

Projekt QS-Plattform

Bündlerschnittstelle

White Paper

Autor	Rez Eslami
Angelegt	18.05.2007
Revision	1.5
Stand	12.11.2010 12:10
Status	Entwurf

Dokument Versionen			
Version	Autor	Beschreibung der Version	Datum
1.3	Reza Eslami	Endversion	21.06.2007
1.4	Michael Chodorowski	Ergänzungen in den Funktionen 6.3 und 6.4	03.09.2009
1.5	Oliver Bandke	Ergänzungen in den Funktionen 6.1, 6.3, 6.25,6.26 und 6.27 Neue Funktionen 6.28	15.07.2010

Inhaltsverzeichnis:

Projekt QS-Plattform 1

1 Einleitung 5

 Allgemeines zum Header:5

 Allgemeines zur Verschlüsselung und Authentifizierung:5

 Anmerkung bezüglich der QS Status Tabelle.....7

 Allgemeine Vorgehensweise7

 Funktion 6.1 CQSSRegisterMemberMsg:8

 Funktion 6.2 CQSSAddLocationMsg:..... 11

 Funktion 6.25 CQSSRegisterAddCertificationSystemMsg und 6.26
 CQSSAddCertificationSystemMsg: 11

 Funktion 6.27 QSChangeAttributeMsg:..... 12

 Funktion 6.3 QSExtLocationQuery und QSExtLocationStandardQuery: 12

 Funktion 6.4 QSExtLocationBtartQuery und QSExtLocationBtartStandardQuery: 13

 Funktion 6.5 QSBdlRequestMsg: 14

 Funktion 6.6 QSBdlStortInfoMsg:..... 15

 Funktion 6.7 QSUpdateLocationMsg: 15

 Funktion 6.8 QSUpdateBtartMsg: 16

 Funktion 6.9 QSUpdateCertBodyMsg: 16

 Funktion 6.10 CQSSUpdateLocAdrMsg:..... 16

 Funktion 6.11 **CQSSUpdateLocComMsg**:..... 17

 Funktion 6.12 QSDeleteMemberMsg 17

 Funktion 6.13 QSDeleteLocationMsg..... 17

 Funktion 6.14 QSLastInspectionMsg 18

 Funktion 6.15 QSLastSentInspectionMsg 18

 Funktion 6.16 QSExtInspectionMsg 18

Funktion 6.17 QSInspectionFromIdMsg	18
Funktion 6.18 QSExtInspectionsInPeriodMsg	19
Funktion 6.19 QSLocationInfoMsg	19
Funktion 6.20 QSBuildBundleMsg	19
Funktion 6.21 CQSSUpdateCompAdrMsg:	20
Funktion 6.22 CQSSUpdateCompComMsg:	20
Funktion 6.23 QSChangeLocationDataMsg:	20
Funktion 6.24 QSCertBodyQueryMsg:	21
Funktion 6.28 QSUpdateCertificationSystemMsg:	21

1 Einleitung

Mit der Neuimplementierung der Web-Service-Schnittstelle für Bündler in der neuen QS-Software-Plattform, ist eine wichtige Funktionalität geschaffen, um den Stammdatenaustausch zwischen den Bündlerorganisationen und QS zu erleichtern. Neben den dokumentierten Nachrichten und Funktionalitäten der Bündlerschnittstelle wollen wir Ihnen im nachfolgenden Text die sonstigen Eigenheiten der Web-Service-Nachrichten erläutern und somit Einblicke in die Funktionsweise geben.

Dabei wird insbesondere auf die verschiedenen Möglichkeiten eingegangen, wann und unter welchen Bedingungen welche Fehlermeldung erzeugt bzw. geworfen wird und wie die einzelnen Felder der Schnittstelle (sowohl bei der Anfrage als auch bei der Antwort) interpretiert worden sind, also welches Mapping vorgenommen worden ist.

Dieses Dokument richtet sich daher primär an Entwickler, die direkt mit der Umsetzung dieser Schnittstelle beauftragt sind. Sie können daran überprüfen, ob das Verhalten der Schnittstelle mit ihren Erwartungen an die Funktionalität konform ist und die übermittelten Daten so interpretiert und ausgewertet werden, wie dies bisher geschehen ist. Insbesondere das Verhalten im Fehlerfall sollte dadurch klarer werden. Dieses Dokument sollte aber auf jeden Fall im Kontext der eigentlichen Definition/Spezifikation der Schnittstelle gesehen werden (<http://www.q-s.info/fileadmin/download/datenbanken/QSNachrichtenV060828.pdf>); dieser Text ist demnach als Ergänzung und Erweiterung zu verstehen.

Allgemeines zum Header:

Im Bezug auf die Altanwendung gibt es hier eine Änderung der Spezifikation des zugrundeliegenden Definition eines SOAP Servers bzgl. der SOAPAction. Seit der Soap Definition 1.1 <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/> ist es notwendig, dass diese SOAPAction definiert wird, auch wenn sie nicht gefüllt wird, weil sie im Moment noch nicht ausgewertet wird: „An HTTP client MUST use this header field when issuing a SOAP HTTP Request“. Aus diesem Grund muss im HTTP-Header der versendeten Soap-Nachricht `SOAPAction` stehen, damit die Nachricht überhaupt vom Server akzeptiert werden kann. Fehlt dieser Eintrag wird die Nachricht zurückgewiesen mit dem Fehler „no SOAPAction header!“ Dieser leere Eintrag war mit der Altanwendung möglich gewesen, aber bei der neuen Definition ist dies nicht möglich.

Allgemeines zur Verschlüsselung und Authentifizierung:

Es muss das HTTPS Protokoll verwendet werden, um auf alle Funktionen der Bündler-Schnittstelle zugreifen zu können. (Diese Lösung ist gänzlich unabhängig von der technischen Implementierung des Webservices und basiert auf einer standardisierten, technischen Umsetzung der Verschlüsselung auf Basis eines Apache Web Servers. Etwaige, diesbezügliche Probleme sind sehr schwer von unserer Seite aus zu lösen und sind wahrscheinlich auf die Konfiguration und Einstellung des jeweiligen Clients zurückzuführen, damit dieser über einen HTTPS-Tunnel kommunizieren kann.)

