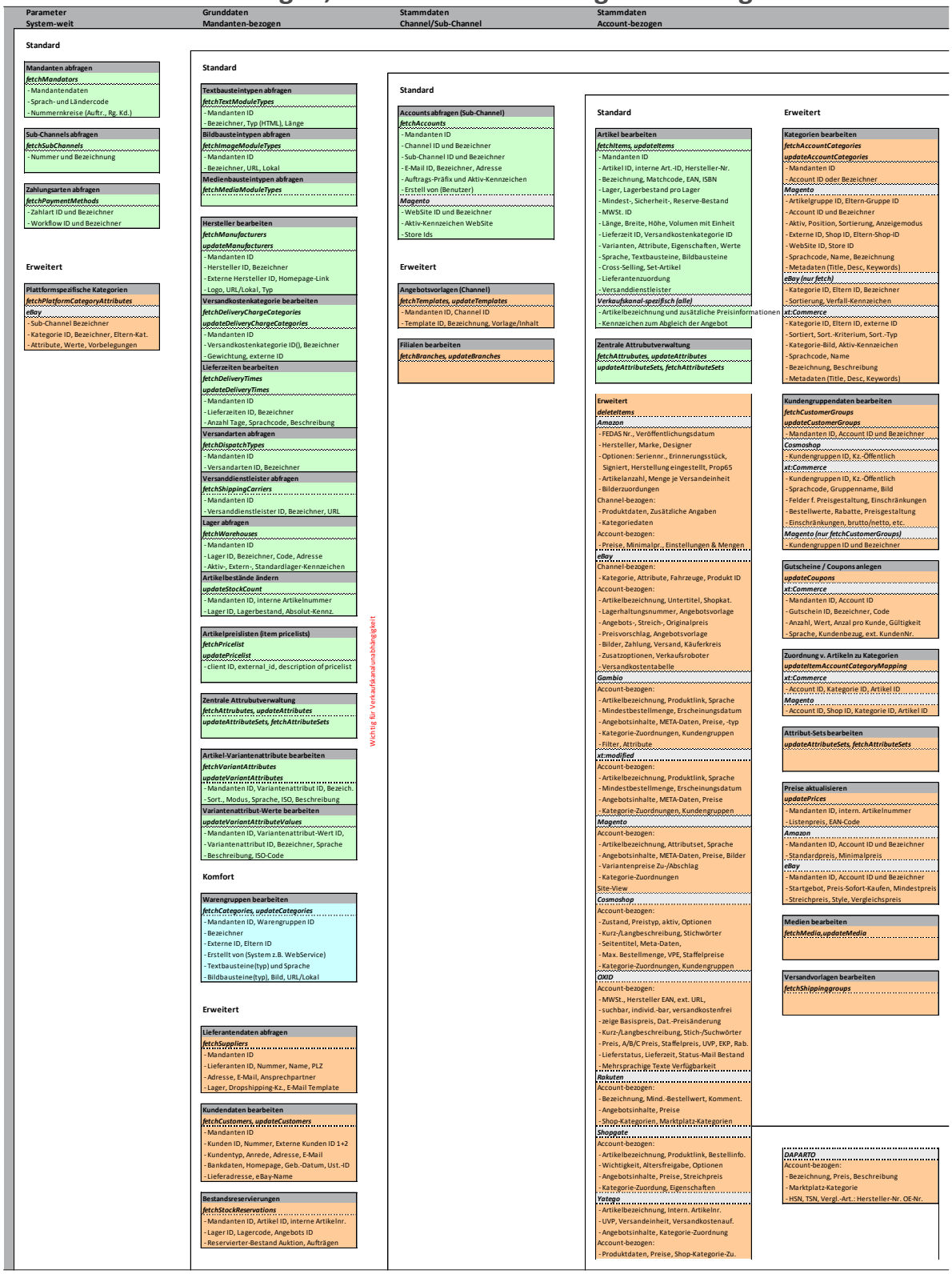

Abbildung 4: Funktionsübersicht - Grundeinstellungen, Stammdaten und Listing

4.2 Angebots- und Auftragsworkflow

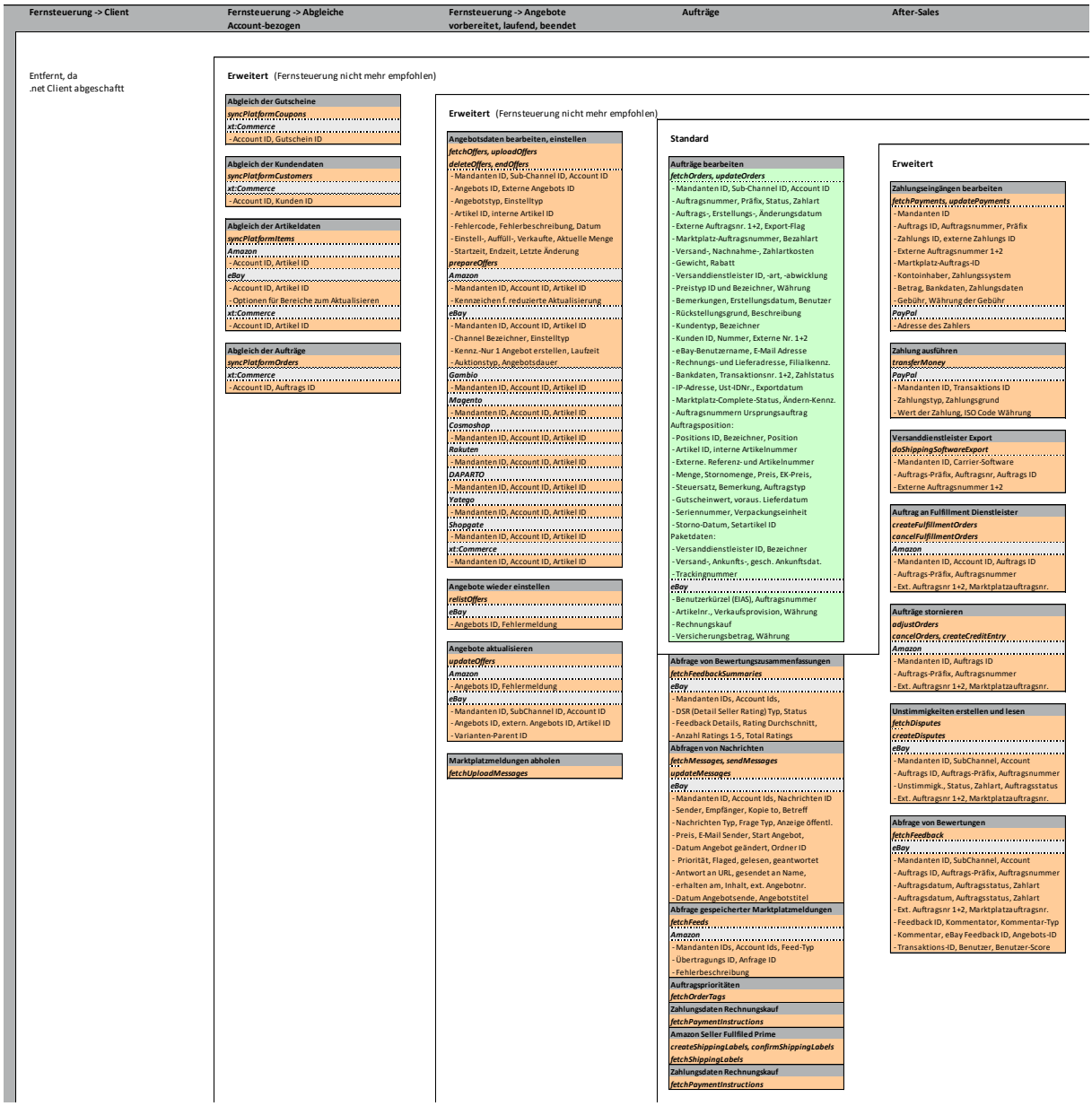

Abbildung 5: Funktionsübersicht - Angebots- und Auftragsworkflow

In den Funktionsübersichten sind die einzelnen Webservice API Calls jeweils farbig unterlegt. Wobei die Farbe

- Grün für die Standard Anbindung,
- Blau für die Komfort Funktion über die Warengruppen und
- Rot für die Erweiterte Anbindung steht.

Sie sollten zunächst mit einer Standard-Implementierung beginnen. Gegebenenfalls die Komfort-Funktion über die Zuweisung von Artikeln zu Warengruppen hinzufügen und erst in einer Phase 2, erweiterte Funktionen mit einbinden.

Die Übersicht steht als Excel Datei auch unter folgendem Link zur Verfügung:
https://partner.speed4trade.com/entwickler/integration-starten/integrationsleitfaden/

5 Wichtige Grundeinstellungen - vorbereitende API Calls

Wie bereits schon angesprochen gibt es einige Daten, wie z.B. die Mandanten-Nummer, die sich selten oder nie ändern. Diese Daten sollten unabhängig vom laufenden Datenaustausch im Vorfeld ermittelt und im SOAP-Client oder dem ERP-System z.B. in einer Konfigurationstabelle zwischengespeichert werden. Die Erfassung und Änderung dieser Daten erfolgt in der Regel in Speed4Trade CONNECT. Die Aktualisierung sollte daher so realisiert werden, dass sie bei Bedarf, sprich bei Änderungen oder regelmäßig z.B. einmal wöchentlich erfolgt. Dies erhöht die Performance des SOAP-Clients erheblich, wenn z.B. beim Aktualisieren eines Artikels nicht vorher erst die dazu auch notwendige Mandanten-Nummer ermittelt werden muss.

Hinweis!
Pflichtfelder können über den Speed4Trade-CONNECT-Client ermittelt werden. Dazu ist es ausreichend beim gewünschten Bereich, z.B. Artikel, einfach die Funktion „Neu“ aufzurufen. In der dann eingeblendeten Erfassungsmaske werden alle Pflichtfelder mit einen „gelben“ Balken am Anfang des Eingabefeldes angezeigt.

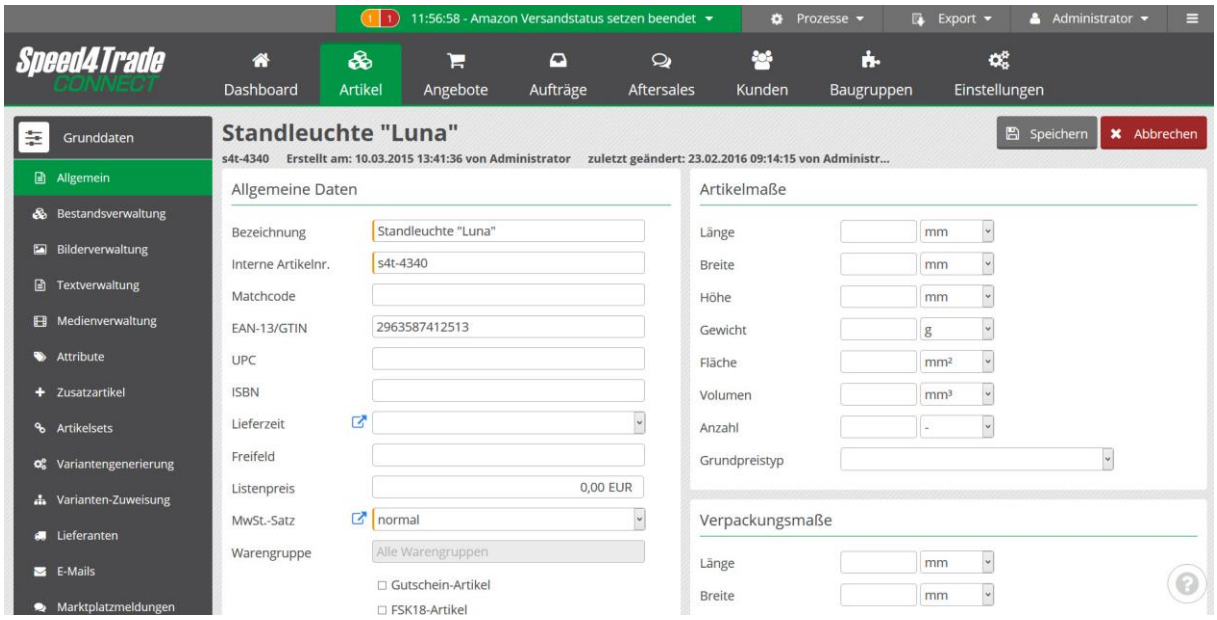

Abbildung 6: Pflichtfelder sind mit einem "gelben" Balken hinterlegt

Es macht durchaus Sinn, während der Programmierung des Webservices immer wieder mit dem Client und dem ***Speed4Trade CONNECT*** Online-Handbuch zu arbeiten, um die Anwendung der Datenfelder besser verstehen zu lernen. Für weitere Optionen, wie bspw. die Änderung der anzuzeigenden Sprache, navigieren Sie zum „Mehr"-Button.

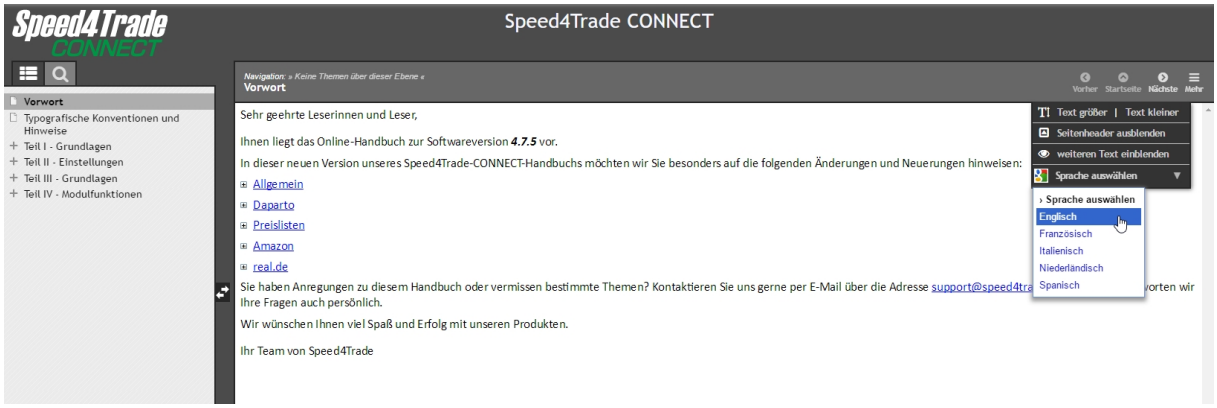

Abbildung 7: Weitere Optionen, bspw. Sprachauswahl im Speed4Trade Online Handbuch

Nehmen Sie spätestens jetzt auch die ***Speed4Trade-CONNECT-API-Dokumentation*** (https://partner.speed4trade.com/entwickler/api-webservice/) zur Hand. Nachfolgend erhalten sie eine Zusammenfassung von vorbereitenden API Calls der wichtigsten Grundeinstellungen.

Calls für die reine Abfrage von Grundeinstellungen:

- fetchMandators (Mandanten)
- fetchSubChannels (Sub-Channels)
- fetchAccounts (Accounts)
- fetchTextModuleTypes (Textbausteintypen)
- fetchImageModuleTypes (Bildbausteintypen)

Calls zum Bearbeiten von Grundeinstellungen, die Plattform-spezifisch schon vorbelegt werden können:

- fetchDeliveryChargeCategories/updateDeliveryChargeCategories (Versandkostenkategorien)
- fetchDeliveryTimes/updateDeliveryTimes (Lieferzeiten)
- fetchManufactures/updateManufactures (Hersteller)
- fetchTemplates (Angebotsvorlagen)

Calls für die Abfrage Auftrags-bezogener Stammdaten:

- fetchPaymentMethods (Zahlungsarten)
- fetchDispatchTypes (Versandarten)
- fetchShippingCarriers (Versanddienstleister)

5.1 Mandanten

Speed4Trade CONNECT Client Pfad: Einstellungen → Benutzerverwaltung → Mandanten

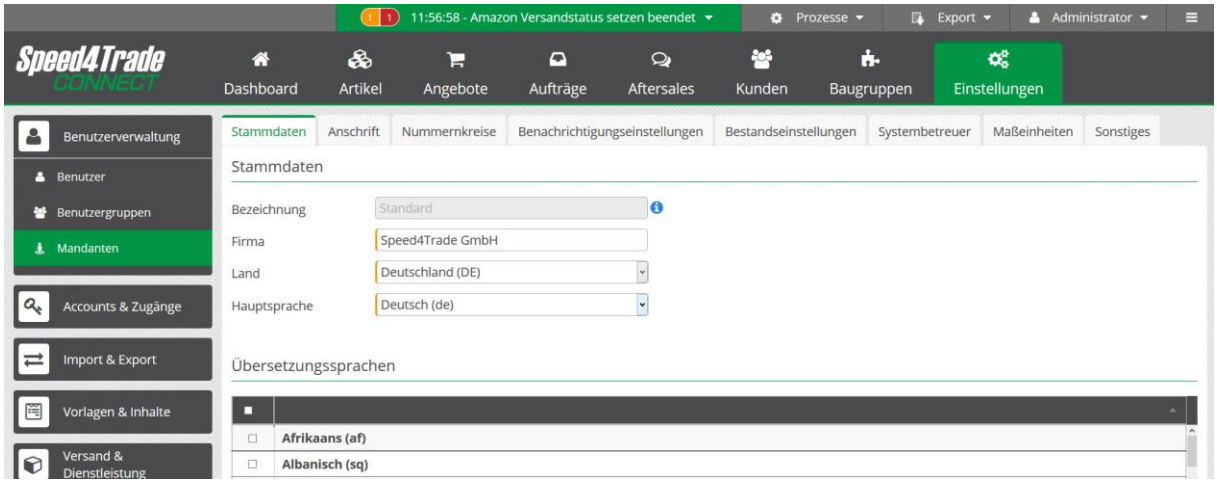

Abbildung 8: Mandanten-Verwaltung

Speed4Trade CONNECT ist Mandanten-fähig. Einem „Benutzer“ der Software können mehrere Mandanten zugeordnet werden. Beim Login muss sich der Anwender für den Mandanten entscheiden, mit dem er arbeiten will. Alle Daten innerhalb der Software sind strikt nach Mandanten getrennt. Im Umkehrschluss bedeutet dies, dass man in jedem Webservice-Call angeben muss, für welchen Mandanten diese Daten gesetzt werden sollen.

Beim Mandanten werden ebenfalls die „Übersetzungssprachen“ festgelegt, d.h. hier wird definiert, für welche Sprachen mehrsprachige Texte bspw. bei einem Artikel hinterlegt werden können.

5.1.1 WebService-Call: fetchMandators (Mandanten)

Standard

Mit dem ersten Funktionsaufruf müssen zunächst die Mandanten-Daten ermittelt werden, da diese für alle weiteren Funktionsaufrufe benötigt werden. Somit ist auch sichergestellt, dass alle weiteren Aufrufe sich auch auf den richtigen Mandanten beziehen.

Mindestens die Mandanten-IDs und die Übersetzungssprachen (ISO-Sprachkürzel) für die mehrsprachigen Inhalte (z.B. Textbausteine) müssen eingelesen bzw. verarbeitet werden. Die Mandanten und Übersetzungssprachen können relativ starr in der Warenwirtschaft hinterlegt werden (z.B. in einer Konfigurationsdatei/-tabelle), da sie sich selten ändern.

5.1.1.1 Sprachen

Neben der Hauptsprache können - falls notwendig - auch noch weitere Übersetzungssprachen (z.B. Englisch, Französisch, etc.) in Speed4Trade CONNECT angegeben werden.

5.1.1.2 Währungen

Die Währungen für die Artikelpreise sind direkt mit dem Länder-spezifischen Verkaufskanal verbunden (z.B. CHF bei eBay.ch) und brauchen daher nicht gesondert ermittelt werden.

5.2 Verkaufskanäle

Als Verkaufskanäle werden die übergeordneten Marktplatz-Namen und Shops bezeichnet; zum Beispiel Amazon oder eBay. Die unterstützten Verkaufskanäle sind über die Lizenzierung fest in der Speed4Trade CONNECT hinterlegt (und können auch nicht über den Speed4Trade CONNECT Client geändert werden). Da aber alle Aktionen auf die Länder-spezifischen Seiten, wie eBay.de oder eBay.at, abgestimmt werden müssen, kann gleich mit den entsprechenden „SubChannels“ gearbeitet werden. Ein Call für „Channels“ ist nicht verfügbar und auch nicht notwendig.

5.3 SubChannels

Mit Speed4Trade CONNECT ist es möglich, Artikel in verschiedene Länder-spezifische Seiten der Verkaufskanäle einzustellen. Einige Einstellungen beim Artikel müssen direkt für einen Verkaufskanal gesetzt werden. Näheres dazu wird bei der Artikelverwaltung erklärt.