Die Funktionen 6.3 und 6.4 können über einen anderen Service unter einer anderen URL über das HTTP Protokoll abgefragt werden.

Außerdem muss über das HTTP Protokoll ein Username und ein Passwort übermittelt werden. Diese entsprechen den Login Daten mit denen auch der Zugang über das WebInterface erfolgt. Nur wenn dieser Login auch über das normale WebInterface funktioniert und dort nicht gesperrt oder sonst welche Probleme auftauchen, kann auch der Webservice funktionieren. Falls dieser Zugang nicht existiert, das Passwort fehlerhaft war, der Benutzername falsch angegeben worden ist, der Login-Account inaktiv bzw. gesperrt sein sollte oder ein sonstiger Fehler an dieser Stelle der Authentifizierung entsteht, wird vom Webserver immer der folgende Fehler zurückgegeben (unabhängig vom Grund für die fehlerhafte Authentifizierung).

```
AxisFault
faultCode: {http://xml.apache.org/axis/}HTTP
faultSubcode:
faultString: (401)Unauthorized
faultActor:
faultNode:
faultDetail:
    {}:return code: 401
```

Des Weiteren wird immer am Anfang der Funktionen überprüft, ob die in der Nachricht übermittelte BündlerID/QuestionerID auch tatsächlich mit dem Bündler übereinstimmt, dessen Login verwendet worden ist. Ist dies nicht der Fall, dann wird die Exception 70 = „No permission granted“ geworfen (dies geschieht ebenfalls, wenn diese ID leer bzw. nicht übermittelt worden sein sollte). Dieselbe Ausnahme 70 wird auch immer dann geworfen, wenn versucht wird auf einen Standort, Systemteilnehmer oder Auditbericht zuzugreifen, der aktuell nicht diesem Bündler zugeordnet ist.

Anmerkung bezüglich der QS Status Tabelle

In Abweichung zur Originaldokumentation wurde von QS der Typ `sla.soap.dtyp.CS8004Type` um drei Stati erweitert. Dieser Typ wurde auf Seite 11 in einer Tabelle als QS-Status beschrieben. Zustand 13 bedeutet, dass der Standort das QS-GAP Audit nicht bestanden hat, Zustand 14 ist für einen QM-Milch zertifizierten Standort vorgesehen und in Zustand 15 kommen alle von QS zugelassenen GlobalGap Standorte (nachdem von QS das vorher vom Bündler erstellte Bündel zugelassen wurde).

Die neuen Zustände 13-15 sind bereits fest in der Hauptanwendung verankert und von daher müssen sie nun auch über die Schnittstelle so übermittelt werden. Sie sind aber bisher nur serverseitig implementiert worden, es wurde also nicht die WSDL Datei dafür angepasst. Dies bedeutet, dass der Server in der Lage ist, diese Stati zu verarbeiten und auch zurückzusenden; ob der Client diese neuen Stati versteht und damit umgehen kann, hängt sehr stark davon ab, wie die diesbezügliche Implementierung aussieht und wie dort mit diesem neuem Wert umgegangen wird.

Allgemeine Vorgehensweise

Die Exception 100 = "Error: please contact support@qs-platform.info" wird immer dann geworfen, wenn eine sonstige Exception nicht passt, insbesondere wenn z.B.

beim Speichern ein Datenbankfehler aufgetreten sein sollte oder aber eine andere unerwartete Exception vom System selber geworfen wird, die nicht anderweitig ausgewertet und gemappt werden kann.

Die Ausnahme 016, "No valid location number" wird geworfen, wenn eine fehlerhafte LocationID oder LocationType eingegeben worden ist. Dies bedeutet hier nur, dass diese Werte nicht übermittelt worden sind, also leer = null waren bzw. nur Leerzeichen enthielten. Falls ein nicht gültiger Wert als String übermittelt wurde (z.B. eine Zahl mit Buchstaben oder eine Zahl, die nicht im Wertebereich definiert worden ist), wird dies bereits vom Webservice Framework abgefangen und ein org.xml.sax.SAXException: Bad types wird zurückgesendet.

Demgegenüber wird die Ausnahme 053, "No such location" geworfen, falls es den Standort gar nicht gibt und er nicht mit diesen Parametern in der Datenbank gefunden werden konnte.

An dieser Stelle sei noch mal angemerkt, dass bei jedweder Exception alle bisherigen Aktionen auf der Datenbank rückgängig gemacht werden (Rollback) und keine weiteren Daten ausgewertet werden. Es werden also auch keine Daten neu angelegt, gespeichert oder verändert, sondern es wird nur die entsprechende Fehlermeldung zurückgesendet.

Funktion 6.1 CQSSRegisterMemberMsg:

Als erstes überprüft, ob unter den übermittelten Daten für LocationNumber und LocationType ein Standort existiert. Wird ein solcher Standort gefunden, erfolgt an dieser Stelle (abweichend vom sonst üblichen Procedere) KEINE Überprüfung, ob dieser Standort dem Bündler gehört (da ein Bündler auf diesem Wege einen abgemeldeten, fremden Standort für sich wieder anmelden kann) und ob der Standort abgemeldet ist.

Sollte er abgemeldet sein, wird überprüft, ob es zu einer Produktionsarten Kollision kommen würde, wenn dieser Standort angemeldet wird (was zu einer „023“, "Data collision: location address is already in database“ führen würde, da zur selben Zeit nicht sich überschneidende Produktionsarten angemeldet sein dürfen).

Jetzt wird für diesen abgemeldeten Standort überprüft, ob er bereits dem Bündler zugeordnet ist oder bisher bei einem anderen Bündler war. Sollte er bei einem anderen Bündler gewesen sein, so wird seine Zugehörigkeit zu diesem Bündler beendet und er wird diesem Bündler zugeordnet, sowie angemeldet. Sollte der Standort aber dem Bündler bereits gehören, so wird auf diesem Standort die Transition „Anmelden“ durchgeführt, um den Standort wieder in den Zustand „angemeldet“ zu bringen. Sollte dies nicht möglich sein, wird mit 100, "Error: please contact support@qs-platform.info geantwortet. In diesem Fall wird kein Update der Daten durchgeführt oder Werte verändert, es erfolgt nur die Anmeldung des Standorts.

Sollte der angemeldet gewesen sein, geschieht gar nichts und es wird nur der aktuelle Status, sowie die Client ID zurückgegeben.

Falls der Standort nun (um-) angemeldet worden ist bzw. bereits vorher schon angemeldet gewesen war, kann es nun sein, dass ein Zertifizierungssystemwechsel ansteht. Wenn also die Funktion 6.25 aufgerufen worden ist und in ihr der Wert für das AuditSchema auf ungleich dem QM-System des Standortes ist, so wechselt er mit diesem Aufruf nach in das angegebene QM-System. Bei diesem Wechsel wird aber als erstes überprüft, ob dieser Standort eine Produktionsart hat, die zum angegebenen QM-System passt. Ist dies nicht der Fall so wird eine Exception 034, "Production line does not mach with stored values" geworfen.

Ist für das neue QM-System eine Zertifizierungstelle notwendig, so muss diese in der Location unter CeritificationBody gesetzt sein, so dass eine Zertstelle unter dieser QSID gefunden werden kann. Ist dieses Feld also leer, kann kein Wechsel erfolgen und er wird die Exception „043“, "No such certication body" geworfen.

Falls der Standort NICHT gefunden werden konnte, so muss er zusammen mit dem Systemteilnehmer/Unternehmen neu angelegt werden. Dazu werden aus den Adresdaten ein neuer Systemteilnehmer angelegt. Dabei muss aber PostCode, Town, Street und Lastname mit Text gefüllt sein, da dies Pflichtfelder sind. Falls eines der Felder fehlen sollte, wird die Fehlermeldung 003, "Address information is ambiguous“ geworfen. Falls das Land fehlen sollte wird mit 058, "No country code given" reagiert. Der Name wird aus dem locationAddress.Name bestimmt. Ein etwaiger, dort angegebener FirstName wird mit einem führenden Leerzeichen an den LastNamen hinten dran gehangen, um so den Namen des Unternehmens fest zulegen (dies geschieht aus Konformitätsgründen zur früheren Anwendung). Für diesen Systemteilnehmer wird dann eine QSID generiert, welche auch am Ende zurückgegeben wird. Da das Feld Bundesland nun ebenfalls als ein Pflichtfeld deklariert worden ist, wenn als Land Deutschland ausgewählt wurde, dieses Bundesland aber nicht direkt in der Schnittstelle übertragen werden kann, wird in einem ersten Schritt hier vom System das Bundesland „Deutschland“ eingetragen. (Später wird dann aus der Postleitzahl das Bundesland errechnet bzw. es muss dann irgendwann über die Schnittstelle mitgeliefert werden).

Die beim Systemteilnehmer zu setzende QS Stufe ergibt sich zwingend aus der jeweiligen QS-Stufe des bündelnden Bündlers. D.h. beim Bündler Landwirtschaft ist das die Stufe „3“(Landwirt)

Falls eine RepresentativAddress vergeben worden ist, wird diese für den Anwender verwendet, der dem Systemteilnehmer als gesetzlicher Vertreter zugeordnet werden soll. Ansonsten wird die Adresse des Systemteilnehmers darauf übertragen. Falls auch diese leer sein sollte, wird auf die Adresse ausgewichen, die vielleicht in der Location übermittelt worden ist.

Es wird für diesen Systemteilnehmer nun ein neu Standort angelegt.

Da die Zertstelle ein Pflichtfeld ist, muss deren QSID übermittelt und diese Zertstelle im System auch so gefunden werden: ansonsten wird ein 043, "No such certification body" Fehler erzeugt und zurückgegeben.

Ebenfalls zwingend notwendig ist bei einem Standort der locationType, wobei nur die locationType ausgewählt werden dürfen, die für die aktuelle QS-Stufe gültig sind, sonst wird eine Exception des Typs 033, "No such production line" geworfen.

Wie es bereits beim Systemteilnehmer beschrieben worden ist sind auch im Standort dieselben Felder PostCode, Town, Street und Lastname Pflichtfelder und müssen übermittelt werden, ansonsten wird eine entsprechende Exception versendet. Auch hier wird aus dem übermittelten Last und FirstName ein Standortname erzeugt.

Auch muss immer eine LocationNumber übergeben werden (sollte sie nicht übermittelt worden sein, wird eine 057, "No location number given" geworfen). Es wird dann überprüft, ob die Kombination aus locationType und LocationNumber sich nicht mit bestehenden Betriebsarten überschneidet, die für diese Standortnummer bereits angemeldet sind. Sollte bereits eine der Grundarten angemeldet sein, so wird die Exception 023, "Data collision: location address is already in database" geworfen (aus Kompatibilitätsgründen zur alten Version des WebServices wurde diese Exception gewählt).

Auch hier wird wie für den Systemteilnehmer das Bundesland mit Deutschland automatisch vorbelegt, wenn als Land Deutschland ausgewählt wurde.

Falls in dem Feld location ein Representativ mitgesendet wurde, werden seine Daten verwendet, um den Ansprechpartner des Standorts damit zu befüllen.

Bzgl. der Kommunikationsdaten (Fax, eMail, Telephon und Mobil) werden keine Exceptions geworfen, da sie nicht verpflichtend sind. An dieser Stelle sei nur angemerkt, dass an den Systemteilnehmer die Daten aus dem obersten Com Element angefügt werden und falls dieses leer ist, die Daten aus dem Com Element der location. An den Anwender des Standorts werden nur die Daten des location.com – Elements gepackt.