Um alle derzeitig unterstützten Verkaufskanäle abzurufen, benötigt man den Call *fetchSubChannels.*

5.3.1 Webservice-Call: fetchSubChannels (Sub-Channels)

Standard

Die über *fetchSubChannels* ermittelten SubChannel-IDs können relativ starr in der Warenwirtschaft hinterlegt werden, da sich hier selten etwas ändert. Es sollte allerdings möglich sein, SubChannels einfach hinzuzufügen (z.B. über eine Konfigurationsdatei/-tabelle), wenn der Kunde auf weiteren Kanälen aktiv werden will.

5.4 Accounts

Speed4Trade CONNECT Pfad: Einstellungen → Accounts & Zugänge

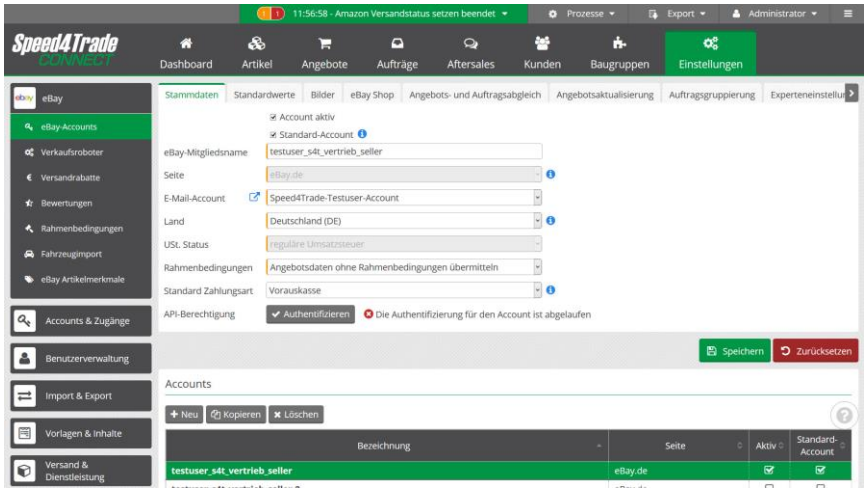

Abbildung 9: Accountverwaltung

Je nach Tarif kann eine unterschiedliche Anzahl von Accounts pro SubChannel gepflegt werden. So kann ein Kunde z.B. auf eBay.de mehrere Verkäufer-Accounts angelegt und im Einsatz haben. Ähnlich wie bei den Verkaufskanälen ist es auch bei den Accounts so, dass bestimmte Einstellungen zu einem Artikel je Account gepflegt werden können, wie bspw. der Preis des Artikels, mit dem dieser Artikel in einem Verkaufskanal zu diesem Account eingestellt werden soll.

5.4.1 WebService-Call: fetchAccounts (Accounts)

Standard

Auch hier können die über *fetchAccounts* ermittelten Account-IDs der Marktplätze relativ starr in der Warenwirtschaft hinterlegt werden, da sich hier selten etwas ändert. Es sollte allerdings möglich sein, weitere Accounts (z.B. über eine Konfigurationsdatei/-tabelle) hinzuzufügen, wenn der Kunde weitere Accounts verwenden will.

5.5 Textbausteintypen

Speed4Trade CONNECT Pfad: Einstellungen → Vorlagen & Inhalte → Textbausteintypen

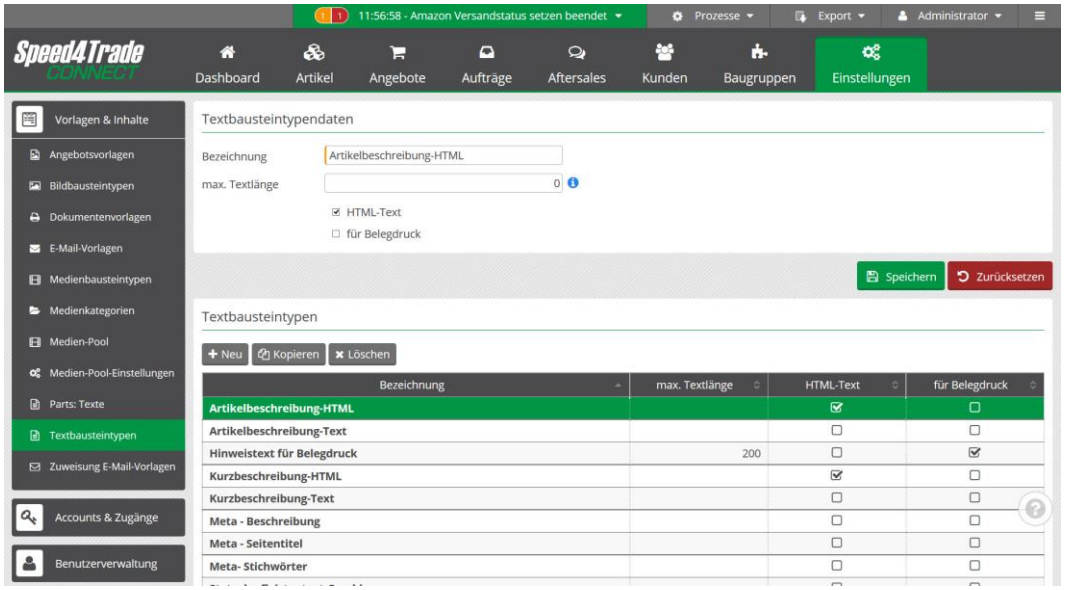

Abbildung 10: Textbausteintypen

Speed4Trade CONNECT arbeitet beim Generieren der Angebotsinhalte und Beschreibungen mit Textbausteinen. Dies ermöglicht ein Maximum an Flexibilität. Über Textbausteine können bei Bedarf pro Artikel unterschiedliche, speziell auf die Anforderungen der Marktplätze abgestimmte, Beschreibungen erstellt werden.

Die Anzahl der Beschreibungsfelder für einen Artikel ist nicht fest definiert, d.h. jeder Anwender kann selbst entscheiden, wie viele Felder er für die Darstellung seiner Artikel benötigt. Dies führt aber wiederum dazu, dass bei der Übergabe der Artikel aus dem externen System beim Webservice-Call angegeben werden muss, auf welchen Textbausteintyp bspw. der Langtext gesetzt werden soll.

5.5.1 WebService-Call: fetchTextModuleTypes (Textbausteintypen)

Standard

Dabei sind die 2 Textbausteintypen HTML- und Plain-Text zu unterscheiden:

- Benötigte Textbausteintypen:
  - Textbausteine für HTML-Text → für eBay und Shop
  - Textbausteine für eingeschränktes HTML[1] → Amazon, Yatego, Rakuten
  - Textbausteine für Plain-Text → für alle Kanäle

- Mögliche zusätzliche Textbausteine (Beispiele)
  - Kurzbeschreibung (HTML-Text)
  - Langbeschreibung (HTML-Text)
  - Kurzbeschreibung (Plain-Text)
  - Langbeschreibung (Plain-Text)
  - Technische Daten (HTML-Text)
  - Meta-Daten-Seitentitel (Plain-Text)
  - Meta-Daten-Beschreibung (Plain-Text)
  - Meta-Daten-Stichwörter (Plain-Text)

Die Textbausteintypen werden zunächst in Speed4Trade CONNECT definiert. Die Pflege der Werte für bspw. Kurz- und Langbeschreibung erfolgt in der Warenwirtschaft. Es muss zwischen den Texten aus der Warenwirtschaft und den Textbausteintypen ein Mapping erstellt werden. Dies erfolgt über die IDs je Textbaustein in der Warenwirtschaft.

Hinweis!
Sind mehrsprachige Artikeltexte gewünscht, können und müssen diese in weiteren, sprachabhängigen Textbausteinen gepflegt werden. Das Mapping der verschiedenen Felder mit den unterschiedlichen Sprachtexten hat dann bei der Übergabe der Texte im Artikel an Speed4Trade CONNECT anhand eines ISO-Sprachcodes zu erfolgen.

[1] siehe auch Tabelle: Übersicht über marktplatzspezifische Eigenheiten (Kapitel 9)

5.6 Bildbausteintypen

Speed4Trade CONNECT Pfad: Einstellungen → Vorlagen & Inhalte → Bildbausteintypen

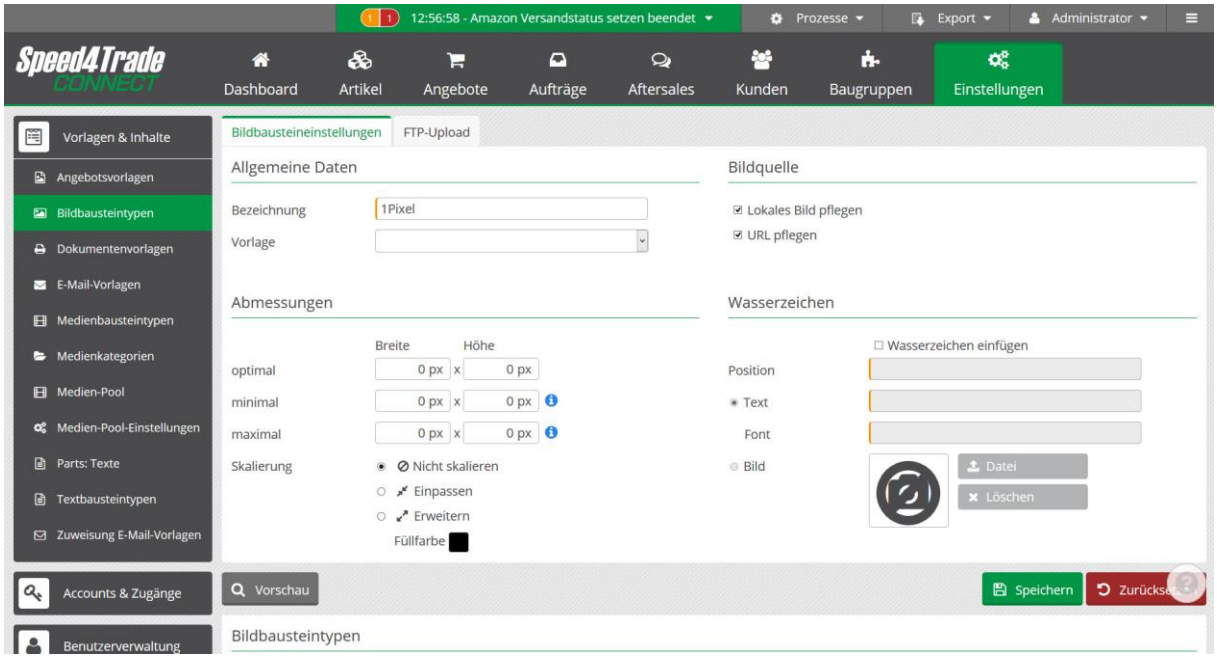

Abbildung 11: Bildbausteintypen

Ähnlich wie bei den Textbausteintypen verhält es sich auch mit den Bildbausteinen, d.h. der Anwender entscheidet selbst, wie viele Bilder er bei den Artikeln hinterlegen will. Über die Typen können sehr flexibel unterschiedliche – ggf. auch Markplatz-spezifische – Bilder pro Artikel hinterlegt werden. Bilder werden immer anhand der hinterlegten Konfiguration skaliert.

Dies führt auch wiederum dazu, dass bei der Übergabe der Artikel aus dem externen System beim Webservice-Call angegeben werden muss, auf welchen Bildbausteintyp bspw. das Hauptbild gesetzt werden soll.

5.6.1 Webservice-Call: fetchImageModuleTypes (Bildbausteintypen)

Standard

Dabei sind die 2 Bildbausteintypen UNC- und URL-Bild zu unterscheiden:

- Benötigte Bildbausteintypen:
  - UNC-Bilder / lokale Bilder → z.B. für eBay und Shop
  - URL-Bilder → z.B. für Amazon, eBay und Yatego, Rakuten

- Mögliche Bildbausteine (Beispiele)
  - Kleineres Bild für Galerieansicht
  - Großes Bild mit maximaler Auflösung für Zoomfunktion
  - Speziell angepasste Bilder pro Verkaufskanal (eBay, Amazon, etc.)

Die Bildbausteintypen werden zunächst in Speed4Trade CONNECT definiert. Die Pflege erfolgt über die Bildfunktionalitäten in der Warenwirtschaft. Es muss zwischen den Bildern aus der Warenwirtschaft und den Bildbausteintypen ein Mapping erstellt werden. Dies erfolgt über die ID je Bildbaustein in der Warenwirtschaft. Die Bilder sollten optimaler Weise bereits in der Warenwirtschaft in der richtigen Auflösung (größer 600 Pixel) und Qualität hinterlegt sein und übergeben werden.

Hinweis!
URL-Bilder werden bevorzugt, da sie von allen Verkaufskanälen verwendet werden können. Lokale Bilder sind nur bei eBay und Shops möglich. Je nach Verkaufskanal können bis zu 12 Bilder pro Artikel verwendet werden. UNC Pfade sind lokale Angaben und keine URL Bilder.

Optional können auch lokale Bilder über den in Speed4Trade CONNECT integrierten Bilderupload auf einen FTP-Account als URL-Bilder bereitgestellt werden.

Wenn Sie Bilder als Base64 übergeben, kann es bei großen Datenmengen (teilweise schon ab 2 MB pro Request) zu Problemen kommen, da dies die Standard-Einstellung im TomCat ist und Requests in dieser Größenordnung nicht angenommen werden.

5.7 Medienverwaltung

Damit beispielsweise PDF-Dokumente und Produktvideos einem Artikel zugewiesen und diese auf die jeweiligen Marktplätze und in Onlineshops hochgeladen werden können, ist in Speed4Trade CONNECT eine Medienverwaltung integriert.

Die Zuweisung der Medien zu den Artikeln kann derzeit nur in Speed4Trade CONNECT erfolgen und kann somit nicht durch die Warenwirtschaft über den Webservice gesetzt werden. Diese Funktionalität wird zukünftig aber noch erweitert.

5.8 Anwendung von Bild-, Text- und Medienbausteinen in Angeboten

Die in den Grundeinstellungen der Software pflegbaren Bild-, Text- und Medienbausteintypen erhalten über die Pflege der Bild-, Text- und Mediendateien im Artikel die artikelspezifischen Angebotsdaten. Die folgende Grafik veranschaulicht die grundlegende Verknüpfung von Bausteinen, Platzhaltern, Artikeln und Angeboten. Hierbei ist zu beachten, dass es z.T. grundlegende Unterschiede bei dieser Systematik in Abhängigkeit von der gewählten Verkaufsplattform gibt. Beachten Sie hierzu die Besonderheiten weiter unten in diesem Abschnitt.

Grundsätzlich gilt: Zum Erstellen eines Angebots muss immer ein Artikel vorhanden sein. Er stellt das zentrale Objekt dar, von dem aus ein Angebot vorbereitet werden kann.

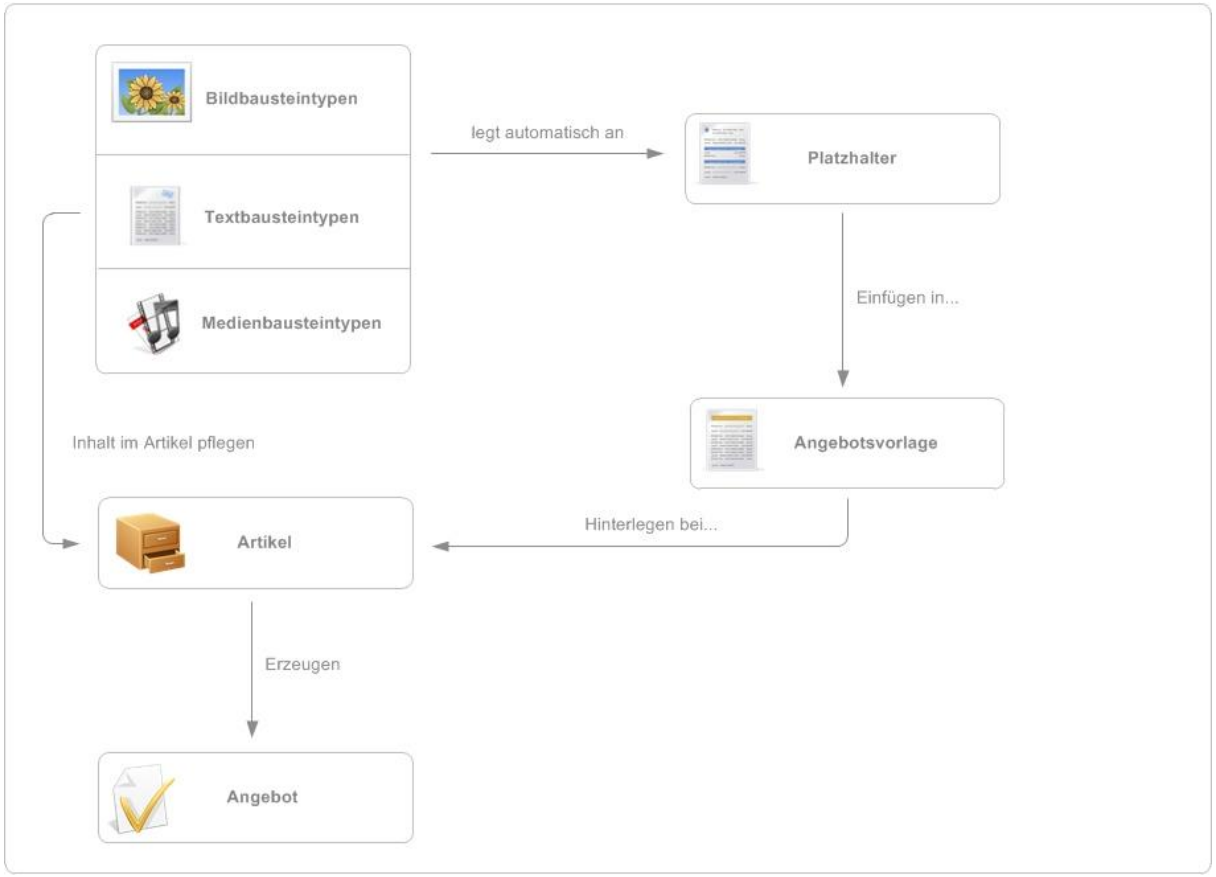

Abbildung 12: Anwendung von Bild-, Text- und Medienbausteinen in Angeboten

Erläuterungen:

- Wird ein Text-, Bild- oder Medienbausteintyp angelegt, so ist dieser ab dem Zeitpunkt der Erstellung im Artikel zur Pflege verfügbar.
- Wird ein Text-, Bild- oder Medienbausteintyp angelegt, so erstellt die Software hierfür automatisch einen Platzhalter, der wiederum in Angebotsvorlagen eingefügt werden kann.
- Die Vorlage und somit die dort eingebundenen Platzhalter werden im Artikel beim jeweiligen Verkaufskanal ausgewählt.

- Beim Erzeugen / Vorbereiten des Angebots werden die Platzhalter durch die Text- oder Bilddaten ersetzt. Ist der Inhalt des jeweiligen Bausteins im Artikel leer, so wird dieser Platz im erzeugten Angebot leer gelassen, ohne den Platzhalter auszugeben.

5.9 Zentralisierte Attributverwaltung (ZAV)

Standard

Mit der Version Speed4Trade CONNECT 3.5.37 wurde erstmals die ZAV eingeführt. Jeder Verkaufskanal ist etwas anders. Wir wollen die Komplexität daher durch eine verkaufskanalunabhängige Zentralisierung der Datenpflege reduzieren. Dazu sind in Zukunft drei Schritte notwendig:

- Attribute und deren Werte anlegen (Stammdaten)
- Matching zu den plattformspezifischen Attributen und Werten
- Attributwerte dem Artikel zuweisen

Im Standard ist die ZAV in Speed4Trade CONNECT zu nutzen (oder Masken über das Mashup einbinden). Als erweiterte Funktion können diese drei Schritte auch über die Speed4Trade CONNECT API direkt aus dem externen System (z.B. PIM) angesprochen werden.

Neue Verkaufskanäle wie Hitmeister, Locafox und Websale basieren bereits ausschließlich auf dieser neuen Systematik. Eine vollständige verkaufskanalunabhängige Schnittstelle ist erreicht, wenn in Zukunft auch noch eine „Preislisten Funktion“ implementiert ist.

Hinweis:
Die ZAV kann nicht mit dem bisherigen Vorgehen gemischt werden. D.h. wenn ein Verkaufskanal die ZAV in einem Call nutzt, müssen alle Verkaufskanäle die bereits umgesetzt sind die ZAV ebenfalls verwenden.