Sollte als Standortnummer eine Nummer gewählt werden, die mit „528528“ beginnt, so bedeutet dies, dass der Bündler den Standort für das System IKB anmelden möchte. Dies ist aber nur möglich, wenn auch die gewählte Produktionsart im 2000er bereich liegt. Ansonsten wird die Exception „033“, "No such production line" geworfen und es erfolgt keine Anmeldung, da IKB nur für diese System gültig ist.

Falls beim Speichern des Systemteilnehmers oder des Standorts festgestellt wird, dass dessen Adresse bereits in der DB enthalten ist (also eine Dublette der Adresse dann vorliegen würde), so wird eine Exception 023, "Data collision: location address is already in database" geworfen.

Als Rückgabewert wird das aktuelle Datum und die generierte ID des Systemteilnehmers angefügt (also nicht die Standortnummer).

Funktion 6.2 CQSSAddLocationMsg:

Bei dieser Funktion muss der Systemteilnehmer bereits angelegt worden sein (also im System existieren), ansonsten wird eine Fehlermeldung 031, "No such participants" generiert.

Auch bei der Suche nach dem Standort wird dieser gefundene Systemteilnehmer mit einbezogen und nur seine Standorte können auf diesem Wege wieder angemeldet werden, wie oben beschrieben. (Da hier die Funktionalität ausgelagert worden ist, kann also auch auf diesem Wege ein Bündlerwechsel und eine EurepGap bzw. QS ummeldung erfolgen, wie es oben beschrieben worden ist).

Falls kein Standort angemeldet werden soll, sondern diesem Systemteilnehmer ein neuer Standort hinzugefügt werden soll, so verhält sich die Anwendung von hier an wie bereits unter 6.1 beschrieben (wobei natürlich kein Systemteilnehmer und kein Anwender für diesen Systemteilnehmer erzeugt werden).

Bei der Rückantwort an den Aufrufenden wird sowohl der aktuelle Status des Standorts, in dem er sich nun befindet, und seine Standortnummer, als auch die ID des Systemteilnehmers mit dem Timestamp zurückgeliefert.

Funktion 6.25 CQSSRegisterCertificationSystemMsg und 6.26 CQSSAddCertificationSystemMsg:

Diese Funktionen unterscheiden sich in ihrer Funktionalität nicht von den entsprechenden Funktionen 6.1 und 6.2. Zusätzlich kann hier noch ein Zertifizierungssystem angegeben werden.

Mögliche Zertifizierungssysteme sind:

- 1 QS
- 2 QS-Gap
- 3 QM-Milch
- 4 IKB
- 5 GlobalGap
- 6 Codiplan
- 7 IKKB

Für manche Zertifizierungssysteme, muss auch für AccreditedUntil ein Wert gesetzt sein, der nicht in der Vergangenheit liegt (sonst wird eine 045, "Date collision" Exception abgesetzt) und auch in CertNr muss eventuell ein Text eingetragen sein (sonst erfolgt eine 081, "No distinct certification body found" Exception). Für GlobalGap muss zusätzlich eine Liste der zertifizierten Kulturen übertragen werden.

Sollte die aktuelle Produktionsart NICHT zum angegebenen Zertifizierungssystem passen, wird die Fehlermeldung 034, "Production line does not mach with stored values" geworfen.

Funktion 6.27 QSChangeAttributeMsg:

Diese Funktion wird nicht weiter unterstützt.

Funktion 6.3 QSExtLocationQuery und QSExtLocationStandardQuery:

Input: Produktionsart, QS-Standortnummer

Die Funktionen QSExtLocationStandardQuery (1) und QSExtLocationQuery (2) unterscheiden sich im Format der übergebenen Parameter. Bei (1) wird die Produktionsart als Integer und die QS-Standortnummer als String übergeben werden. Bei (2) wird die Produktionsart als `CS840018Type` und die QS-Standortnummer `AN15Type` als übergeben.

Die Funktionalitäten der Funktionen QSExtLocationStandardQuery und QSExtLocationQuery sind identisch. Diese Funktion ist sowohl verschlüsselt über HTTPS mit Authentifizierung, als auch unter dem HTTP Protokoll ohne Authentifizierung durch einen anderen Service verfügbar, wobei aber dieselbe Funktionalität hinterlegt ist. Demnach gibt es bei dieser Funktion keine Exception 70 = „No permission granted“, da hier die öffentliche Suche nachimplementiert wurde. Diese Funktion erwartet, dass eigentlich nur Grundarten abgefragt werden. Der Modulo 1000 Rest der übermittelten BtartId darf also nur eine echte 2er Potenz sein und muss kleiner als 256 sein. Falls aber mehrere Grundarten gleichzeitig abgefragt werden, weil eine Zahl abgefragt wird, die aus mehrere 2er Potenzen besteht, wird genau diese Produktionsart überprüft. Ob also für genau diese Produktionsart eine Lieferberechtigung vorliegt. Es erfolgt in diesem Fall also keine Aufspaltung in die Grundarten, sondern es würde für diese gegebene Produktionsart ausgewerten, ob ein Standort dafür die Lieferberechtigung besitzt.

Bei einem leerer Text, der für die BtartId oder die LocationID übermittelt wird, wird als Antwort sofort der Wert 0 und ein „No“ zurückgeliefert, ohne dass eine weitere Auswertung erfolgt. Es erfolgt also in dieser Funktion nicht das sonst übliche werfen einer entsprechenden Exception, sondern es wird ein einfaches „No“ zurückgegeben.

Bei der Suche nach Standorten bzgl. der übergebenen LocationID werden aber nur angemeldete Standorte beachtet, da abgemeldete Standorte nicht zu berücksichtigen sind. Falls kein Standort gefunden wird, wird die oben erwähnte Meldung „No“ zurückgegeben.