Speed4Trade CONNECT Pfad: Einstellungen → Kennzeichnung → Attribute

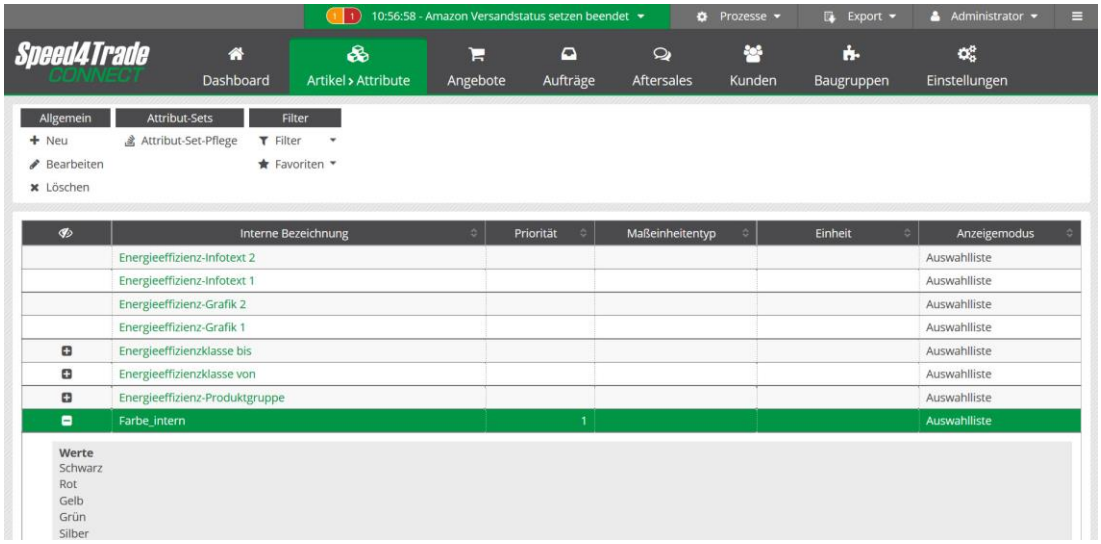

Abbildung 13: Anwendung von zentralen Attributen

Im Folgenden sehen Sie ein Beispiel im Überblick, wie die Zentralisierte Attributverwaltung integriert werden kann. Die Zuordnung der Attribute zum Artikel wird in Kapitel 7.2 nochmal aufgegriffen.

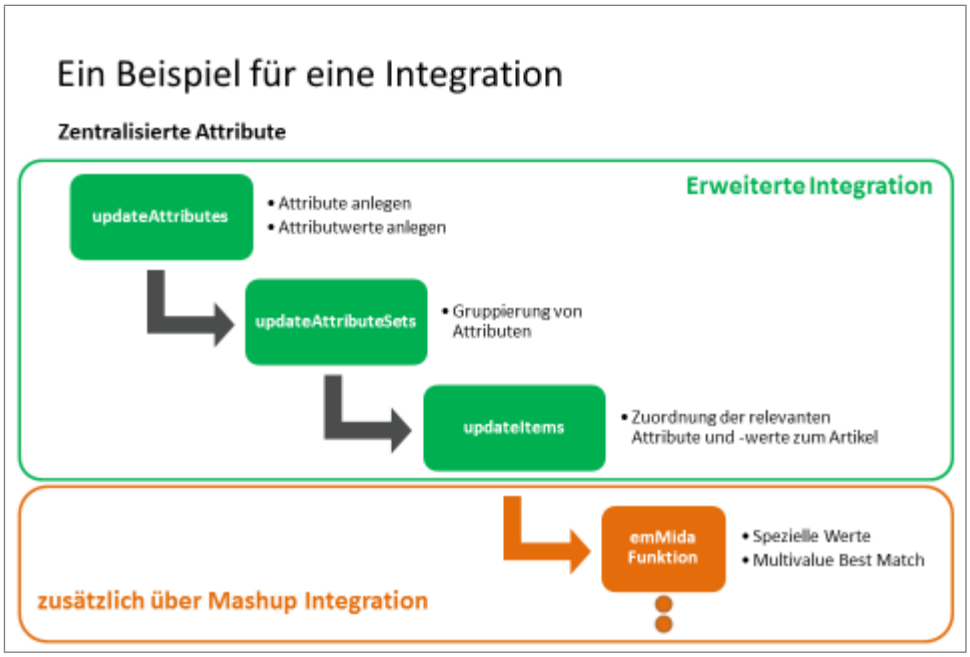

Abbildung 14: Beispiel für Integration von zentralen Attributen

5.9.1 WebService-Call: fetch / updateAttributes (zentrale Attribute und Werte)

- Pflege und Neuanlage der Attribute über neuen Call updateAttributes
- Vorhandene Attribute abholen über neuen Call fetchAttributes für Datenübernahme
- Große Anzahl vordefinierter Werte zu einem Attribut vermeiden: Freitext-Attribute verwenden
- Reihenfolge wird bei Neuanlage festgelegt, veränderbar über GUI
- attribute_identifier und attribute_id dienen als Schlüssel für weitere Calls

5.9.2 Webservice-Call: fetch / updateAttributeSets (zentrale Attributgruppierungen)

- In Attributsets können Sie Attribute in Sets gruppieren.
- Diese Sets lassen sich im Artikel auswählen und blenden dann die gruppierten Attribute im Artikel vor.
- Attributzuordnung über Attributnamen oder Speed4Trade CONNECT interne ID

5.10 Preislisten (updatePricelist)

Standard

Mit der Version Speed4Trade CONNECT 4.6.46 wurden erstmals Preislisten eingeführt. Jeder Verkaufskanal ist etwas anders. Die Umsetzung von Preislisten ist für eine verkaufskanalunabhängige Schnittstelle notwendig und für eine zukünftige Zertifizierung verpflichtend.

- Abholen von Preislisten und Preisen (Datenübernahmen oder Abgleich) – fetchPricelist
- Anlegen von Preislisten und Pflege von Preisen

Ziel ist es bei Ihrer Umsetzung, dass neue Verkaufskanäle in Speed4Trade CONNECT sofort in Ihrem System einsetzbar sind. Dazu muss es in Ihrer Umsetzung möglich sein, im ERP eine neue Preisliste anzulegen und Verkaufspreise zu pflegen. Die neue Preisliste muss per Konfiguration (ohne Programmierung Ihrer Schnittstelle) zugewiesen werden können.

Hinweis:
Bei Umstellung auf neue Preislisten bitte beachten:

- Mischformen möglich, d.h. Verwendung der Preisliste für eBay, Amazon und Speed4Trade COMMERCE, für die anderen Kanäle die bisherigen Methoden
- Preislistenpreise können nicht mehr in der Oberfläche von Speed4Trade CONNECT gepflegt werden. Preise müssen über die API oder über CSV-Dateien eine Basispreisliste (BPL) importiert werden. Somit schließt sich die Anwendung des Mashup-Integrationsmodells hierfür aus.
- Weitere Preise, wie z.B. Kundengruppen- und Staffelpreise, Streichpreise, Grundpreise, etc. werden noch wie bisher abgebildet.
- Preise für neue kommende Verkaufskanäle können nur noch über die Preislistenfunktion definiert werden.

Speed4Trade CONNECT Pfad: Einstellungen → Stammdaten → Preislisten

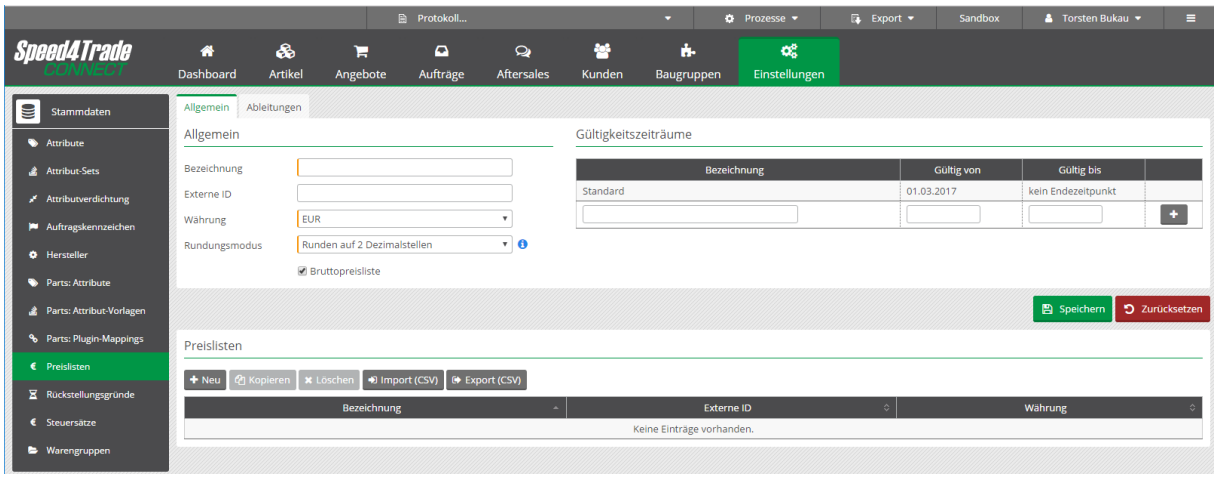

Abbildung 15: Basis- und abgeleitete Preislisten (BPL & APL)

Es gibt in Speed4Trade CONNECT die Funktion Basispreislisten (BPL) und Abgeleitete Preislisten (APL).

Eine Standardintegration nutzt die Basispreislisten (BPL) wie im folgenden Beispiel dargestellt.

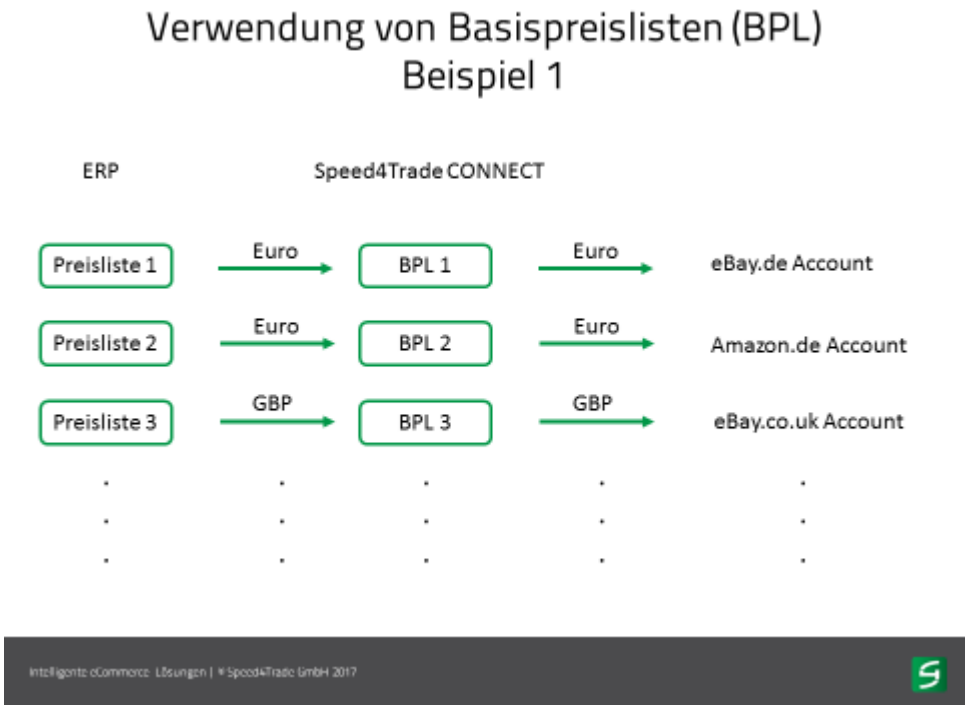

Abbildung 16: Anwendungsbeispiel von Basispreislisten

Wenn das externe System nicht mehrere Verkaufspreise für einen Artikel übergeben kann, können wir Ihnen im Integrationsworkshop oder auf Anfrage weitere Möglichkeiten für eine Umsetzung aufzeigen.

5.11 Versandkostenkategorien

Speed4Trade CONNECT Pfad: Einstellungen → Versand & Dienstleistung → Versandkostenkategorien

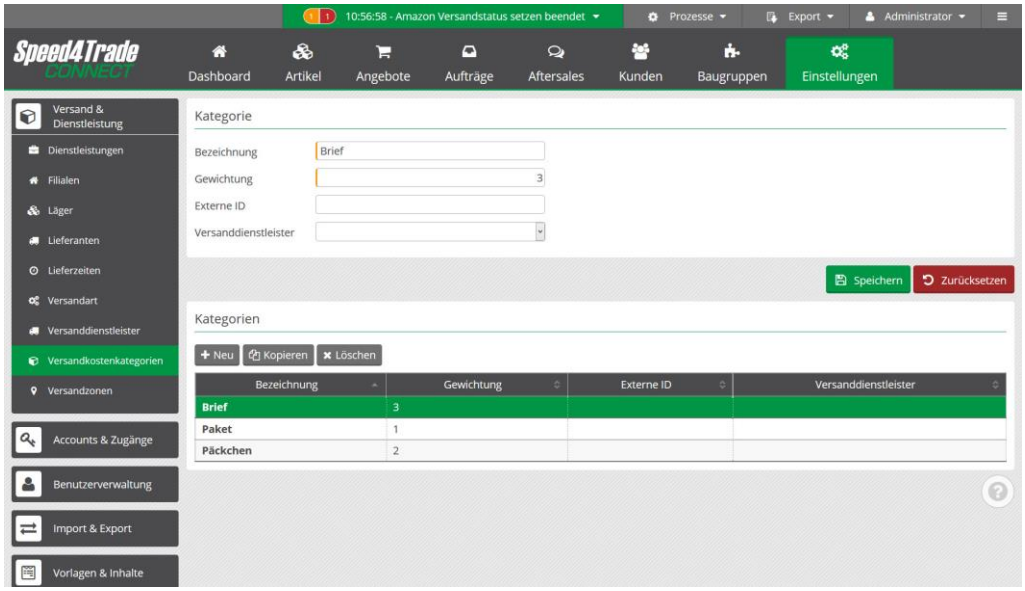

Abbildung 17: Versandkostenkategorien

5.11.1 Webservice-Call: fetchDeliveryChargeCategories / updateDeliveryChargeCategories (Versandkostenkategorien)

Standard

Die Versandkostenkategorie ist eine Eigenart des Marktplatzes eBay zur Definition der Versandkosten. Die Versandkostenkategorien können in Speed4Trade CONNECT erfasst werden. Somit ist ein Anlegen durch die Warenwirtschaft (*updateDeliveryChargeCategories*) nicht unbedingt notwendig. Die in Speed4Trade CONNECT vorhandenen Versandkostenkategorien sollten allerdings mit *fetchDeliveryChargeCategories* ausgelesen werden, um sie in einem späteren Schritt den Artikeln, die auf eBay eingestellt werden sollen, zuweisen zu können. Es reicht dazu die Speicherung des Datenfeldes *„delivery_charge_category_id“.*

5.11.1.1 Versandkosten

Bei den anderen Verkaufskanälen werden die Versandkosten meist direkt im Backend des Marktplatzes eingestellt.

5.12 Lieferzeiten

Speed4Trade CONNECT Pfad: Einstellungen → Versand & Dienstleistung → Lieferzeiten

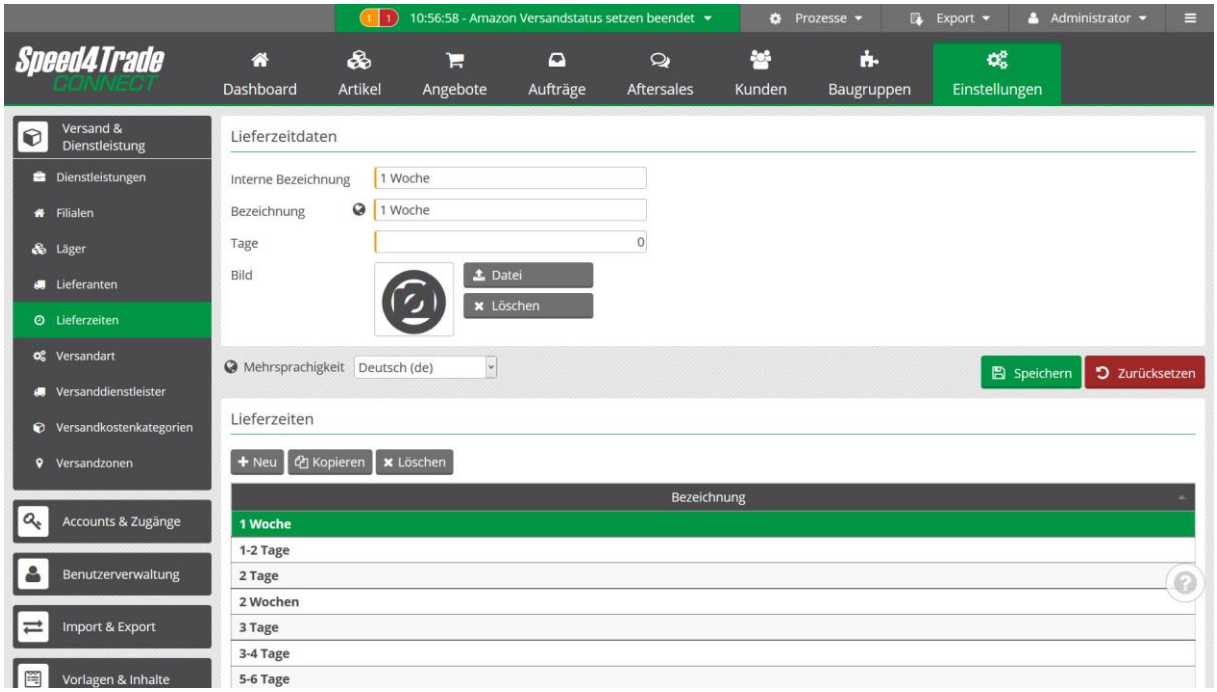

Abbildung 18: Lieferzeiten

5.12.1 Webservice-Call: fetchDeliveryTimes / updateDeliveryTimes (Lieferzeiten)

Standard

Die Definition von Lieferzeiten ist vor allem für Amazon- und Shop-Anbindungen wichtig. Wird z.B. bei Amazon keine Lieferzeit angegeben, geht Amazon automatisch von einer Lieferzeit von 1-2 Tagen aus.

Die Lieferzeiten werden normalerweise in Speed4Trade CONNECT gepflegt. Sind sie aber bereits in der Warenwirtschaft vorhanden, können sie auch direkt über den Webservice Call *updateDeliveryTimes* gesetzt und per *fetchDeliveryTimes* wieder ausgelesen werden.

Gegebenenfalls ist ein Mapping zwischen ERP- und Speed4Trade CONNECT-Lieferzeiten durchzuführen. Ist dies erfolgt, können die Lieferzeiten beim Übergeben des Artikels über *updateItems-> delivery_time_id / delivery_time_identifier* direkt gesetzt werden.

5.13 Hersteller

Speed4Trade CONNECT Pfad: Einstellungen → Kennzeichnung → Hersteller

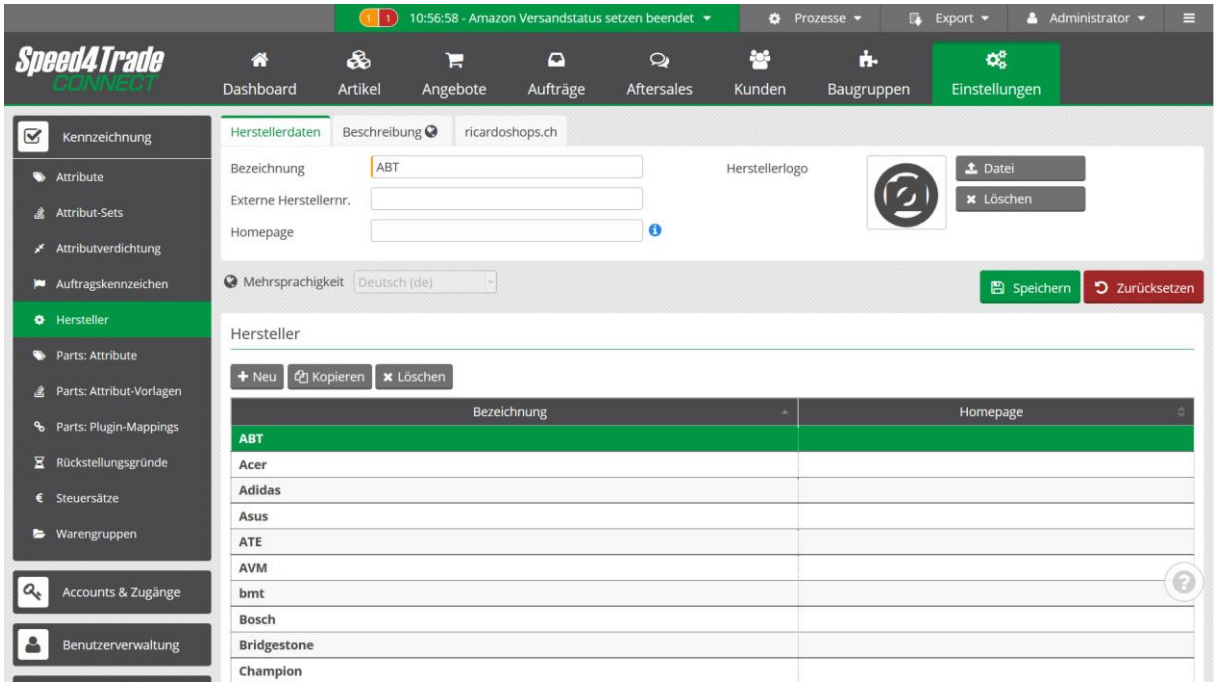

Abbildung 19: Hersteller

5.13.1 Webservice-Call: fetchManufacturers / updateManufacturers (Hersteller)

Standard

Die Angabe eines Herstellers ist in bestimmten Kategorien auf einigen Verkaufskanälen ein Pflichtfeld. Zudem wird der Hersteller oft auch als Such- und Filterkriterium verwendet. Daher erhöht die Pflege eines Herstellers zum Artikel die Verkaufschancen.