Sollte aber mindestens ein Standort gefunden worden sein, werden all diese Standorte der Reihe nach durchgegangen. Sollte seine 1000-Stelle nicht mit der ange-

fragten Produktionsart übereinstimmen, wird sofort zum nächsten Standort übergangen. Anschließend wird überprüft, ob es eine geprüfte Produktionsart gibt und diese von der gemeldeten Produktionsart abweicht. Ist dies der Fall, so wird die Schnittmenge aus der geprüften und der gemeldeten Produktionsart berechnet und dieser Wert wird dann anschließend als auszuwertende Produktionsart weiterverwendet. Ansonsten wird die gemeldete Produktionsart im weiteren Verlauf ausgewertet.

Wenn die angefragte Grundproduktionsart in der auszuwertenden Produktionsart des aktuellen Standorts nicht vorkommt, so wird der nächste Standort untersucht. Sollte auf diese eben beschriebene Art und Weise kein Standort gefunden werden, dessen Produktionsart mit der angefragten übereinstimmt, so wird ein „No“ zurückgeliefert.

Falls aber die angefragte Produktionsart doch in der aktuellen Standort-Produktionsart enthalten sein sollte, so ist damit der Standort gefunden, dessen Status nun ausgewertet werden muss.

Bei der Auswertung des Status werden sowohl der Lieferstatus, als auch der Sperrstatus ausgewertet und das Ergebnis wird gesetzt, wobei dabei auch das Salmonellen Monitoring in der Antwort entsprechend berücksichtigt wird.

Eine weitere Implementierung der `QSExtLocationStandardQuery` liefert zusätzlich in der Antwort ein Kennzeichen zur Freilandhaltung bei den Produktionsarten 2001 und 1002 zurück (`QSExtLocationFreeRangeQueryReply`). Diese Funktionalität ist nur in der Serviceversion ohne Authentifizierung vorhanden.

Funktion 6.4 QSExtLocationBtartQuery und QSExtLocationBtartStandardQuery:

Input: Produktionsart, QS-Standortnummer

Die Funktionen `QSExtLocationBtartStandardQuery` (1) und `QSExtLocationBtartQuery` (2) unterscheiden sich im Format der übergebenen Parameter. Bei (1) wird die Produktionsart als Integer und die QS-Standortnummer als String übergeben werden. Bei (2) wird die Produktionsart als `CS840018Type` und die QS-Standortnummer `AN15Type` als übergeben.

Die Funktionalitäten der Funktionen `QSExtLocationBtartStandardQuery` und `QSExtLocationBtartQuery` sind identisch. Für diese Funktion gilt dieselbe Einleitung wie für die Funktion 6.3. Im Unterschied zu 6.3 wird hier aber nicht nur eine einzige Grundart zugelassen, sondern im Ergebnisreply werden alle in der übermittelten Produktionsart enthaltenen Grundarten überprüft und für jede dieser Grundarten wird das jeweilige Ergebnis übermittelt. Demnach wird z.B. bei der Übermittlung von 1005 als `btartID`, sowohl 1004, als auch 1001 überprüft, während bei der 1015 jede Grundarten bis zur 8 enthalten ist.

Falls keine BtarID oder keine LocationID übermittelt werden, wird ein leerer Reply zurückgesendet, der keine Daten enthält, außer den standard Werten (also wieder keine Exception).

Auch hier werden wieder nur angemeldete Standorte berücksichtigt, abgemeldete werden nicht in die Überprüfung aufgenommen.

Falls kein Standort gefunden wurde, wird analog wie zu 6.3 für alle abgefragten Grundproduktionsarten ein „no“-Result zurückgeliefert.

Es werden wieder alle gefundenen Standorte überprüft, wobei Standorte mit einer anderen Produktionsgruppe im 1000er Bereich direkt übergangen werden.

Auch wird hier, wie in der anderen Funktion, die auszuwertende Produktionsartgrundmenge aus der Schnittmenge zwischen der geprüften und der gemeldeten Produktionsartgrundmenge berechnet.

Falls für den aktuellen Standort ein Status ungleich 0 gefunden wird, muss dieser Status in die Antwort für die abgefragte Grundartenmenge übertragen werden und zwar für alle Grundarten, die sowohl in der abgefragten, als auch in der auszuwertende Produktionsartgrundmenge sind.

Wenn am Ende noch abgefragte Grundarten nicht auf diese Art von den Produktionsgrundarten der Standorte abgedeckt worden sein sollten, wird für diese verbleibenden Grundarten die Antwort „no“ erzeugt, da kein Standort gefunden werden konnte, der für diese Grundart einen Status ungleich 0 hatte.

Funktion 6.5 QSBdlRequestMsg:

Bei einer -1 als BundleNr werden nur angemeldete Standorte gezeigt, die diesem Bündler aktuelle zugeordnet und die außerdem keinem Bündel zugeordnet sind.

Bei der Zahl -2 als BundleNr erfolgt eine Abweichung von der bisherigen Implementierung, da QS nicht mehr nur noch ein einziges Tempbündel festgelegt hat, sondern für die Landwirtschaft drei verschiedene Tempbündel vorgesehen hat. Deswegen werden jetzt bei einer Zahl kleiner als -1 die verschiedenen Tempbündel des Bündlers angezeigt, die für diesen Bündler aktuelle erstellt worden sind. Dabei wird bei -2 das Tier-Tempbündel, bei -3 das OGK-Tempbündel und bei -4 das Ackerbau-Tempbündel zurückgeliefert.

Wenn eine BundleNr übermittelt worden ist, so wird nach diesem Bündel gesucht. Ist bei diesem Bündler kein Bündel mit dieser Nummer vorhanden, wird 007, "Bundle could not be retrieved" zurückgegeben. Ansonsten werden alle Standorte gesucht, die in diesem Bündel und die diesem Bündler zugeordnet sind, sowie aktuell angemeldet sind.

In beiden Fällen wird für den Standort jeweils die LocationID = Standortnummer und die BtartID = Produktionsart, sowie der Status zurückgegeben.

Funktion 6.6 QSBdIStortInfoMsg:

Sowohl LocationID als auch BtartId müssen übermittelt werden, sonst wird eine 016, "No valid location number abgesetzt. Sollte mit diesen Angaben kein Standort gefunden werden, wird dies mit der Exception 053, "No such location" quittiert. Falls dieser Standort nicht dem Bündler gehört, wird zur Kenntlichmachung die Exception 070", "No permission granted verwendet.

Die Daten für die Adresse (Street, Town, Country, Lastname, ZipCode) werden aus dem Standort selber genommen (nicht vom Systemteilnehmer). Um die Kommunikationsdaten zu bestimmen, wird zuerst versucht den Anwender des Standorts zu finden und ansonsten wird der Anwender des zugehörigen Systemteilnehmer genommen.

Funktion 6.7 QSUpdateLocationMsg:

Wenn sowohl die alten, als auch die neuen Daten bzgl. der LocationID und BtartId leer sein sollten, wird die Exception 016, "No valid location number" geworfen.

Auch hier wird wie beim Einfügen eines neuen Standorts geprüft, ob die neue Kombination aus LocationId = Standortnummer und BtartId = Produktionsart eine Überschneidung mit bereits bestehenden Standorten erzeugen würde, was in diesem Fall dann die Fehlermeldung 023, "Data collision: location address is already in database bedeuten würde.

Wird als Standortnummer eine Nummer gewählt die mit 528528 anfängt bzw. eine von dieser Start-Kennnummer verschiedene, wobei der Standort früher eine 528528 hatte, so bedeutet dies einen Wechsel nach IKB bzw. im zweiten Fall von IKB nach QS. Damit dieser Wechsel klappt muss aber die Produktionsart für IKB gültig sein, da sonst ein 034, "Production line does not mach with stored values" zurückgegeben werden muss, um dies zu signalisieren. Auch hier kann es bei einem Wechsel zurück nach QS (wenn also eine Nummer eingegeben wird, die nicht 528528 entspricht und er aktuell im IKB ist) die Zertstelle „verloren“ gehen durch den wechsel, weswegen sie standardmäßig anschließend auf eine Dummy-Zertstelle gesetzt wird. Deswegen sollte im Vorfeld bzw. im Nachhinein die Zertstelle entsprechend vorbereitet bzw. nachgezogen werden (siehe auch bei EurepGap).

Funktion 6.8 QSUpdateBtartMsg:

Das was bzgl. 6.7 gesagt wurde, kann hier analog auf diese Funktion übertragen werden, wobei hier aber die Produktionsart gewechselt wird. Deswegen ist hier noch ein zusätzlicher Test eingebaut, ob diese Produktionsart mit der QS-Stufe des Systemteilnehmers des Standorts verträglich ist (im Fehlerfall: 033, "No such production line).

Funktion 6.9 QSUpdateCertBodyMsg:

Hier wird zuerst überprüft, ob es überhaupt so eine Zertifizierungsstelle gibt, die mit dieser CertificationBody = QSId gemeldet ist. Falls der CertificationBody leer sein sollte wird eine 014, "No valid system member geworfen, falls keine Zertstelle gefunden wird, der Fehler 043, "No such certication body". (Es werden also dann auch keine Änderung durchgeführt!) Ansonsten wird genauso verfahren wie bei den anderen Update Funktionen und die Daten werden entsprechend angepaßt.

Diese Funktion ist ins Besondere beim Wechseln von einem Fremdsystem = EurepGap oder IKB zurück nach QS wichtig, da der Standort seine alte Zertstelle verliert, wenn er zu diesen Fremdsystem wechselt. Daher muss bei einem wechsel zurück entweder vorher eine gültige Zertstelle mit dieser Funktion gesetzt werden oder aber dies muss im Nachhinein erfolgen.

Funktion 6.10 CQSSUpdateLocAdrMsg:

Hierbei werden die einzelnen Felder optional behandelt, d.h. wenn ihre Daten nicht übermittelt werden, so werden sie auch nicht geupdated. Abweichend von der Dokumentation der Schnittstelle, wurde hier auf das Verfahren der SLA umgeschwenkt und es ist NICHT mehr zwingend notwendig, dass die Daten von Old Value und den Daten aus der Datenbank übereinstimmen (diese Praxis musste eingeführt werden, da die Altdaten bzgl. der Standorte nicht zwingend bekannt waren oder nicht mehr rekonstruiert werden konnten). Demnach ist es nicht mehr zwingend erforderlich, dass für alle gesetzten Felder sichergestellt sein muss, dass der zugehörige Old Value ebenfalls gesetzt ist und dass dieser Wert mit dem aktuellen Wert in der Datenbank übereinstimmt. OldValue wird in dieser Version daher überhaupt nicht ausgewertet.

Es ist aber nicht möglich gar kein Objekt im NewValue zu übertragen, also dieses Feld auf Null zu setzen, da dann ein Fehler 005, "Address could not be retrieved" erzeugt wird.

Jetzt werden alle gesetzten Felder innerhalb von NewValue entsprechend auf den zugehörigen Wert im Standort in der Datenbank übertragen. Wenn ein Feld den Wert Null hat, so wird nichts gemacht und es wird zum nächsten Feld weitegegangen. Um einen Eintrag zu löschen, was eigentlich nicht möglich ist, da es ja meistens Pflichtfelder sind, kann ein Leer-String (=ein String mit einem Leerzeichen) übertragen werden, welches dann übernommen wird.

Der hier verwendete PostCode wurde auf die Postleitzahl des Standortes gemappt (auf die Postleitzahl des Postfachs kann so nicht zugegriffen werden).

Der Name des Standorts wird dabei aus dem übermittelten First- und LastName konstruiert, indem zuerst der LastName, dann ein Leerzeichen und dann der FirstName hinter einander gesetzt werden.