Die Hersteller können in Speed4Trade CONNECT angelegt werden. Sind sie jedoch bereits in der Warenwirtschaft gespeichert, macht es Sinn, sie über den Webservice mit dem Call *updateManufacturers* an Speed4Trade CONNECT zu übergeben. In der Folge kann die Hersteller-ID über *fetchManufacturers* aus Speed4Trade CONNECT ausgelesen und per *updateItems* in das Feld *vendor_id* im Artikel wieder gesetzt werden.

5.14 Lager

Speed4Trade CONNECT Pfad: Einstellungen → Versand & Dienstleistung → Läger

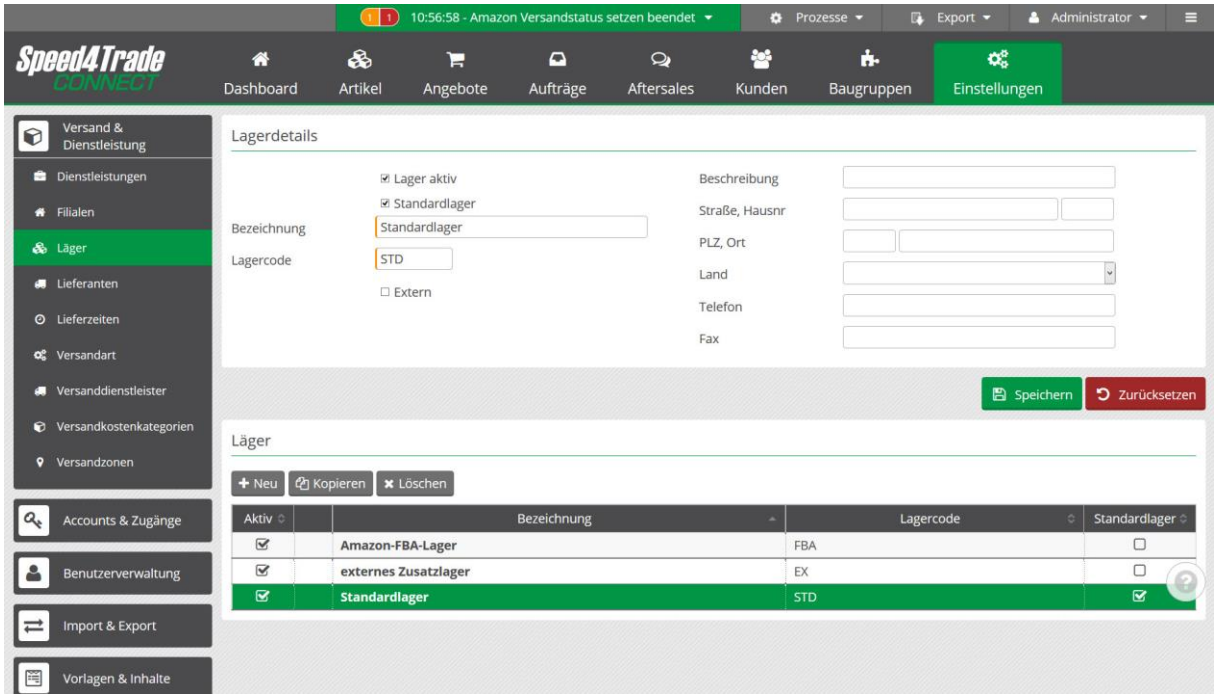

Abbildung 20: Lagerverwaltung

5.14.1 Webservice-Call: fetchWarehouses (Lager)

Standard

Speed4Trade CONNECT ist Mehrlager-fähig. Das heißt, immer wenn Lagerbestände über *updateItem* oder *updateStockCount* zugebucht werden, ist auch das entsprechende Lager im Feld *warehouse_id* mit anzugeben. Daher ist es sinnvoll, die verfügbaren Lager bereits im Vorfeld auszulesen und zwischenzuspeichern.

6 Warengruppen

Die Definition von Warengruppen ist eine sehr effiziente Methode, um große Mengen gleichartiger Artikel komfortabel den Marktplatzkategorien zuzuordnen. Der Aufbau der Warengruppen entspricht im Wesentlichen dem Aufbau der Warengruppen in der Warenwirtschaft. Diese Warengruppen können wie ein Baum beliebig tief strukturiert werden. Da Warengruppen meist ein elementarer Bestandteil der Warenwirtschaft sind, sollten diese auch als Basis für die Warengruppen der Speed4Trade CONNECT dienen.

Die Warengruppen sollten daher durch das externe System in Speed4Trade CONNECT angelegt werden und dabei auch dem Aufbau aus dem externen System entsprechen. Somit findet sich der Anwender auch in Speed4Trade CONNECT leicht in seinem Produktsortiment zurecht.

Jeder Artikel muss immer auch einer Marktplatz- und/oder Shop-Kategorie zugeordnet werden. Dazu werden die Warengruppen aus der Warenwirtschaft in Speed4Trade CONNECT mit den Kategorien der Marktplätze und/oder Shop „gemappt“. Ein Artikel kann dabei immer nur einer Warengruppe zugeordnet sein.

Hinweis!
Die Warengruppen entsprechen noch nicht direkt den Kategorien eines Marktplatzes oder Onlineshops. Die Zuordnung wird erst durch das Mapping innerhalb Speed4Trade CONNECT hergestellt.

Die Warengruppen erfüllen darüber hinaus noch folgende Komfortfunktionen in Speed4Trade CONNECT:

- Filterung von Artikeln, d.h. bei Auswahl einer Warengruppe werden nur die zugeordneten Artikel angezeigt. Somit kann das Artikelsortiment gut strukturiert werden.
- Vererbung von Eigenschaften, d.h. bei einer Warengruppe kann man bspw. Standard-Textbausteine oder Standard-Bildbausteine hinterlegen. Jeder Artikel in dieser Warengruppe bekommt diese Werte automatisch vererbt. Die Vererbung kann bspw. dann verwendet werden, wenn es sich bei der Warengruppe um einen bestimmten Hersteller handelt, für den je Artikel immer ein Herstellerbild nebst Beschreibung definiert werden soll.

Die Verwaltung der Warengruppen ist wie folgt zu erreichen:

Speed4Trade CONNECT Pfad: Einstellungen → Kennzeichnung → Warengruppen

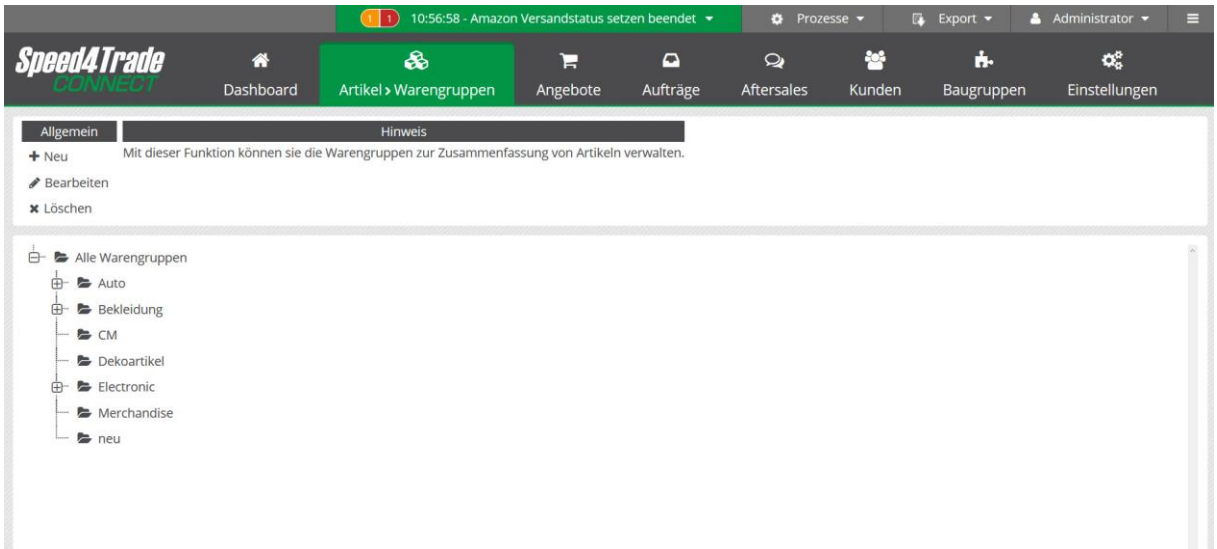

Abbildung 21: Warengruppen

6.1 Webservice-Call: updateCategories / fetchCategories (Warengruppen)

Komfort

Über die Aufrufe können die Warengruppen angelegt (*updateCategories*) und auch wieder ausgelesen (*fetchCategories*) werden. Dabei kann später bei der Artikelübergabe sowohl die Warengruppen-ID (*external_id*) der Warenwirtschaft als auch die interne Speed4Trade CONNECT Warengruppen-ID (*category_id*) verwendet werden.

Benötigte vorbereitende Calls:

- fetchMandators (Mandanten)
- fetchTextModuleTypes (Textbausteintypen)
- fetchImageModuleTypes (Bildbausteintypen)

7 Artikelstammdaten

Im Artikelstamm werden sowohl die Grunddaten, die für alle Verkaufskanäle gleich gelten, als auch die Verkaufskanal-spezifischen Daten gepflegt. Im nachfolgenden Kapitel wird exemplarisch nur auf eBay und Amazon eingegangen. Die anderen Marktplätze und Shops entnehmen Sie bitte der Dokumentation.

Benötigte vorbereitende Calls:

- fetchMandators (Mandanten)
- fetchTextModuleTypes (Textbausteintypen)
- fetchImageModuleTypes (Bildbausteintypen)

Zusätzliche Calls, um weitere Grunddaten vorzubelegen:

- fetchDeliveryChargeCategories (Versandkostenkategorien)
- fetchDeliveryTimes (Lieferzeiten)
- fetchManufactures (Hersteller)

7.1 Artikel Grunddaten

Neben den vorbereitenden Calls sollte bei jedem Artikel auch eine Referenz auf die zuvor angelegte Warengruppe gesetzt werden, um die erwähnten Komfortfunktionen nutzen zu können (siehe Felder *category_id* und *external_category_id* im Call *updateItem*).

Hinweis!
Einige Grundwerte werden automatisiert auch für Verkaufskanäle verwendet. So wird bspw. die interne Artikelnummer zu eBay übergeben, der EAN-Code und die interne Artikelnummer zu Amazon usw.

7.1.1 Grunddaten - Allgemein

Speed4Trade CONNECT Pfad: Artikel → Grunddaten → Allgemein

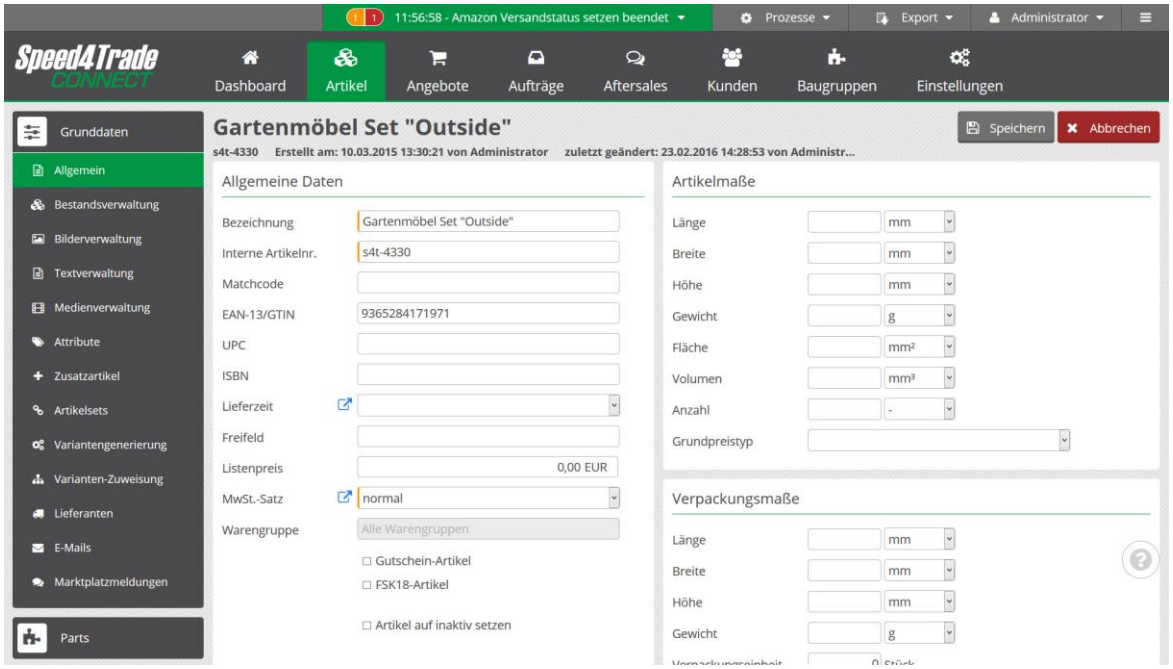

Abbildung 22: Artikel Grunddaten

7.1.2 Grunddaten - Textverwaltung

Speed4Trade CONNECT Pfad: Artikel → Grunddaten → Textverwaltung

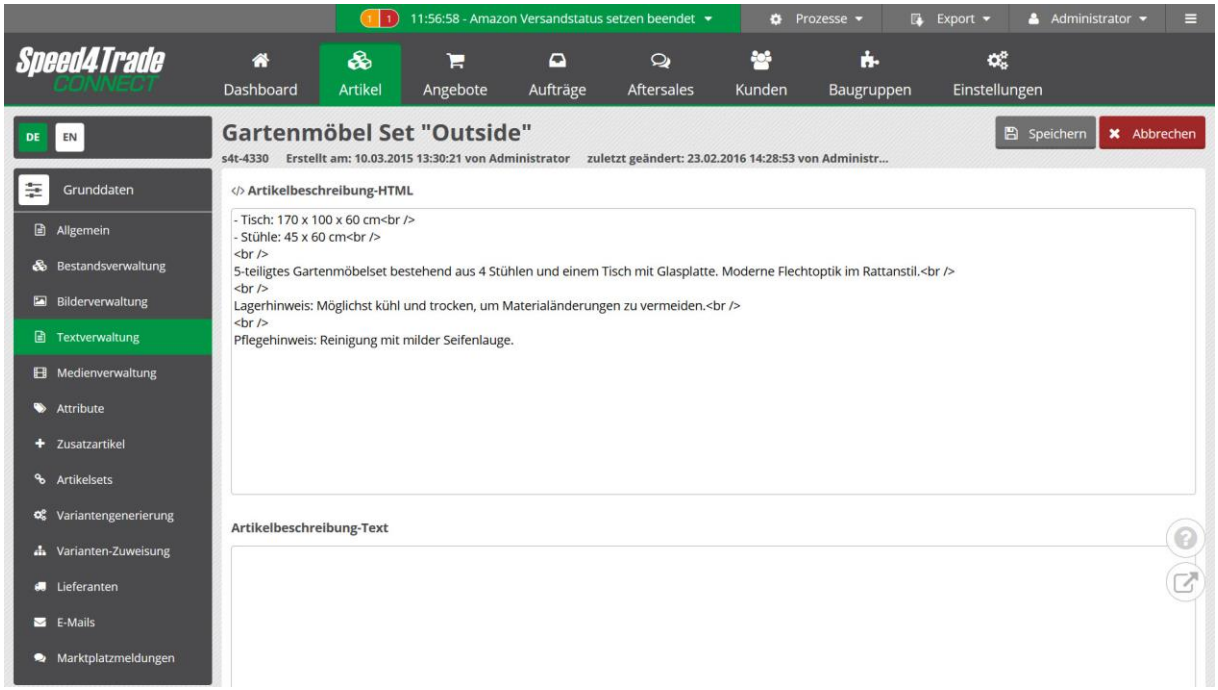

Abbildung 23: Artikel Textverwaltung

7.1.3 Grunddaten - Bilderverwaltung

Je nach Einstellung kann ein lokales Bilder oder eine URL je Bildbausteintyp gesetzt werden.

Speed4Trade CONNECT Pfad: Artikel → Grunddaten → Bilderverwaltung

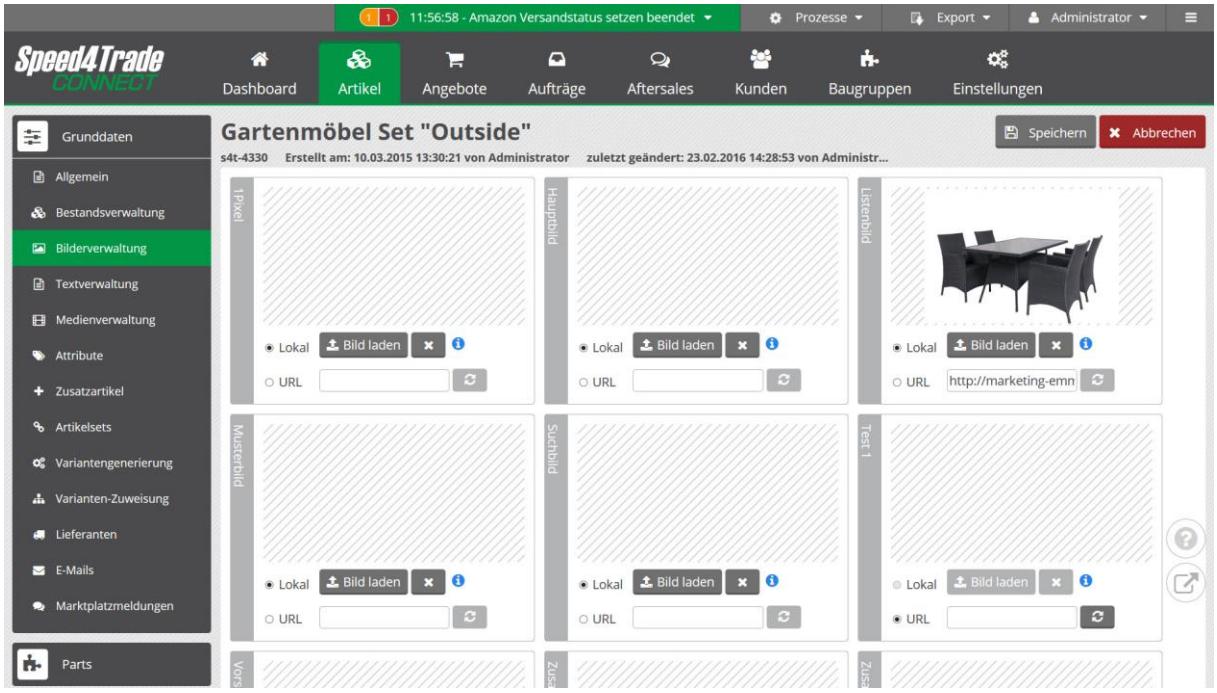

Abbildung 24: Artikel Bildverwaltung

7.2 Webservice-Call: updateItems (Artikelstammdaten)

Standard

Nachfolgend eine Auflistung von grundlegenden Feldern im Artikelstamm, welche über *updateItems* gesetzt werden sollten. Eine komplette Liste finden Sie direkt in der Online-Dokumentation der Webservice-API beim Call *updateItems*. Bitte beachten Sie dort auch die Wertelisten und Hinweise am Ende der Seite beim entsprechenden Funktionsaufruf.

Das Feld "significant_key" gibt an, welches Feld zur Identifikation des Artikels beim Update genutzt wird. Dies kann eine Datensatznummer "item_id" oder eine interne Artikelnummer sein "internal_item_number". Wird keine Nummer angegeben, wird ein neuer Datensatz, sprich Artikel angelegt.

7.2.1 Artikel -> Grunddaten -> Allgemein

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung WebService
Artikelbezeichnung	Pflichtfeld	identifier
interne Artikelnummer	Pflichtfeld	internal_item_number
Matchcode	Bspw. für Suche nach Artikel	match_code
EAN	Wichtig für Amazon, eBay und Google Produktsuche. (es sollte grundsätzlich keinen Artikel ohne EAN-Nummer geben)	ean_code
Hersteller-Artikelnummer	Artikelnummer des Herstellers	vendor_item_number
Hersteller	Hersteller ID	vendor_id (Hersteller ist bereits in Speed4Trade CONNECT angelegt, ID wurde via Webservice-Call ermittelt = *fetchManufacturers*) external_vendor_id (Hersteller wurde über WaWi neu in Speed4Trade CONNECT angelegt = *updateManufacturers*)
Lieferzeit	Wichtig für Shop und Amazon (Pflege kann sowohl in der Warenwirtschaft als auch in Speed4Trade CONNECT erfolgen)	delivery_time_id (Lieferzeit wurde in Speed4Trade CONNECT angelegt, ID wurde via WebService-Call ermittelt = *fetchDeliveryTimes*) delivery_time_identifier (Lieferzeit wird über WaWi neu in Speed4Trade CONNECT angelegt)
Freifeld	Internes Filterkriterium	variable_field
Listenpreis	Listenpreis des Artikels Dieser Preis dient ausschließlich zur Info und findet keine „funktionale“ Anwendung	list_price

MwSt.-Satz	Pflichtfeld:
	Übergabe der jeweiligen MwSt.-Satz-ID aus der Warenwirtschaft
	• Steuersätze:
	  ○ normal
	  ○ steuerfrei
	  ○ vermindert
		tax_type_id
		Feste Werte:
		1 = Frei
		2 = Reduziert
		3 = Normal

Warengruppe	Kann beim Anlegen des Artikels gesetzt werden, sonst keine Zuordnung	category_id (Kategorie wurde in Speed4Trade CONNECT angelegt, ID wurde via Webservice-Call ermittelt = *fetchCategories*) external_category_id (Kategorie wurde durch WaWi in Speed4Trade CONNECT neu angelegt = *updateCategories*)
FSK 18-Artikel	Kann gesetzt werden, wenn FSK-Artikel verkauft werden. Relevant für Shops	is_adult_product
Artikel auf inaktiv setzen	Hinweis: Das Setzen dieses Kennzeichens hat nur Auswirkungen bei Shops. Es werden dadurch Angebote nicht automatisch beendet.	active
Artikelmaße	Sollte bei entsprechenden Artikeln gesetzt werden, da viele Kunden danach fragen • Länge / Einheit • Breite / Einheit • Höhe / Einheit • Gewicht / Einheit • Fläche / Einheit • Volumen / Einheit	length / distance_unit_id width / distance_unit_id height / distance_unit_id weight / weight_unit_id area / area_unit_id volume / volume_unit_id
Grundpreistyp	Grundpreistyp des Artikels. Relevant für grundpreispflichtige Artikel (z.B. Grundpreis für 100 ml bei Getränken)	base_price_type_id Bei Angabe des Grundpreis-typs muss der dazugehörige Maßeinheitenwert gesetzt sein.
Verpackungsmaße	Sollte bei entsprechenden Artikeln gesetzt werden. • Länge / Einheit • Breite / Einheit • Höhe / Einheit • Gewicht / Einheit	package_length / package_distance_unit_id package_width / package_distance_unit_id package_height / package_distance_unit_id package_weight / package_weight_unit_id
Versandkostenkategorien	Versandkostenkategorien sind für eBay wichtig. Die zugehörigen Versandkosten können in Speed4Trade CONNECT gepflegt werden. Die	delivery_charge_category_id

	Versandkostenkategorien (ID) ist dann über die Warenwirtschaft zu setzen.	

7.2.2 Artikel -> Grunddaten -> Zusatzartikel

Erweitert

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Zusatzartikel	Derzeit nur für Shops und Yatego möglich, an Amazon wird eine Empfehlung übergeben.	cross_selling_item

7.2.3 Artikel -> Grunddaten -> Artikelset

Erweitert

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Artikelset	Speed4Trade CONNECT kann mit Set-Artikeln/Bundles umgehen. Es ist im Vorfeld immer zu klären, ob Speed4Trade CONNECT „wissen“ muss, ob es sich um einen Set-Artikel handelt, oder ob nur ein einziger Artikel übergeben wird und die Warenwirtschaft alles, bspw. Bestandsverwaltung, für Set-Artikel / Bundles / Stücklisten übernimmt.	item_set.

7.2.4 Artikel-Varianten

Die Speed4Trade CONNECT beherrscht auch den Umgang mit Varianten. Dabei sind die für die Varianten notwendigen Eigenschaften (Farbe, Größe, etc.) und zugehörigen Werte (z.B. rot, blau, bzw. M, L, XL etc.) zu definieren.

Erweitert

Die Erzeugung von Varianten kann direkt auch über den Call *updateItems* unter dem Wrapper-Element *variant* erfolgen. Neben den verkaufbaren Variantenartikeln (z.B. T-Shirt in Farbe Blau und Größe L) auch SKU (Stock Keeping Unit) genannt, muss auch immer ein (nicht kaufbarer) Hauptartikel generiert werden, der die Varianten logisch gruppiert bzw. zusammenfasst. Der Hauptartikel kann als solcher mit dem Feld *variant_standard* gekennzeichnet werden. Zuerst muss der Hauptartikel in einem ersten *updateItems-Call* übergeben werden. In einem zweiten *updateItems-Call* können dann die Varianten zugewiesen werden.

Zusätzlich wird beim *updateItems-Call* geprüft, ob die Eigenschaften/Werte Kombination bereits in den Stammdaten vorhanden ist. Ist dies nicht der Fall, wird die Kombination neu angelegt.

7.2.5 Artikel-Attribute (Zentralisierte Attributverwaltung - ZAV)

Zuweisung von zentralen Attributwerten zum Artikel über den Call updateItems unter dem Wrapper-Element *item_attributes*.

Hinweise:
Eine Ausnahme sind die variantenbildenden Attribute, die weiterhin über variant/attributes zugewiesen werden. Eine Neuanlage von Attributen und Attributwerten inkl. Marktplatzdaten ist über diesen Call auch möglich, aber keinesfalls Best Practice und nicht für große Datenmengen geeignet. Das sollte über *updateAttribute*s erfolgen.

7.3 Webservice-Call: fetchItems

Nachdem nun ausführlich erläutert wurde, wie die Artikelstammdaten übergeben werden können, soll noch kurz erwähnt werden, dass über den Call fetchItems die Artikeldaten abgeholt werden können. Dies ist vor allem nützlich, wenn man an eine bestehende Speed4Trade-CONNECT-Installation anbindet. Dabei müssen initial alle vorhandenen Artikelstammdaten in die Warenwirtschaft importiert werden. Dieser Reimport funktioniert einerseits über einen CSV-Export der Daten aus Speed4Trade CONNECT oder über den Call fetchItems.

8 Artikel - Verkaufskanal-spezifische Daten

Jeder Verkaufskanal bietet unterschiedliche Einstellungen, die sich wiederum auch von der Komplexität unterscheiden. Es sollten über das externe System so viele Daten wie möglich gesetzt werden, um den Pflegeaufwand in Speed4Trade CONNECT so gering wie möglich zu halten. Das Setzen von Verkaufskanal-spezifischen Daten zählt zur 'Erweiterten' Implementierung. Davon ausgenommen ist das Übergeben von Artikelbezeichnung und Preis je Verkaufskanal.

8.1 Artikelbezeichnung und Preis

Standard

Warum ist das Setzen von Artikelbezeichnung und Preis je Verkaufskanal über das ERP-System so wichtig? Der Verkaufspreis und die Bezeichnung sind die wichtigsten Felder je Verkaufskanal. Dadurch unterscheiden sich die Artikel am meisten. Vor allem der Preis sollte zentral im führenden System, sprich der Warenwirtschaft gepflegt und verwaltet werden. Zudem können diese beiden Felder nicht über die Speed4Trade CONNECT Komfortfunktionen Templates, Vererbung und Mehrfacheditieren gesetzt werden. Das heißt: Können Artikelpreis und -bezeichnung bereits über das externe System gesetzt werden, ist das eine erhebliche Arbeitserleichterung für den Händler.

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Artikelbezeichnung	Pflichtfeld	title
Standardpreis	Standardpreis für Amazon	price[2]
Startpreis	Startpreis für eBay	minimal_bid
Sofort-Kaufen-Preis	Sofort-Kaufen-Preis für eBay	buy_it_now_price

[2] Besser: Preislisten verwenden!

8.2 Marktplatz Amazon

Speed4Trade CONNECT Pfad: Artikel → Amazon → Allgemein

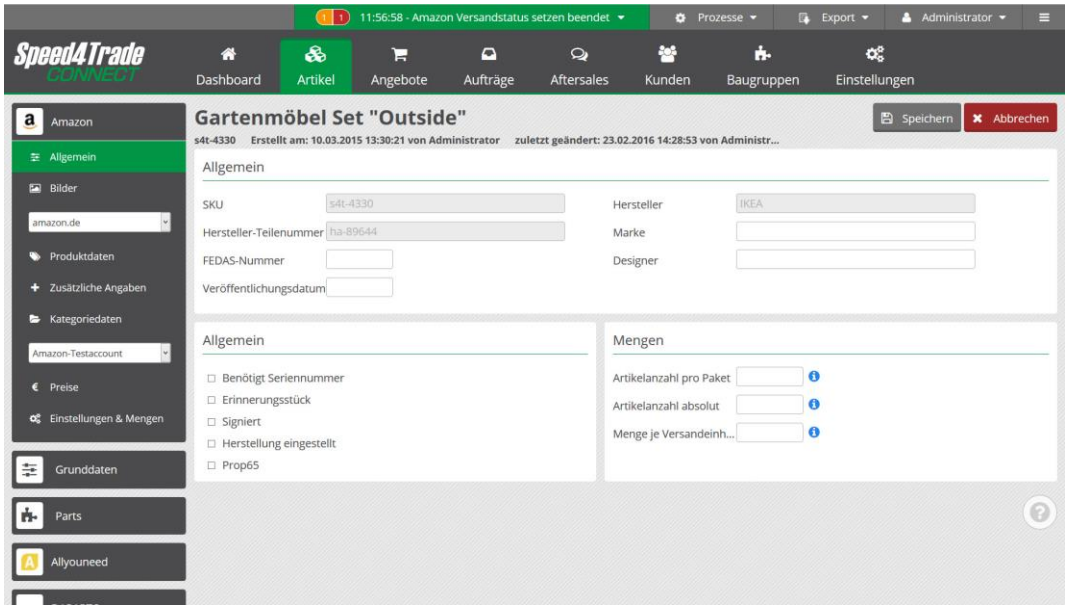

Abbildung 25: Artikeldaten - Amazon

Bei Amazon gibt es Daten, die allgemein für alle Amazon Länderplattformen gelten. Dazu zählt bspw. der Hersteller oder die SKU=interne Artikelnummer. Weiterhin gibt es Einstellungen, die nur für die jeweilige nationale Amazon Plattform – in diesem Fall z.B. für amazon.de - gesetzt werden können. Abschließend gibt es noch Werte, die je Amazon-Account gesetzt werden können, wie bspw. der Preis.

Benötigte vorbereitende Calls:

- fetchSubChannels (Sub-Channels)
- fetchAccounts (Accounts)

8.2.1 Webservice-Call: updateItems (Amazon-spezifische Daten)

Erweitert

Im Vergleich zu anderen Online-Marktplätzen gibt es bei Amazon gewisse Eigenheiten. Bei Amazon ist es bspw. so, dass ein sogenannter Produktkatalog zur Verfügung gestellt wird. Wenn der Artikel, welcher bei Amazon eingestellt werden soll, bereits durch einen anderen Verkäufer gelistet wurde, so dass bereits die Artikelinformationen im Produktkatalog vorhanden sind, dann müssen in der Speed4Trade CONNECT nur noch folgende Daten gepflegt werden.

- Hersteller
- Artikelbezeichnung
- Bestand
- Preis (besser: Preislisten verwenden!)

- Zustand
- EAN / ASIN-Nummer
- Lieferzeit

Ist der Artikel, welcher eingestellt werden soll, noch nie zuvor bei Amazon gelistet worden, müssen neben den oben aufgeführten Punkten zusätzliche Daten gepflegt werden.

8.2.1.1 Artikel -> Amazon -> Allgemein

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
SKU	wird von den Artikelgrunddaten übernommen (interne Artikelnummer)	internal_item_number
Hersteller	wird von den Artikelgrunddaten übernommen	vendor_id
Hersteller-Teilenummer	wird von den Artikelgrunddaten übernommen	vendor_item_number

Im Bereich "Amazon -> Allgemein" müssen keine Daten durch die Warenwirtschaft gesetzt werden, da diese von den Grunddaten übernommen werden. Für die SKU-Nummer wird die interne Artikelnummer verwendet.

8.2.1.2 Artikel -> Amazon -> Bilder

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Hauptbild	Bildbaustein-ID	main_image (optional)
Musterbild	Bildbaustein-ID	swatch_image
Suchbild	Bildbaustein-ID	search_image
Zusatzbild 1	Bildbaustein-ID	pt1_image
Zusatzbild 2	Bildbaustein-ID	pt2_image
Zusatzbild 3	Bildbaustein-ID	pt3_image
Zusatzbild 4	Bildbaustein-ID	pt4_image
Zusatzbild 5	Bildbaustein-ID	pt5_image
Zusatzbild 6	Bildbaustein-ID	pt6_image
Zusatzbild 7	Bildbaustein-ID	pt7_image
Zusatzbild 8	Bildbaustein-ID	pt8_image

Die Bilder, welche für den Marktplatz Amazon verwendet werden sollen, müssen u.a. folgende Voraussetzungen erfüllen:

- freigestellte Bilder → ohne Hintergrund
- Auflösung mind. 1024 x 1024px
- dürfen kein Wasserzeichen enthalten

Hinweis!
Wenn nur Artikel bei Amazon eingestellt werden, welche bereits im Produktkatalog vorhanden sind, ist das Hinterlegen der Bildbausteine im Reiter "Amazon" nicht zwingend erforderlich.

8.2.1.3 Artikel -> Amazon -> Produktdaten

Länderspezifische Daten zum Beispiel für Amazon.de setzen.

Benötigte vorbereitende Calls:

- fetchTemplates (Angebotsvorlagen)

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Artikelbezeichnung	Pflichtfeld	title
Aufzählungspunkt 1	wenn möglich da verkaufsfördernd	bulletpoint1
Aufzählungspunkt 2	wenn möglich da verkaufsfördernd	bulletpoint2
Aufzählungspunkt 3	wenn möglich da verkaufsfördernd	bulletpoint3
Aufzählungspunkt 4	wenn möglich da verkaufsfördernd	bulletpoint4
Aufzählungspunkt 5	wenn möglich da verkaufsfördernd	bulletpoint5
Beschreibung	Zuweisung einer Angebotsvorlage. Kann in Speed4Trade CONNECT via Mehrfacheditieren erfolgen, oder man holt die Angebotsvorlagen vorab via *fetchTemplates* ab.	description_template_id
Standard-Produktnummer	Pflichtfeld EAN-Nummer wird von den Artikelgrunddaten übernommen ASIN-Nummer könnte – sofern bekannt – gesetzt werden.	standard_product_id
Standard-Produktnummer-Typ	Legt fest welche Produktnummer verwendet wird (EAN, ASIN, ISBN, etc.)	standard_product_id_type

Für das Einstellen auf fremdsprachige Verkaufskanäle sind zusätzlich die Artikelbezeichnungen und Beschreibungen mehrsprachig zu pflegen.

Die Pflege der Aufzählungspunkte ist nur relevant, wenn der Amazon-Account berechtigt ist, Informationen im Produktkatalog neu anzulegen bzw. zu überschreiben. Das Gleiche gilt für die Pflege der "Zusätzlichen Angaben" in den Amazon Markplatz-spezifischen Artikeldaten.

Die Pflege der ASIN- und EAN-Nummer erfolgt normalerweise in der Warenwirtschaft. In Abhängigkeit welches "Flag" (standard_product_id_type) - EAN oder ASIN - in der Warenwirtschaft gesetzt wurde, wird der jeweilige Wert an die Speed4Trade CONNECT übergeben.

8.2.1.4 Artikel -> Amazon -> Kategoriedaten

Länderspezifische Daten zum Beispiel für Amazon.de setzen.

Die sogenannten *"RecommendedBrowseNotes"* können entweder bei jedem Artikel einzeln oder je Warengruppe gesetzt werden.

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
RecommendedBrowseNotes 1	Gibt den vorgeschlagenen Suchknoten 1 des Artikels an.	recommended_browse_node1
RecommendedBrowseNotes 2	Gibt den vorgeschlagenen Suchknoten 2 des Artikels an.	recommended_browse_node2

Artikel -> Amazon -> Preise

Der Preis eines Artikels kann für jeden Verkaufsaccount, welcher in der Speed4Trade CONNECT hinterlegt ist, einzeln gesetzt werden (Account-spezifische Daten). Inzwischen ist es die bessere Alternative den Standardpreis mittels der zentralen Preislisten zu übergeben.

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Standardpreis	Standardpreis für Amazon	price
Sonderangebot /Startdatum	Startdatum für Sonderangebot	sale_start_date
Sonderangebot /Endedatum	Endedatum für Sonderangebot	sale_end_date
Sonderangebot /Preis	Preis des Sonderangebotes	sale_price

Artikel -> Amazon -> Einstellungen & Menge

<u>Hinweis zu Fulfillment by Amazon:</u>
Der Wert "AFN" bedeutet, dass der bestellte Artikel über ein FBA-Lager verschickt wird. Wurde der Wert "MFN" gesetzt, dann wird der Artikel über ein anderes / eigenes Lager an Käufer verschickt.

Derzeit kann über die Speed4Trade CONNECT der gleiche Artikel unter einem Account nur einmal bei Amazon eingestellt werden.

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Startdatum	Hierüber kann ein Datum definiert werden, ab wann ein Artikel verfügbar ist.	launch_date
Vorlaufzeit der Lieferung	Wird von den Artikelgrunddaten übernommen (Lieferzeit)	fulfillment_latency
Switchfulfillment to	- der Wert "AFN" muss gesetzt werden, wenn ein Bestand auf dem FBA-Lager vorhanden ist	switch_fulfillment_to

	- der Wert "MFN" muss gesetzt werden, wenn kein Bestand auf dem FBA-Lager mehr vorhanden ist und der Artikel über ein anderes Lager verschickt werden muss.	
Zustand	Zustand eines Artikels	condition_type
Zustandsbeschreibung	Genauere Beschreibung des Zustandes	condition_type_note

8.3 Marktplatz eBay

Speed4Trade CONNECT Pfad: Artikel → eBay → Kategorien

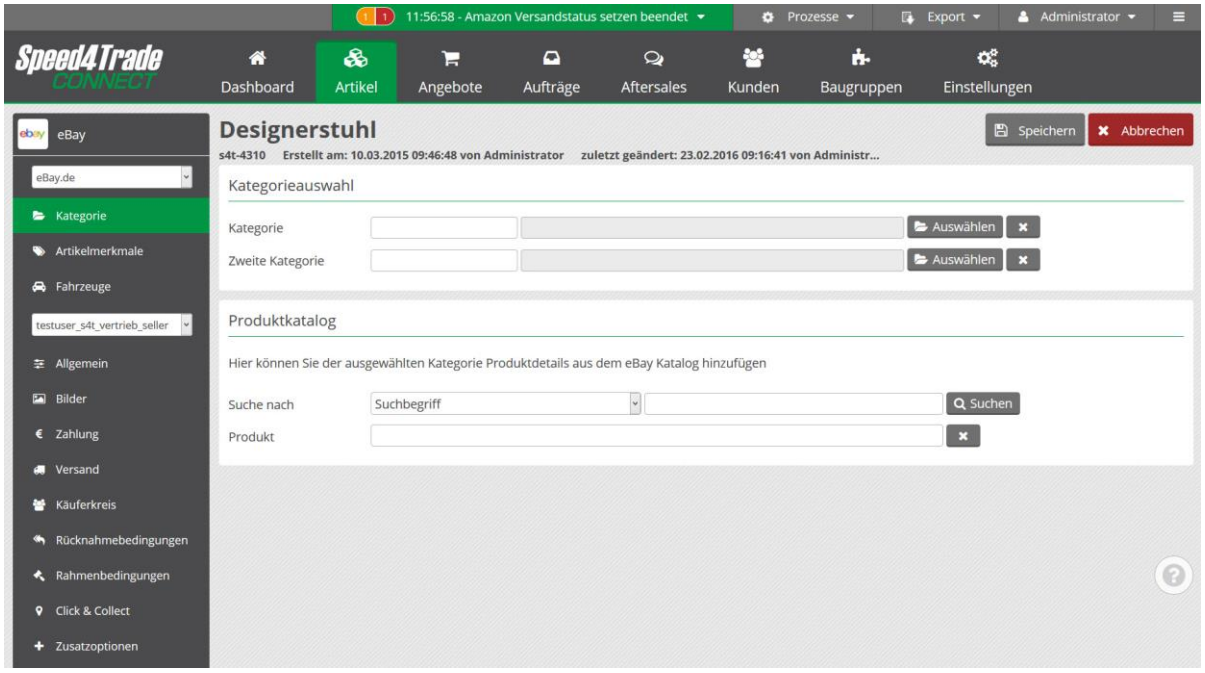

Abbildung 26: eBay Kategorien

Ähnlich wie bei Amazon gibt es auch bei eBay Einstellungen, die Account-unabhängig für den Artikel je eBay-Plattform gelten. Dazu zählt bspw. die eBay-Kategorie.