Funktion 6.11 CQSSUpdateLocComMsg:

Da Kommunikationsdaten nicht im Standort, sondern nur in einem Anwenderobjekt gespeichert werden können, wird ein Standortanwender gesucht. Falls es diesen nicht geben sollte, wird ein neuer Anwender angelegt und auf diesen werden dann die übermittelten Daten übertragen.

Auch hier wurde, wie in Funktion 6.11 mit derselben Begründung, eine Abweichung von der Beschreibung dieser Funktionalität vorgenommen: OldValue wird also auch in dieser Version daher überhaupt nicht ausgewertet.

Ebenfalls ist es hier auch nicht notwendig alle Felder zu setzen; nur Felder, die einen Wert, der von NULL abweicht, werden auf das entsprechende Feld in der Datenbank übertragen.

Funktion 6.12 QSDeleteMemberMsg

Falls der Systemteilnehmer über die ClientID nicht gefunden werden konnte, so wird die Fehlermeldung 031, "No such participants" abgesetzt. Auch hier darf nur auf Systemteilnehmer zugegriffen werden, die diesem Bündler zugeordnet sind (sonst 070, "No permission granted). Wenn der Systemteilnehmer gefunden wurde, wird bei jedem zugeordneten Standort die Transition „Standort Abmelden“ ausgeführt und anschließend gespeichert, bevor zum nächsten Standort übergegangen wird.

Funktion 6.13 QSDeleteLocationMsg

Bei dieser Funktion geschieht das Gleiche, was auch sonst geschieht, wenn ein Standort anhand seiner Daten gesucht wird. Anschließend wird wie bei der Funktion 6.12 die Transition „Standort abmelden“ für diesen Standort durchgeführt und anschließend gespeichert.

Generelles zu den nachfolgenden Auditbericht-Funktionen 6.14 – 6.18:

Bündler können nur Auditberichte von Standorten abrufen können, die ihnen aktuell zugeordnet sind. Für Standorte, die zu einem Zeitpunkt in der Vergangenheit dem Bündler zugeordnet waren, ist dies nicht mehr möglich!

Funktion 6.14 QSLastInspectionMsg

Wenn das Datum DateFrom und DateTo nicht gesetzt sind bzw. in der Zukunft liegen oder das From Datum nach dem To Datum liegen wird sofort die Fehlermeldung 045", "Date collision" geworfen. DateFrom und DateTo beziehen sich in dieser Abfrage auf das Datum an dem die Auditierung stattgefunden hat.

Bei der Suche nach den Auditberichten werden aber nur Berichte berücksichtigt, die sowohl vom verantwortlichen Auditor freigegeben, als auch unter Umständen von QS entsperrt worden sind und die somit akzeptiert und gültig sind. Gesperrte oder nicht freigegebene Berichte können auf diese Weise nicht angezeigt werden.

Es sei hier aber darauf hingewiesen, dass für jeden Standort immer nur ein einziger und zwar der aktuellste Prüfbericht zurückgegeben wird, dessen Auditierung im abgefragten Zeitraum lag. Das heißt also: Wenn ein Standort keinen Auditbericht im Zeitraum hatte, wird er nicht berücksichtigt, bei mehreren Berichten, wird der neueste ausgewählt und die älteren übergangen.

Funktion 6.15 QSLastSentInspectionMsg

Diese Funktion unterscheidet sich von der vorherigen 6.1.4 nur darin, dass anstelle des Datums für die Einschränkung der Abfrage nicht der Auditianfang, sondern das Datum des Versands des Mängelberichts verwendet wird.

Ansonsten wird das Ergebnis auf die gleiche Art und Weise erzeugt und es werden dieselben Fehler geworfen.

Funktion 6.16 QSExtInspectionMsg

Als erstes wird nach dem üblichen Verfahren der Standort gesucht und versucht dem Bündler zuzuordnen, mit den Möglichkeiten die bekannten Exceptions dabei zu werfen.

Für diesen so gefundenen Standort wird dann versucht einen Auditbericht, der an dem angegebenen Tag der Inspection angefangen wurde, zu finden. Ist dies nicht möglich so wird der letzte verfügbare Auditbericht gesucht. Falls auch dann kein Auditbericht gefunden werden kann, wird die Exception 050, "No such verification report" zurückgegeben. In beiden Abfragen werden aber nur akzeptierte Berichte (siehe oben) gesucht.

Funktion 6.17 QSInspectionFromIdMsg

Bei dieser Funktion wird eine 050, "No such verification report" geworfen, wenn eine leere InspectionId übermittelt worden ist oder aber kein Auditbericht für diese Nummer gefunden bzw. geladen werden konnte. Bei der Abfrage wird zur Sicherheit noch die BündlerID = QuestionerID mit in den Filter aufgenommen, um sicherzustellen,

dass nur Berichte abgefragt werden können, die auch tatsächlich von Standorten stammen, die zu diesem Bündler gehören. Des Weiteren können nur akzeptierte Berichte auf diese Weise erfragt werden.

Funktion 6.18 QSExtlInspectionsInPeriodMsg

Diese Funktion hat am Anfang dieselben Datums-Exceptions und Standort-nicht-gefunden bzw. keine-Zugriffsberechtigungs Fehler, wie es an dieser Stelle erwartet wird und bereits mehrfach beschrieben worden ist.

Bei der Füllung des Filters wird wieder das Datum des Auditanfangs als Bezugsdatum gewählt. Auch können wieder nur akzeptierte Berichte abgefragt werden.

Die Auditberichte werden aufsteigend nach diesem Datum sortiert jeweils nacheinander so in das Ergebnis gepackt, wie es bereits in den vorherigen Funktionen auch geschehen ist.

Falls während des Erstellens einer Antwort für einen der Auditbericht aus der Liste ein Fehler auftaucht, so wird dieser abgefangen und dieser Auditbericht wird nicht zur Ergebnisantwort hinzugepackt, sondern es wird beim nächsten Bericht weitergemacht.