Alle weiteren Einstellungen können je Account individuell gesetzt werden, wie bspw. der eBay-Titel oder der Preis.

Benötigte vorbereitende Calls:

- fetchSubChannels (Sub-Channels)
- fetchAccounts (Accounts)

8.3.1 Webservice-Call: updateItems (eBay-spezifische Daten)

Erweitert

Artikel -> eBay -> Kategorien

Länderspezifische Daten zum Beispiel für eBay.de setzen.

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Kategorie	Pflichtfeld	ebay_category_id_1
Zweite Kategorie	Achtung! Verursachen doppelte Einstellgebühren bei eBay, falls gesetzt	ebay_category_id_2

Wenn in der Speed4Trade CONNECT eBay-Accounts von verschiedenen eBay-Ländern hinterlegt sind, dann muss je eBay-Land auch eine Kategorie gesetzt werden (in Speed4Trade CONNECT, via Webservice).

Sofern in der Warenwirtschaft der eBay-Kategoriebaum dargestellt werden soll, kann Speed4Trade eine Datei mit den Daten zur Verfügung stellen.

Artikel -> eBay -> Attribute

Länder-spezifische Daten zum Beispiel für eBay.de setzen.

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Artikelzustand	Pflichtfeld	item_condition

In Abhängigkeit, welche eBay-Kategorie ausgewählt bzw. hinterlegt wurde, werden Kategorie-spezifische Attribute angezeigt. Diese können auch über die Warenwirtschaft gesetzt werden. Allerdings wird in einem einfachen Integrationsmodell davon abgeraten, da die Attribute sehr dynamisch sind und darauf geachtet werden muss, dass die Werte genau so wie bei eBay geschrieben werden müssen. Werden die Attribute anders geschrieben, dann werden neue angelegt. Allerdings werden diese nicht von eBay in der Suche berücksichtigt. Aus diesem Grund erfolgt die Pflege der eBay-Attribute in der Regel in der Speed4Trade CONNECT.

Artikel -> eBay -> Allgemein

Nachfolgende Einstellungen werden beim jeweiligen eBay-Account hinterlegt und an die entsprechenden Artikel vererbt. Diese müssen nicht durch die Warenwirtschaft gesetzt werden.

- Bildbausteine
- Zahlungsoptionen
- Versandkosten(-kategorien)
- Käuferkreis
- Zusatzoptionen

Benötigte vorbereitende Calls:

- fetchAccountsCategories (Kategorien)
- fetchTemplates (Angebotsvorlagen)

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Artikelbezeichnung	Pflichtfeld (max. 80-Zeichen)	title
Untertitel	Achtung! Kostenpflichtig, kann u.a. auch für Grundpreisangaben verwendet werden.	sub_title
Shop-Kategorie	eBay-Shop-Kategorie, die jeder eBay-Verkäufer selbst definieren kann. Setzen via Webservice oder manuell in Speed4Trade CONNECT beim Artikel oder Mehrfacheditieren. Alternativ: Pflege über Warengruppen → Vererbung	shop_category_id_1 (*fetchAccountCategories*)
Zweite Shop-Kategorie	eBay-Shop-Kategorie, die jeder eBay-Verkäufer selbst definieren kann. Setzen via Webservice oder manuell in Speed4Trade CONNECT beim Artikel oder Mehrfacheditieren. Alternativ: Pflege über Warengruppen → Vererbung	shop_category_id_2 (*fetchAccountCategories*)
Startpreis	Startpreis	minimal_bid
Sofort-Kaufen-Preis	Sofort-Kaufen-Preis	buy_it_now_price[3]
Preisvorschlag	Preisvorschlag generell akzeptieren	best_offer_enabled
Preisvorschlag akzeptieren	Preisvorschlag akzeptieren ab	best_offer_accept_threshold
Preisvorschlag ablehnen	Preisvorschlag ablehnen unter	best_offer_reject_threshold
Angebotsvorlage für Höchstgebot	Zuweisung einer Angebotsvorlage. Kann in Speed4Trade CONNECT via Mehrfacheditieren erfolgen, oder man holt die Angebotsvorlagen vorab via *fetchTemplates* ab. Alternativ: Pflege über Warengruppen → Vererbung	auction_template_id
Angebotsvorlage für Festpreis	Zuweisung einer Angebotsvorlage. Kann in Speed4Trade CONNECT via Mehrfacheditieren erfolgen, oder man holt die Angebotsvorlagen vorab via *fetchTemplates* ab. Alternativ: Pflege über Warengruppen → Vererbung	template_id_bin

[3] Bevorzugt nun über Preislisten umzusetzen, statt einem kanalspezifischen Preis.

eBay Plus Kennzeichen	Absichtserklärung, dass dieser Artikel ein eBay Plus Artikel ist	ebay_plus

9 Übersicht über marktplatzspezifische Eigenheiten

	eBay	Amazon	Yatego	Rakuten
URL-Bilder	✓	✓	✓	✓
Lokale-Bilder	✓	-	-	-
Länge der Artikelbezeichnung	80 Zeichen	200 Zeichen *[4]	255 Zeichen	100 Zeichen
Länge der Artikelbeschreibung	65536	2000 Zeichen	*[5]	Keine Beschränkung
HTML-Text	Keine Einschränkung	fett, kursiv, unterstreichen, Aufzählung	fett, kursiv, unterstreichen, Aufzählung	fett, kursiv, unter-streichen, Aufzählung
Plain-Text	✓	✓	✓	✓
Versandkosten-verwaltung	in Speed4Trade CONNECT	auf Marktplatz	auf Marktplatz	auf Marktplatz
Template	✓	✓	✓	✓
Preis	Festpreis Startpreis	Standardpreis Sonderpreis	Bruttopreis UVP	Listenpreis Sonderpreis
Shop	✓	Nur für Händler mit Platinum-Shop Berechtigung	✓	✓
Darstellung von Varianten	✓	✓	✓	✓

*[4] durch Amazon von 500 auf 200 am 15.7.2015 reduziert
*[5] Die Zeichenanzahl für die Kurzbeschreibung bei Yatego beträgt 130 Zeichen. Bei der Langbeschreibung gibt es keine Zeichenbegrenzung.

10 Bestandsverwaltung

Bestände müssen regelmäßig aus der Warenwirtschaft und möglichst aktuell an die Speed4Trade CONNECT übergeben werden. In der Artikelverwaltung gibt es einen eigenen Bereich für die Bestandsverwaltung.

Speed4Trade CONNECT Pfad: Artikel → Grunddaten → Bestandsverwaltung

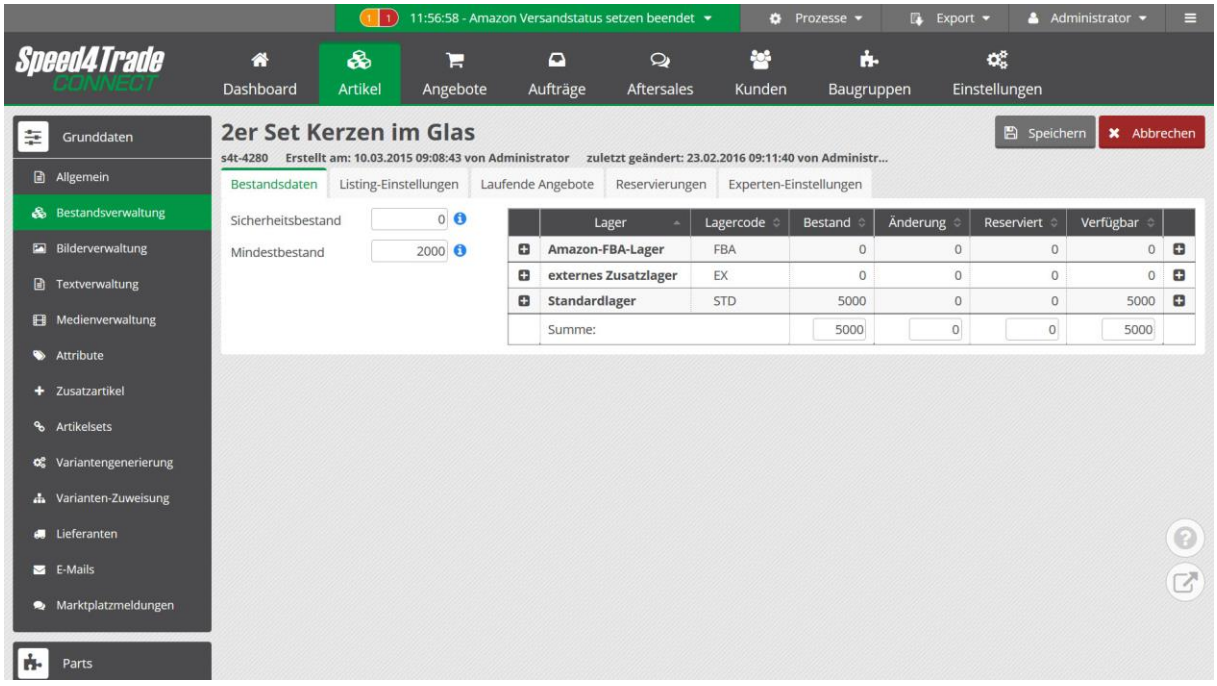

Abbildung 27: Bestandsverwaltung

Es gibt dazu 3 wichtige Bestandsfelder, die gesetzt werden müssen:

- Lagerbestand (stock_count)
- Sicherheitsbestand (safety_stock_count)
- Mindestbestand (minimal_stock-count)

Und optional noch 2 Mengenfelder für das Einstellen der Angebot, die gesetzt werden können:

- Einstellmenge (placing_count)
- Auffüllmenge (refill_count)

Die Speed4Trade CONNECT verfügt dabei über einen sogenannten „interaktiven Bestandsabgleich“.

<u>Dies bedeutet im Detail:</u>
Bei den „Listing-Einstellungen“ in der Bestandsverwaltung legt man einen primären Verkaufsaccount fest. Weiterhin definiert man je Account, mit welcher Menge der Artikel in den jeweiligen Verkaufskanal eingestellt und ab welcher Menge der Bestand wieder aufgefüllt werden soll.

<u>Beispiel:</u>

Einstellmenge 10, Auffüllmenge 8: Sinkt der Bestand im Verkaufskanal unter 8, dann wird die Menge wieder auf 10 erhöht. Diese Systematik ist insbesondere bei eBay relevant, um den sogenannten „SalesRank“ zu bilden.

Sinkt der eigentliche Lagerbestand unter den Sicherheitsbestand, dann werden alle Angebote bis auf den primären Verkaufsaccount beendet. Sinkt der Lagerbestand auch noch unter den Mindestbestand, dann werden auch die Angebote des primären Verkaufsaccounts beendet.

Durch den interaktiven Bestandsabgleich ist es auch möglich, einen Lagerbestand gleichmäßig über verschiedene Accounts zu verkaufen.

Beispiel:
Lagerbestand: 5, Sicherheitsbestand: 0, Mindestbestand: 0
Der Artikel wird sowohl in Amazon als auch in eBay und dem Online-Shop mit der Menge 5 eingestellt. Wird bspw. ein Artikel bei eBay verkauft, dann wird die Menge in Amazon und dem Online-Shop auf 4 reduziert.

Hinweis:
Wird für die Einstellmenge der Wert -1 übergeben, dann ist der interaktive Bestandsabgleich für dieses Angebot deaktiviert.

10.1 Bestandsübergabe per Artikelstammdaten

Grundsätzlich gibt es zwei Varianten, Bestände in der Speed4Trade CONNECT über den Webservice zu setzen. Eine Möglichkeit ist, dass die Bestände über den Webservice-Call "*updateItems*" gesetzt werden, welcher auch für das Setzen der Artikelstammdaten verwendet wird. Die zweite Möglichkeit besteht darin, dass der Bestand über den Webservice-Call "*updateStockCount*" gesetzt wird. Wir empfehlen immer den Bestand über den Webservice-Call "*updateStockCount*" zu setzen, da dies wesentlich performanter ist.

Hinweise zur Bestandsübergabe:

1. Die Bestandsübergabe sollte – wie erwähnt – regelmäßig und möglichst zeitnah zum Ereignis der Bestandsänderung im externen System geschehen, damit die Speed4Trade CONNECT die Bestände in den Verkaufskanälen gegebenenfalls anpassen kann, damit bspw. keine Überverkäufe auftreten.
2. Die „Listing-Einstellungen“ können ebenfalls über den Webservice gesetzt werden. Jedoch ist es auch möglich, dass man diese über eine Mehrfacheditierfunktion in Speed4Trade CONNECT setzt.
3. Einstell- und Auffüllmengen können auch über die Verkaufsaccount-Konfiguration vorbelegt werden. Diese Vorbelegung wird allerdings nur bei Neuanlage eines Artikels initial übernommen.

10.1.1 Webservice-Call: updateItems (Artikelbestand)

Standard

Bei der Verwendung von "*updateItems*" müssen nicht zwangsläufig immer alle Felder übergeben werden, d.h. beim Setzen der Artikelstammdaten übergibt man alle möglichen Werte aus dem externen System, beim Setzen des Bestandes nur den entsprechenden Wert aus dem externen System. Somit kann man für die Bestandsaktualisierung mit einem relativ „schlanken“ Call arbeiten.

Mögliche Ereignisse für die Bestandsübergabe:

- Wareneingang
- Telefonische/manuelle Aufträge im externen System
- Retouren/Rückläufer
- …

Benötigter vorbereitender Call:

- fetchWarehouses (Lager)

10.1.2 Artikel -> Grunddaten -> Bestandsverwaltung

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Lager	Lager, für das der Bestand gesetzt werden soll	Warehouse_id (Abfrage über *fetchWarehouses*)
Lagerbestand	Absolute Menge an Lagerbestand zum Artikel.	stock_count
Sicherheitsbestand	Wert, ab dem Angebote bei den Marktplätzen beendet werden soll, bis auf primärer Verkaufsaccount	safety_ stock_count
Mindestbestand	Wert, ab dem Angebote bei den Marktplätzen beendet werden soll, gilt auch für primären Verkaufsaccount	minimal_ stock_count
Einstellmenge	Menge, mit der der Artikel auf einem Marktplatz gelistet werden soll	placing_count
Auffüllmenge	Menge, ab der ein Wiederauffüllen stattfinden soll	refill_count
Primärer Verkaufsaccount	Darf je Artikel nur einmal gesetzt werden	primary

10.2 Bestandsübergabe per separater Bestandsaktualisierung

Benötigter vorbereitender Call:

- fetchWarehouses (Lager)

10.2.1 Webservice-Calls: updateStockCount (Artikelbestände)

Standard

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Artikelbezeichnung	Pflichtfeld	identifier
interne Artikelnummer	Pflichtfeld	internal_item_number
Lagerbestand	Absolute Menge an Lagerbestand zum Artikel.	stock_count
Lager	Lager, für das der Bestand gesetzt werden soll	Warehouse_id (Abfrage über *fetchWarehouses*)

11 Auftragsverwaltung

Die Funktionen der Auftragsbearbeitung wie bspw. Zahlungsabgleich, Rechnungsstellung, Mahnwesen findet in der Regel zentral in der Warenwirtschaft statt. Über die Speed4Trade CONNECT werden die Aufträge regelmäßig automatisch von den Marktplätzen abgeholt. Diese stehen dann unmittelbar der Warenwirtschaft zur Abholung per Speed4Trade CONNECT API Webservice Schnittstelle zur Verfügung. Damit die Aufträge im ERP-System entsprechend differenziert werden können, sollte ein getrennter Auftragsnummernkreis in Speed4Trade CONNECT verwendet werden. Der Marktplatz eBay kann z.B. das Präfix "eBy" erhalten, Amazon z.B. "Ama" und der Shop das Präfix "Shp". Anhand dieser Unterscheidungen können entsprechende Auswertungen in der Warenwirtschaft durchgeführt werden. Nachfolgend werden zwei mögliche Auftragsworkflows näher erläutert.

11.1 Auftragsworkflow ohne externes System

Innerhalb von Speed4Trade CONNECT gibt es einen festgelegten Ablauf für den Abverkauf von Artikeln, der sich auch in der Hauptnavigation widerspiegelt.

Abbildung 28: Navigationsleiste

Ausgangspunkt von allem sind Artikel. Aus den Artikeln werden „Vorbereitete Angebote“. Nach dem Upload in die Verkaufskanäle werden diese zu „Laufenden Angeboten“. Gibt es keinen Käufer zu einem Angebot bzw. wird ein laufendes Angebot beendet, dann findet man dieses unter „Beendete Angebote“.

Abbildung 29: Angebotsworkflow

Im Falle eines Verkaufs wird der Auftrag automatisiert aus dem jeweiligen Verkaufskanal abgeholt und es befindet sich ein neuer Eintrag unter „Aufträge“.

Abbildung 30: Auftragsworkflow

Für jeden Auftrag unter „Neue Aufträge“ wird eine Erstbenachrichtigung verschickt. Anschließend wird der Auftrag im Standardfall in den Status „Offene Zahlungen“ verschoben. Nach Zahleingang (CSV, PayPal) wird der Auftrag an den Status „Versand“ weitergereicht. Von hier aus sind der Export zu Versanddienstleistern, der Ausdruck einer Rechnung und insbesondere der Abschluss des Auftrags möglich.

11.2 Auftragsworkflow mit externem System (ERP/WWS)

Der Workflow bis zur Auftragsanlage bleibt auch bei Anbindung eines externen Systems identisch. Je nach Leistungsumfang des externen Systems gibt es verschiedene Szenarien, wann ein Auftrag durch das externe System weiterverarbeitet wird:

1. Speed4Trade CONNECT holt nur den Auftrag ab. Alle weiteren Prozesse im externen System.
2. Speed4Trade CONNECT holt den Auftrag ab und verschickt die Erstbenachrichtigung. Alle weiteren Prozesse im externen System.
3. Speed4Trade CONNECT holt den Auftrag ab, verschickt die Erstbenachrichtigung und führt einen Abgleich der Zahleingänge durch. Alle weiteren Prozesse im externen System.
4. Speed4Trade CONNECT holt den Auftrag ab, verschickt die Erstbenachrichtigung, führt einen Abgleich der Zahleingänge durch, erstellt die Exporte für Versanddienstleister und die Rechnung. Der Auftrag wird nur für Statistiken vom externen System abgeholt.

Neben den allgemeinen Auftragsdaten, der Rechnungs- und Lieferanschrift werden bei den Positionen auch verkaufskanalspezifische Positionsdaten zurückgeliefert, wie bspw. die eBay-Artikelnummer.

Speed4Trade CONNECT Pfad: Auftrag → Auftragsdaten → Grunddaten

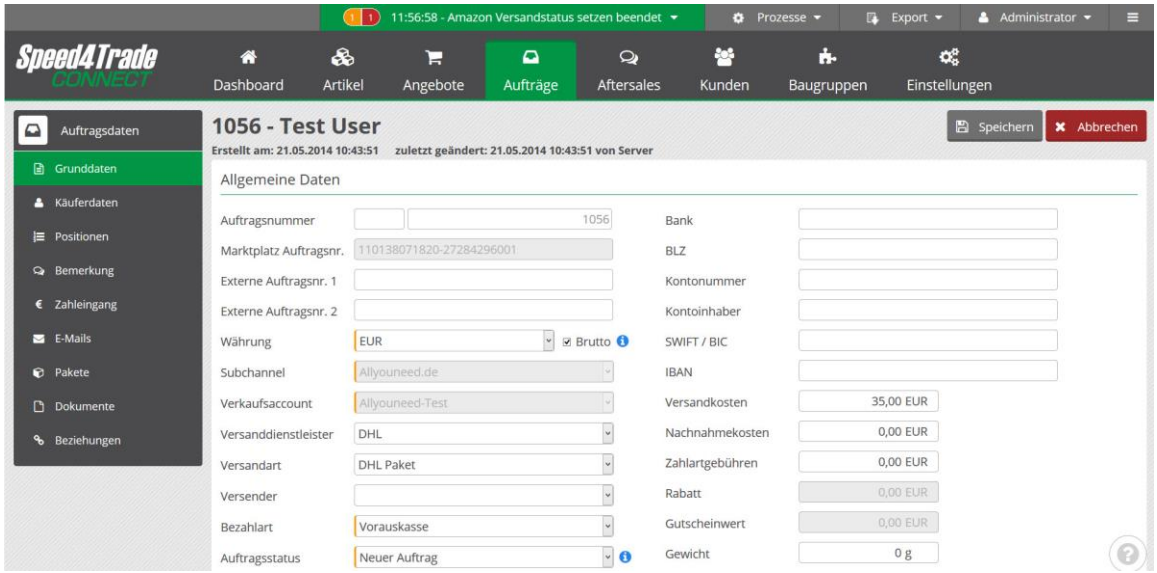

Abbildung 31: Auftragsdaten

11.2.1 Standard-Auftragsworkflow

Die Definition des Workflows sollte in jedem Fall mit der Speed4Trade GmbH abgesprochen werden, damit es möglichst wenig „Brüche“ oder doppelte Arbeit für den Anwender gibt. Nachfolgend beispielhaft ein möglicher Auftragsworkflow:

1. Kunde kauft auf Marktplatz / Onlineshop
2. Abholen der Aufträge durch Speed4Trade CONNECT
3. Versand der Erstbenachrichtigungs-E-Mail an Käufer über Speed4Trade CONNECT
4. Abholen der Aufträge von der Warenwirtschaft über Schnittstelle
   → WebService-Call: *fetchOrders*
5. Setzen des "WaWi-Export"-Flags (das Feld export_date enthält ein Datum) sowie der externen Auftrags- und Kundennummer
   → WebServiceCall: *updateOrders*
6. Durchführung des Zahlungsabgleichs in der Warenwirtschaft
7. Alle Statusmeldungen an Speed4Trade CONNECT → WebServiceCall: *updateOrders*
8. Automatischer Versand der Zahlungseingangs-E-Mail an den Käufer über Speed4Trade CONNECT
9. Übergabe des Status an Marktplatz / Onlineshop durch Speed4Trade CONNECT
10. Durchführung des Versands in der Warenwirtschaft
11. Statusmeldung an Speed4Trade CONNECT → WebServiceCall: *updateOrders*
12. Automatischer Versand der Versandbestätigungs-E-Mail an den Käufer über Speed4Trade CONNECT
13. Übergabe des Status an Marktplatz / Onlineshop durch Speed4Trade CONNECT

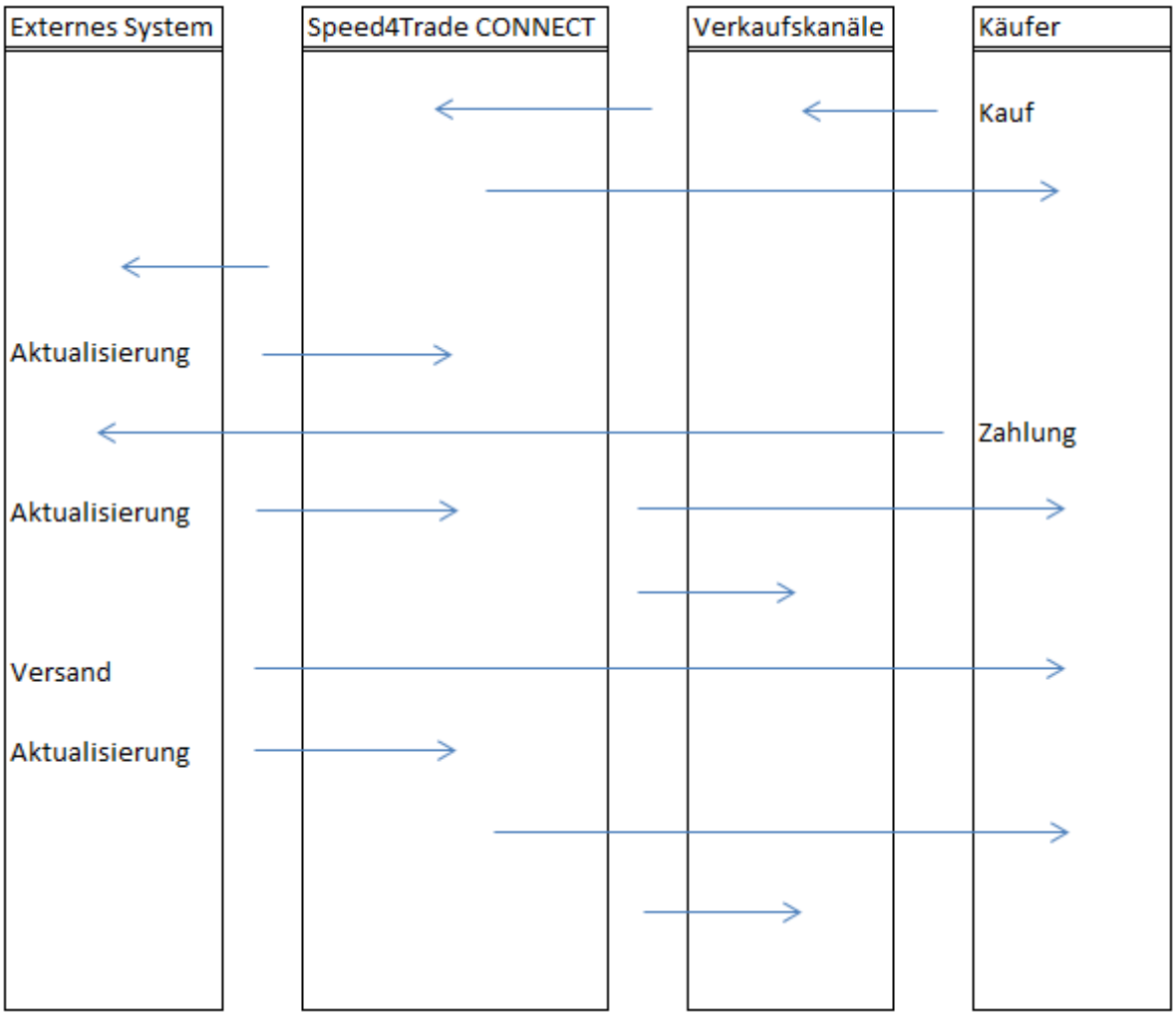

Abbildung 32: Auftragsworkflow

Hinweise!
Die Auftragsnummer aus dem externen System sollte in das Feld „Externe Auftragsnummer 1“ gespeichert werden, d.h. nach dem Abholen der Aufträge sollte gleich eine Aktualisierung des Auftrags durchgeführt werden.

Da durch den Auftragsimport eine Bestandsänderung im externen System vorgenommen wird, sollte auch gleich eine Bestandsaktualisierung erfolgen.

11.2.1.1 Dubletten-Prüfung

Die Aufträge werden immer mit den Auftrags- und Kundendaten in Speed4Trade CONNECT bereitgestellt. Die Dubletten-Prüfung der Kundendaten erfolgt anhand der E-Mail Adresse.

Sollte die Warenwirtschaft über eine erweiterte Dubletten-Prüfung verfügen, ist diese nachzuschalten. Es kann vorkommen, dass Käufer über die unterschiedlichen Verkaufskanäle mit unterschiedlichen E-Mail Adressen einkaufen. Auf jeden Fall wird den Käufern bei Amazon eine eigene, von Amazon generierte, E-Mail Adresse zugewiesen.

11.2.1.2 Aufträge mit Kunden-Kommentaren

Aufträge, die einen Kommentar durch einen Käufer beinhalten, sollten in der Warenwirtschaft gesondert behandelt werden, da diese durch einen Mitarbeiter des Verkäufers geprüft werden muss.

11.2.2 Auftragsworkflow bei eBay-Aufträgen ohne abgeschlossenen eBay-Checkout

1. Kunde kauft auf Marktplatz eBay und schließt Checkout nicht ab
2. Abholen der Aufträge durch Speed4Trade CONNECT mit Standardzahlungsart
3. Versand der Erstbenachrichtigungs-E-Mail an Käufer über Speed4Trade CONNECT
4. Abholen der Aufträge von der Warenwirtschaft über Schnittstelle
   → Webservice-Call: *fetchOrders*
5. Anlegen z.B. eines sog. eBay-Vorbelegs (Info-Beleg) in der Warenwirtschaft zur Bestands-reservierung
6. Kunde schließt Checkout-Prozess ab
7. Abholen des "Complete"-Status von der Warenwirtschaft über Schnittstelle
   → Webservice-Call: *fetchOrders*
8. Setzen des "WaWi-Export"-Flags sowie der externen Auftrags- und Kundennummer
   → Webservice-Call: *updateOrders*
9. Durchführung des Zahlungsabgleichs in der Warenwirtschaft
10. Statusmeldung an Speed4Trade CONNECT → Webservice-Call: *updateOrders*
11. Automatischer Versand der Zahlungseingangs-E-Mail an Käufer über Speed4Trade CONNECT (oder Kommunikations-Automatismen der WaWi nutzen)
12. Übergabe des Status an Marktplatz / Onlineshop
13. Durchführung des Versands in der Warenwirtschaft
14. Statusmeldung an Speed4Trade CONNECT → WebServiceCall: *updateOrders*
15. Automatischer Versand der Versandbestätigungs-E-Mail an Käufer über Speed4Trade CONNECT (oder Kommunikations-Automatismen der WaWi nutzen)
16. Übergabe des Status an Marktplatz / Onlineshop

Wird ein eBay-Auftrag von Speed4Trade CONNECT abgeholt, bei welchem der Checkout noch nicht abgeschlossen ist, dann sollte in der Warenwirtschaft ein sog. eBay-Vorbeleg angelegt werden. Dieser dient dazu, dass in der Warenwirtschaft bereits eine Bestandsreservierung stattfinden kann. Zu erkennen sind diese Aufträge am „Complete“-Status. Nachdem der Checkout bei eBay abgeschlossen wurde, muss noch einmal der Webservice-Call "*fetchOrders*" durchgeführt werden. Danach wird über den Webservice-Call "*updateOrders*" das Flag "WaWi-Export" (das Feld export_date enthält ein Datum) gesetzt. Im Anschluss daran kann der Auftrag entsprechend in der Warenwirtschaft weiter verarbeitet werden.

Hinweis!
Ein noch offener Checkout bei eBay könnte per Parametereinstellung durch die Speed4Trade CONNECT nach einem definierten Zeitraum abgeschlossen werden. Dies ist aber nicht sehr kundenfreundlich.

Besser - in Abhängigkeit der Warenwirtschaftsmöglichkeiten: Wenn bspw. ein eBay-Vorbeleg zu einem Auftrag in der Warenwirtschaft vorhanden ist und es kann diesem ein Zahlungseingang zugeordnet werden, dann soll der Vorbeleg automatisch in eine Rechnung umgewandelt werden. Danach muss der Webservice-Call "*updateOrders*" durchgeführt werden, damit das "WaWi-Export"-Flag, externe Auftrags- und Kundennummer in der Speed4Trade CONNECT gesetzt wird. Speed4Trade CONNECT schließt immer beim Setzen des „WaWi-Export“-Flags den Checkout von eBay.

11.2.3 Auftragsworkflow bei Amazon-Aufträge - auf E-Mail warten

Da es bei Amazon keinen direkten Checkout-Status gibt, muss hier zusätzlich auf den Auftragsstatus abgeprüft werden.

Der Workflow für Amazon-Aufträge kann wie nachfolgend gestaltet werden:

1. Auftrag wird mit den Kopfdaten abgeholt (noch keine E-Mail-Adresse) = Auftragsstatus "Auftragsrückstand" (ID 995) -> Speed4Trade CONNECT Reiter - Rückstellungen
2. Auftrag wird von Amazon als gültiger Vorgang bestätigt (mit E-Mail-Adresse) = Auftragsstatus "Bestätigt/Versenden" (ID 220) -> Speed4Trade CONNECT Reiter - Versand
3. Auftrag wird als versendet markiert (manuell oder durch Warenwirtschaft) = Auftragsstatus "Versendet/Wartet auf Bezahlung" (ID 230) -> Speed4Trade CONNECT Reiter - Offene Zahlungen
4. Auftrag wird von Speed4Trade CONNECT als bezahlt markiert = Auftragsstatus "Bezahlt/Abgeschlossen" (ID 240) -> Reiter Speed4Trade CONNECT - Alle Aufträge

Amazon-Aufträge dürfen daher erst dann abgeholt werden, wenn der Auftragsstatus "Bestätigt/ Versenden" (ID 220) gesetzt ist. Im Webservice-Call für die Amazon-Aufträge muss folglich neben dem Flag <export_date> zusätzlich auch auf den Status 220 bei <order_status_id> eingeschränkt werden.

Hinweis!
Zu 3. Senden des Amazon „Versand-Status“
Der „versendet“ Status (ID 230) darf nur im Bulk und nicht je Auftrag gesendet werden.

Zu 4. Senden des Amazon „Bezahl-Status“.

Option 1: „Best Practice“
Speed4Trade CONNECT holt von Amazon den Settlement-Report ab und verarbeitet diesen intern automatisch → externes System darf keinen Bezahlt-Status liefern

Option 2: „von externem System gesteuert“
Externes System liefert den „Bezahlt-Status“. Settlement-Report-Abholung in Speed4Trade CONNECT muss dann deaktiviert sein! Externes System darf den „Bezahlt Status“ nur für Aufträge aktualisieren, die in Speed4Trade CONNECT den Status (ID 230) besitzen.

11.3 Webservice-Call: fetchOrders

Standard

Mit dem Webservice-Call fetchOrders können alle Aufträge aus der Speed4Trade CONNECT abgerufen werden. Beachten Sie dazu die umfangreichen Filtermöglichkeiten zum Call (Aufträge von/bis, nur mit bestimmten Auftragsstatus, usw.)

Benötigte vorbereitende Calls:

- fetchMandators (Mandanten abfragen)
- fetchSubChannels (Sub-Channels abfragen)
- fetchAccounts (Accounts abfragen)

11.3.1 Relevante Felder im Auftrag

Nachfolgend eine Liste der am häufigsten benötigten Felder beim Abrufen von Aufträgen. Eine vollständige Liste ist in der online Dokumentation der Webservice-API beschrieben.

11.3.1.1 Aufträge -> Grunddaten

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Auftragsnummer	Präfix + Speed4Trade CONNECT Auftragsnummer (Getrennte Felder!)	order_number_prefix order_number
Marktplatz Auftragsnummer	Wird von den Marktplätzen wie bspw. eBay, Amazon und vom Shop geliefert Muss von der Warenwirtschaft verarbeitet werden	marketplace_order_id
Externe Auftragsnummer 1	Kann durch die Warenwirtschaft gesetzt werden, damit in beiden Systemen nach der gleichen Nummer gesucht werden kann	external_order_number_1
Externe Auftragsnummer 2	Kann durch die Warenwirtschaft gesetzt werden, damit in beiden Systemen nach der gleichen Nummer gesucht werden kann	external_order_number_2
Währung	Währung des Auftrags	currency_code_iso
Brutto	net / gross 0 / 1	price_type_identifier, alternativ: price_type_id
Subchannel	Wird von Marktplatz übertragen	sub_channel_identifier
Verkaufsaccount	Wird von Marktplatz übertragen	account_identifier
Versanddienstleister		shipping_provider_name

Versandart	Wird zum Teil von Marktplatz mit übertragen.	identifier (im wrapper element dispatch_type)
Bezahlart	Wird von Marktplatz übertragen	payment_method_identifier
Auftragsstatus	Wird im Workflow über die Warenwirtschaft gesetzt	order_status_identifier
Bezahlstatus	Wird über die Warenwirtschaft gesetzt	payment_status
Rückstellungsgründe	ID eines Rückstellungsgrundes zur vorübergehenden Unterbrechung des Auftragsworkflows.	hold_reason_id
Rückstellungsbemerkungen	"Fulfillment By Amazon" → Kennzeichen eines FBA-Auftrags von Amazon	hold_reason
WaWi Export	Wird über Warenwirtschaft gesetzt Webservice-Call: updateOrder	export_date
Bank	Wird ggf. nur vom einem Shop gesetzt	bank_name
BLZ	Wird ggf. nur vom einem Shop gesetzt	bank_code
Kontonummer	Wird ggf. nur vom einem Shop gesetzt	bank_account_number
Kontoinhaber	Wird ggf. nur vom einem Shop gesetzt	depositor
SWIFT / IBC	Wird ggf. nur vom einem Shop gesetzt	swift_code
IBAN	Wird ggf. nur vom einem Shop gesetzt	iban_code
Versandkosten	Werden vom Marktplatz übertragen	shipping_charge
Nachnahmekosten	Werden vom Marktplatz übertragen	cash_on_delivery_charge
Zahlartgebühren	Werden vom Marktplatz übertragen in Abhängigkeit des dort eingesetzten Payment-Anbieters	payment_method_charge
Rabatt	Kann sowohl positiv als auch negativ sein	rebate
Gutscheinwert	Kann sowohl positiv als auch negativ sein	coupon_percent
Gewicht	Gewicht in Gramm	weight_in_gram
Auftragskennzeichen	Priority Shipping und Click&Collect	Wrapper Element order_tags

11.3.1.2 Aufträge ->Pakete

Speed4Trade CONNECT Feldbezeichnung	Beschreibung	Feldbezeichnung Webservice
Versanddienstleister	muss über Warenwirtschaft gesetzt werden	Call *updateOrders* -> shipping_carrier_identifier

Paketnummer	muss über Warenwirtschaft gesetzt werden	Call *updateOrders* -> tracking_number

Wenn dem Käufer ein Tracking-Code für die Paketnachverfolgung übermittelt werden soll (per E-Mail oder bspw. zu eBay, Amazon), dann müssen die oben aufgeführten Felder bei der Übergabe des Status "Versand" durch die Warenwirtschaft gesetzt werden. Es können mehrere Pakete eines Auftrags mit unterschiedlichen Tracking-Nummern an Speed4Trade CONNECT übergeben werden.

11.3.1.3 Auftragsstatus: umgruppierte Aufträge bei eBay

Bei einem eBay-Auftrag kann es vorkommen, dass dieser den Auftragsstatus "Umgruppiert" hat. Dieser Status wird angezeigt, wenn ein Käufer 2 Aufträge zu einem Auftrag zusammenführt, während der jeweilige Checkout noch nicht abgeschlossen war. Die Aufträge, welche den Status "Umgruppiert" haben, haben keine Auftragspositionen mehr.

Bei der Verarbeitung von umgruppierten eBay-Aufträgen ist folgendes zu beachten:
Wenn ein Käufer jeweils einen Artikel bei eBay kauft, so dass zwei Aufträge erzeugt werden und den Checkout nicht abschließt, dann werden diese beiden Aufträge von der Speed4Trade CONNECT mit der definierten Standardzahlungsart abgeholt. Anschließend holt die Warenwirtschaft die Aufträge von der Speed4Trade CONNECT über die Schnittstelle zur weiteren Verarbeitung ab. In der Warenwirtschaft werden bzgl. der Bestandsbuchung zwei Info-Belege - einer pro Auftrag - angelegt. Ist zu den Aufträgen bereits ein Zahlungseingang vorhanden, dann werden automatisch im ERP-System die Info-Belege in Rechnungen umgewandelt. Werden nun parallel die Aufträge bei eBay durch den Käufer zu einem gruppiert, dann werden beim erneuten Abholen der Aufträge diese auch in Speed4Trade CONNECT zu einem zusammengefasst. Das bedeutet, dass die Positionen einem Auftrag zugeordnet werden. Damit die umgruppierten Aufträge von der Warenwirtschaft über die Schnittstelle von der Speed4Trade CONNECT abgeholt werden können, muss nochmals der Webservice-Call "*fetchOrders*" mit einem zusätzlichen Filterkriterium aufgerufen werden. Nachfolgend finden Sie hierfür einen Beispiel-Call.

```
<request method="fetchOrders" version="1.0.0">
<filter>
      <mandator_id filter_method="=" filter_value="1" />
      <order_status_id filter_method="=" filter_value="985" />
      <last_changed from_date="2012-01-01T00:00:00.000+02:00"
allow_null="true" />
</filter>
</request>
```

11.4 Webservice-Call: updateOrders

Standard

Die Aktualisierung des Auftrags in Speed4Trade CONNECT durch das externe System ist ein sehr wichtiger Vorgang, um den Käufer über den aktuellen Stand seiner Bestellung zu informieren und somit den Kommunikationsaufwand mit dem Käufer zu minimieren.

Beispielszenario:

1. Nur Erstbenachrichtigung in Speed4Trade CONNECT

2. Restlicher Prozess im externen System
3. Es handelt sich um einen eBay Auftrag mit Bezahlart „Vorauskasse“

Nach Zahleingang im externen System wird der „Bezahlt“-Status an Speed4Trade CONNECT übergeben. Speed4Trade CONNECT wiederum gibt diesen Status an eBay weiter, so dass der Käufer dies in seinem „Mein eBay“-Bereich erkennen kann.

Nach Versand der Ware gibt das externe System diesen „Versendet“-Status – im Optimalfall inkl. der Trackingnummer – an Speed4Trade CONNECT weiter. Speed4Trade CONNECT setzt daraufhin wieder den Status in eBay und übergibt – sofern vorhanden – die Trackingnummer.