Funktion 6.19 QSLocationInfoMsg

Nachdem der Standort auf dem üblichen Weg gefunden worden ist (wobei diesmal auch abgemeldete Standorte gesucht werden und es im Filter keine Einschränkung auf diesen Bündler gab), wird überprüft, ob dieser Standort aktuell vom anfragenden Bündler gebündelt wird. Ist dies nicht der Fall, so wird `ownLocation` auf `false` gesetzt und das `signOffDate` eingetragen (falls es aktuell gesetzt sein sollte). Anschließend wird diese Information sofort als Antwort zurückgesendet und es erfolgt keine weitere Auswertung.

Falls der Standort dem Bündler gehört so wird `ownLocation` auf `true` gesetzt. Falls sich der Standort in einem Bündel befindet, so wird dessen Name ermittelt und mit übermittelt.

Abschließend wird der Status des Standortes eingetragen und gegebenenfalls noch sein Abmeldedatum, falls dieses gesetzt sein sollte.

Funktion 6.20 QSBuildBundleMsg

In dieser Funktion gibt es Abweichungen zur Beschreibung, denn ein landwirtschaftlicher Bündler kann nur noch EurepGap Standorte selber in einem Bündel bündeln. Es werden also nur noch Standorte für die Bündelung zugelassen, die dem Zertifizie-

zungsschema EurepGap zugeordnet sind und die noch bisher keinem Bündel zugeordnet sind.

Der Name wird entsprechend der Vorgabe gesetzt, aber die Bündelnummer wird vom System vergeben: Dabei werden die Nummern aufsteigend von 1 bzgl. eines Bündlers vergeben.

Um die Standorte dem Bündel zuzuordnen werden sie jeweils einzeln anhand ihrer QSStandortnummer, Betriebsart gesucht, wobei aber nur angemeldete Standorte auf diese Weise gefunden werden können. Konnte der Standort auf diese Weise nicht gefunden werden, wird dieser Standort mit dem Fehlercode 53, „No such location“ in die Fehlerliste übertragen. Sollte dieser Standort aktuell nicht dem Zertifizierungsschema EurepGap zugeordnet sein, so wird als die am meisten passende Fehlermeldung 34, „Production line does not mach with stored values“ ausgewählt. Falls der Standort schon gebündelt sein sollte, wird für diesen Standort die Fehlermeldung 63, "Location is already bundled" der Fehlerliste angefügt. Falls ein Standort ausgewählt worden sein sollte, der nicht diesem Bündler zugeordnet ist, wird die dafür übliche Fehlermeldung 70, "No permission granted" diesem Standort zugefügt.

Falls der Standort durch diese Tests durchkommt, wird ihm dieses Bündel zugeordnet und der nächste Standort wird nach demselben Verfahren versucht diesem Bündel zuzuordnen.

Funktion 6.21 CQSSUpdateCompAdrMsg:

Bei dieser Funktion wird im Prinzip das Gleiche gemacht wie bei ihrem Pendant für den Standort, nur dass die Daten nicht im Standort, sondern in den Systemteilnehmer eingetragen werden. Natürlich wird hier eine Exception 031, "No such participants" geworfen, wenn die ClientID nicht gesetzt war bzw. kein Systemteilnehmer gefunden werden konnte.

Auch hier wird der Name des Unternehmens aus LastName + (Leerzeichen) + First-Name zusammengesetzt.

Funktion 6.22 CQSSUpdateCompComMsg:

Diese Funktion ähnelt der entsprechenden des Standorts, nur dass diesmal als Anwenderobjekt nicht der Standortanwender, sondern der des Systemteilnehmer verwendet wird. Die restliche Funktionalität ist genau analog implementiert.

Funktion 6.23 QSChangeLocationDataMsg:

Nach dem erfolgreichen Laden des Standorts wird der übermittelte Wert für Groupflag direkt auf den Standort übertragen. Sollte bei dieser Funktion aber ein leeres Feld (also der Wert Null) für die Groupflag übertragen werden, so bedeutet dies in diesem Fall, dass der Wert für das Groupflag gelöscht werden soll.

Funktion 6.24 QSCertBodyQueryMsg:

Dies ist eine standart Select Abfrage, die wie die anderen funktioniert: Erst den Standort suchen. Prüfen ob auf diesen Standort zugegriffen werden darf, die benötigte Information (hier die QS-Id der Zertifizierungsstelle) heraussuchen und eine Antwort erstellen. Falls es dabei Probleme gibt, werden die bereits vorher beschriebenen Exceptions an den jeweiligen Stellen zurückgegeben.

Funktion 6.28 QSUpdateCertificationSystemMsg:

Diese Funktion ersetzt die Funktion 6.27.

Mit dieser Funktion kann das Zertifizierungssystem eines bestehenden Standortes geändert oder angepasst werden.

Auch hier sind folgende Zertifizierungssysteme möglich:

- 1 QS
- 2 QS-Gap
- 3 QM-Milch
- 4 IKB
- 5 GlobalGap
- 6 Codiplan
- 7 IKKB

Je nach Zertifizierungssystem müssen noch die Zertifikatslaufzeit und Zertifikatsnummer angegeben werden. Für GlobalGap muss zusätzlich die Liste der zertifizierten Kulturen angegeben werden. Bei einem Systemwechsel kann es auch nötig sein, das eine Produktionsart und Zertifizierungsstelle übermittelt werden muss. Dies ist zum Beispiel bei dem Wechsel von QM-Milch nach QS notwendig, da die Produktionsart 1320 ausschliesslich für QM-Milch gültig ist. Für das Zertifizierungssystem irrelevante, übermittelte Daten werden ignoriert.

Wenn das angegebene Zertifizierungssystem mit dem Zertifizierungssystem im Standort überein stimmt, so werden nur einzelne, übermittelte Parameter angepasst. Dies kann die Zertifikatslaufzeit wie auch die Zertifikatsnummer sein. Bei GlobalGap kann hierüber die Liste der zertifizierten Kulturen aktualisiert werden.