Insbesondere bei Amazon ist es sehr wichtig, dass der „Versendet“-Status übergeben wird. Jeder Auftrag bei Amazon muss zwingend als „versendet“ gekennzeichnet werden, damit man das Geld von Amazon für diesen Auftrag erhält. Geschieht dies innerhalb eines bestimmten Zeitraums nicht, dann geht Amazon davon aus, dass die Ware nicht versendet wurde und das Geld wird wieder an den Käufer zurücküberwiesen.

Hinweis:
Eine Anlage von Aufträgen über diesen Call ist nur möglich, wenn es sich um den Kanal „Manual Sales“ handelt, z.B. Telefonverkauf. Aufträge zu Marktplätzen oder Shopsystemen dürfen nicht auf diesem Weg angelegt werden, da es in den nachfolgenden Prozesses zu Fehlern in der Abstimmung mit den Markplätzen oder Shops kommen kann.

Benötigte vorbereitende Calls:

- fetchMandators (Mandanten)
- fetchSubChannels (Sub-Channels)
- fetchAccounts (Accounts)

12 Weitere Funktionen

12.1 E-Mail Benachrichtigung an Käufer

Die Benachrichtigung des Kunden über den Auftragsstatus erfolgt via E-Mail über die Speed4Trade CONNECT oder die Warenwirtschaft. Dies muss im konkreten Fall festgelegt werden. Auf keinen Fall sollten E-Mail von beiden Systemen doppelt versandt werden. Es ist in Speed4Trade CONNECT möglich, für jeden einzelnen Status den automatischen Versand festzulegen.

12.2 Zahlungsabgleich

Der Zahlungsabgleich findet in der Regel über PayPal und Bankkonto direkt in der Warenwirtschaft statt. Die Settlement-Reports von Amazon werden direkt über den Amazon-Webservice über Speed4Trade CONNECT eingelesen, in Speed4Trade CONNECT verarbeitet und die dem Auftrag zugewiesenen Zahlungen können in das ERP-System via „*fetchPayments*“ abgeholt werden.

Beim Setzen des Bezahlstatus über die Schnittstelle, wird immer der komplette Bruttobetrag an Speed4Trade CONNECT übertragen.

12.2.1 Webservice-Call: fetchPayments

Erweitert

Sollte Ihr System keine Möglichkeit haben Paymentdaten (z.B. PayPal) abzugleichen, so kann dies auch über Speed4Trade CONNECT erfolgen. Die Zuweisung von Zahlungen zu Aufträgen wird dadurch im externen System deutlich erleichtert.

Der Workflow ist dabei wie folgt:

1. Die Aufträge werden durch das externe System aus Speed4Trade CONNECT abgeholt.
2. Speed4Trade CONNECT holt über die Automatismen Zahlungseingänge z.B. von PayPal ab.
3. Über den *fetchPayments*-Call können alle Zahleingänge ermittelt werden. Zahleingänge, die automatisiert durch Speed4Trade CONNECT zugeordnet werden konnten, enthalten dabei gleich die Referenz zum Auftrag.

Hinweis:
Bei Amazon kann über fetchReports der gesamte unverarbeitete Settlement Report abgeholt werden, welcher weitere Detailinformationen enthält.

12.3 Versand der Aufträge über FBA

Die Aufträge dürfen mittels FBA nur verschickt werden, wenn der komplette Auftrag darüber abgewickelt werden kann, d.h. alle Positionen mit allen Mengen können geliefert werden. Der Versand von Teilmengen oder einzelnen Positionen eines Auftrages über FBA ist nicht möglich.

Hinweis!
Damit Aufträge z.B. aus dem eigenen Online-Shop via FBA verschickt werden können, müssen hierfür entsprechende Webservice-Calls ausgeführt werden. Da sich diese Erweiterung des Webservices noch in der Realisierung befindet, werden die entsprechenden Calls zu einem späteren Zeitpunkt in diesem Dokument ergänzt.

12.4 Teillieferungen

Es können über Speed4Trade CONNECT keine Teillieferungen abgewickelt werden. Sollte es in der Warenwirtschaft zu einer Teillieferung kommen, dann muss in der Anbindung entschieden werden, ob der Auftrag in Speed4Trade CONNECT nach der ersten Teillieferung oder erst nach dem kompletten Versand der Ware auf „fertiggestellt/versendet“ gesetzt wird.

12.5 Aftersales

12.5.1 eBay

Die Abgabe von Bewertungen und das Anstoßen von Unstimmigkeiten aus dem Aftersales-Bereich kann über Webservice-Calls durch die Warenwirtschaft ausgelöst werden.

12.5.2 Amazon

Das Anpassen und Stornieren von Amazon-Aufträgen erfolgt in zwei Stufen:

1. Durchführung der Stornierung in Speed4Trade CONNECT
2. Durchführung der Stornierung bei Amazon

Komplettstorno:

- Im Call **updateOrders** die order_status_id auf 990 (storniert) setzen.
- Danach den Call **cancelOrders** ausführen

Teilstornos:

- Im Call **updateOrders** entsprechend die Menge anpassen bei den betreffenden Auftragspositionen bei jedem order_item. Die order_status_id nicht verändern.
- Danach den Call **adjustOrders** ausführen

Gutschrift (Ware wird vom Kunden nicht zurückgesendet):

- Keine Anpassung des Auftrags.
- Es reicht aus den Call **createCreditEntry** auszuführen.

13 Fernsteuerung von Speed4Trade CONNECT

13.1 Allgemein

Wenn ein Artikel an Speed4Trade CONNECT übergeben wird, dann wird dieser nicht automatisch in einem Marktplatz oder Shop eingestellt. Wenn ein Warenwirtschaftssystem aber alle Daten, die für das Einstellen notwendig sind, liefern kann, dann können alle relevanten Prozesse angestoßen werden.

13.2 Workflow zum automatischen Einstellen

Erweitert

Der Ablauf ist dabei wie folgt:

5. Durch den Call *updateItems* wird ein Artikel mit den marktplatzspezifischen Daten in Speed4Trade CONNECT angelegt. Als Antwort erhält man durch die Speed4Trade-CONNECT-API die interne item_id des Artikels.
6. Via *prepareOffers* und der item_id kann man bspw. für eBay ein Angebot erzeugen. Dabei erhält man eine offer_id zurück.
7. Diese offer_id kann wiederum im Call *uploadOffers* verwendet werden, um den Artikel direkt bei eBay einzustellen.

<u>Hinweis:</u>
Bitte beachten Sie an der Stelle, dass das Hochladen eines Angebots zu einem Marktplatz asynchron läuft, d.h. *uploadOffers* stößt zwar den Upload-Vorgang an, das eigentliche Hochladen kann etwas zeitversetzt sein, so dass der Response von *uploadOffers* nur anzeigt, dass die Anfrage entgegengenommen wurde. Sie gibt aber keinen Hinweis darauf, ob der Upload zum Marktplatz an sich fehlerfrei verlaufen ist. Sie können aber mittels *fetchOffers* ermitteln, ob es nach einer gewissen Zeit noch dieses spezielle „Vorbereitete Angebot“ gibt.

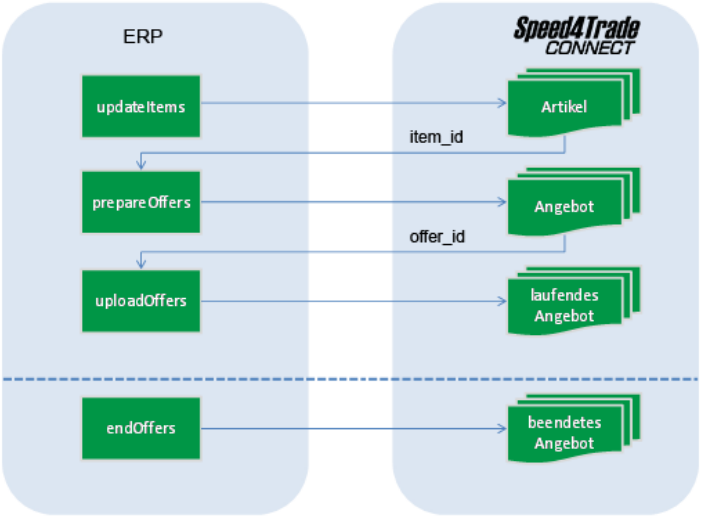

Abbildung 33: Workflow zum automatischen Einstellen

13.3 Abgleich von neuen Artikeldaten zu Marktplätzen

Erweitert

Bis auf den Bestand werden neue Artikeldaten nicht automatisch mit Marktplätzen abgeglichen. Neue Artikeldaten können entweder über den Speed4Trade-CONNECT-Client oder via *syncPlatformItems* in Richtung Marktplätze und Shops aktualisiert werden.

13.4 Masken von Speed4Trade CONNECT in einem Container anzeigen

Erweitert

In bestimmten Implementierungen ist es wünschenswert, dass Masken aus Speed4Trade CONNECT direkt in einer Warenwirtschaft angezeigt werden. Dazu muss das Warenwirtschaftssystem einen Container z.B. einen IFrame, ein WebControl o.ä. anbieten können, an den eine URL übergeben wird. Diese URL muss einen bestimmten Aufbau haben, damit auch die Anmeldung an der Speed4Trade CONNECT automatisch erfolgen kann.

13.4.1 Aufbau der Masken-Adresse für den automatischen Login

Die URL für den Aufruf einer Speed4Trade-CONNECT-Maske besteht aus zwei Teilen. Zum einen aus der Standard URL zum Aufruf der Web-Maske, z.B. https://Servername:Portnummer/emmidaweb/de-de/items/common.html. Und zum anderen aus den zusätzlichen Parametern, die für die automatische Anmeldung der Warenwirtschaft an Speed4Trade CONNECT notwendig sind. Diese sind im Einzelnen:

- mid = Mandant-ID
- user = Benutzername in der Speed4Trade CONNECT
- nonce = Zeitstempel
- seckey = Signatur zur Authentifizierung

Wenn alle Parameter berechnet und mit der aufzurufenden Masken-Adresse zusammengesetzt sind, ergibt sich eine Gesamt-URL, die wie folgt aufgebaut ist:
https://Servername:Portnummer/emmidaweb/de-de/items/common.html?itemId=3747503&mid=1&user=max_mustermann&nonce=201401101345&secKey=dGXKxOVp2dwnsa8Iz2Itu%2b4QL4kObGkzC%2fi15Pt2h2MgcC0%2bisLbgg%3d%3d

13.4.2 Informationen zur Authentifizierung

Hat sich der Nutzer einmal angemeldet, bleibt er so lange angemeldet, bis er sich wieder abmeldet. Um Replay-Attacken zu vermeiden, wird ein Zeitstempel als Parameter übertragen. Liegt der Zeitstempel außerhalb eines 10-Minuten-Intervalls, so schlägt der Login fehl bzw. wenn der Benutzer schon angemeldet ist, wird keine Seite angezeigt.

Als Parametername wird „nonce“ verwendet und enthält den Zeitstempel im Format Jahr, Monat, Tag, Stunden, Minuten (YYYYMMDDHHmm).

Beispiel für die technische Umsetzung:
Parameter:

- itemId=3747503
- mid=1
- user=max_mustermann
- nonce=201401101345

Alle Parameter werden als Zeichenkette zusammengesetzt (ohne Trennzeichen):
itemId=3747503mid=1user=max_mustermannnonce=201401101345

Für diese Zeichenkette wird dann über das DSA-Verfahren eine Signatur erstellt und als Parameter „seckey“ wie folgt an den URL-String angehängt:
https://Servername:Portnummer/emmidaweb/de-de/items/common.html?itemId=3747503&mid=1&user=max_mustermann&nonce=201401101345&secKey=dGXKxOVp2dwnsa8Iz2Itu%2b4QL4kObGkzC%2fi15Pt2h2MgcC0%2bisLbgg%3d%3d

Für die Generierung der Signatur wird in der Speed4Trade CONNECT ein privater Schlüssel für den Mandanten generiert, der auch in die Warenwirtschaft übernommen und dort sicher hinterlegt werden muss. Den Login-Schlüssel findet man in der Speed4Trade CONNECT im Bereich Extras -> Grundeinstellungen -> Mandanten -> Sonstiges -> Externe Systeme -> Login-Schlüssel.

Dieser Schlüssel ist der private Schlüssel aus einem DSA-Schlüsselpaar im Format PKCS#8 im Encoding DER und zur besseren Übertragbarkeit als Zeichenkette noch Base64-kodiert.

Um die Anbindung etwas einfacher zu gestalten, gibt es im Partnerportal ein Paket mit Programmierbeispielen inkl. Bibliotheken, Quellcode und Demo-Applikation für die Berechnung einer Signatur mit DAS. Für die beiden Programmiersprachen JAVA und C# gibt es jeweils ein Paket zum Herunterladen. Diese Daten können für die Anbindung an die Speed4Trade CONNECT frei genutzt werden.

13.5 Ausblenden des Rahmens des Speed4Trade-CONNECT-WEB-Templates

Damit die Maske der Speed4Trade CONNECT besser in einem Container, z.B. IFrame, eingebunden werden kann, gibt es die Möglichkeit, den Rahmen auszublenden. Dieser Rahmen ist für die Standalone-Nutzung notwendig, um zwischen Artikeln, Angeboten und Aufträgen hin und her zu wechseln.

Dafür muss in der URL noch ein weiterer Parameter „frameless“ angegeben werden. Es können hier 2 Werte angegeben werden. frameless=1 schaltet den Rahmen aus, frameless=0 blendet ihn wieder ein. Der Standardwert ist 0, wenn der Parameter nicht mit angegeben ist.

13.6 Definition einer Callback-Adresse

Über einen zusätzlichen Parameter in der Masken-Adresse (URL) kann man das Verhalten der angezeigten Speed4Trade CONNECT -Maske nach dem Speichern steuern. Hierzu gibt es 3 unterschiedliche Fälle, die am Beispiel der Pflege von Artikeldaten verdeutlicht werden.

13.6.1 Parameter callback nicht gesetzt

Wenn der Parameter 'callback' nicht gesetzt ist, verhält sich die Eingabemaske analog einem direkten Aufruf von Speed4Trade-CONNECT-WEB. Dies bedeutet im obigen Beispiel unter Punkt 13.4.1, sobald der Benutzer mit der Bearbeitung der Maske fertig ist und auf Speichern oder Abbrechen klickt, wird zu einer übergeordneten Maske gesprungen. In unserem Beispiel editiert der Anwender die Daten eines Artikels und gelangt danach zurück zur Artikelliste.

13.6.2 Parameter callback = no

Wird der callback-Parameter mit dem Wert „no“ übergeben, wird keine Abbrechen-Schaltfläche angezeigt. Beim Klick auf die Speichern-Schaltfläche werden die Eingaben gespeichert, aber die Maske bleibt danach offen.

Verdeutlicht an unserem obigen Beispiel heißt dies, dass der Benutzer immer auf der aktuellen Maske zum Bearbeiten eines Artikels bleibt, die von der Warenwirtschaft aus aufgerufen wurde. Die URL dazu sieht folgendermaßen aus: https://.../items/commin.html?itemId=4711...&**callback=no** &seckey=...

13.6.3 Parameter callback auf eine andere Maske setzen

Mit dem Parameter 'callback' kann auch direkt der Fluss zwischen verschiedenen Maskenaufrufen gesteuert werden. Damit kann man von der Warenwirtschaft aus festlegen, welche Folgemaske nach der aktuell Aufgerufenen angezeigt werden soll. Da parallel kein 'callback = no' gesetzt werden kann, ist das Verhalten nach dem Aufruf der Folgemaske wie unter Punkt 13.6.1 beschrieben. Das heißt, das weitere Verhalten der Maske ist identisch zum Standardaufruf von Speed4Trade-CONNECT-WEB über den Browser.

Die Beispiel-URL hierzu kann wie folgt aussehen:
https://.../items/common.html?itemId=4711...&**callback=orders/list.html**&seckey=...
Es wird die Bearbeitungsmaske des Artikels mit der ID 4711 geöffnet. Ist der Anwender mit der Bearbeitung der Eingabemaske fertig, kann er speichern oder abbrechen. In der Folge wird dann die Maske mit der Auftragsliste angezeigt.

14 Zusammenfassung

Anhand dieses Dokuments wurden Ihnen die wichtigsten Einstellungen aufgezeigt, die man in Speed4Trade CONNECT setzen muss, bevor man mit der eigentlichen Anbindung beginnen kann. Diese Grundkonfiguration wird dabei entweder durch den Anwender selbst oder in Zusammenarbeit mit der Supportabteilung der Speed4Trade GmbH durchgeführt.

Damit keine falschen Erwartungen beim Anwender entstehen und um den genauen Funktionsumfang abzustimmen, sollten Sie sich rechtzeitig mit der Speed4Trade GmbH in Verbindung setzen. Somit kann bei Schulungen genau darauf eingegangen werden, was der Anwender in der Speed4Trade CONNECT nachpflegen muss, wie er dies am schnellsten machen kann und wie der Workflow bei den Aufträgen ist.

Die beschriebenen Funktionen bilden nur das Grundgerüst, das aus Sicht der Speed4Trade GmbH vorhanden sein sollte. Natürlich kann es auch zu Abweichungen kommen, dies hängt sehr stark vom Anforderungsprofil des Anwenders ab.

14.1 Übersichtsdiagramm der Workflows

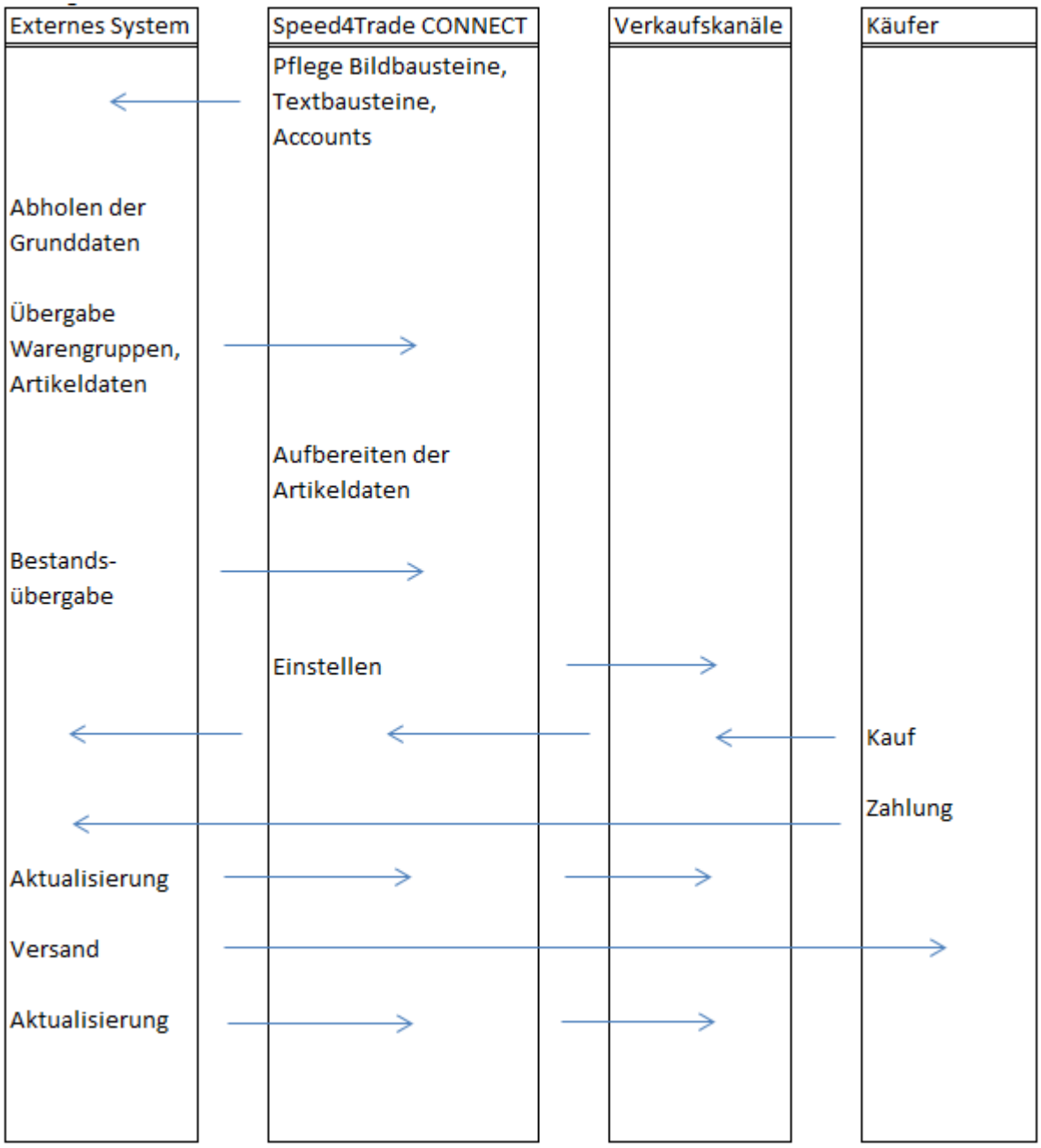

Abbildung 34: Übersichtsdiagramm der Workflows