



Spezifikation der SOAP-WebService-Schnittstelle für das QS-Futtermittelmonitoring

Autor	Reza Eslami
Angelegt	03.06.2008
Revision	1.0
Stand	02.11.2009 14:21
Status	Freigabe

Dokument Versionen			
Version	Autor	Beschreibung der Version	Datum
1.00	Reza Eslami	Endversion für die erste Vorstellung der Spezifikation	6.1.2008

Inhaltsverzeichnis:

1	Fachlicher Rahmen	7
2	Unterteilung der Futtermittel-Schnittstellen in eine Auftraggeber und eine Labor-Schnittstelle:	9
3	Allgemeine Richtlinie der Schnittstelle	9
4	Schnittstelle für Auftraggeber (Systemteilnehmer, Bündler, Probennahmestitution)	9
4.1	Probenliste herunterladen: (<i>getSampleList</i>):	9
4.1.1	Input	9
4.1.2	Output	10
4.1.3	Vorgehensweise	10
4.2	Probe herunterladen (<i>getSample</i>):	10
4.2.1	Input	10
4.2.2	Output	10
4.2.3	Vorgehensweise	12
4.3	Probe hochladen: (<i>setSample</i>):	12
4.3.1	Input	12
4.3.2	Output	12
4.3.3	Vorgehensweise	12
4.4	Probe Daten bestätigen (<i>confirmSampleData</i>)	13
4.4.1	Input	13
4.4.2	Output	13
4.4.3	Vorgehensweise	13
4.5	Zustand ändern (<i>changeSampleState</i>)	13
4.5.1	Input	13
4.5.2	Output	14
4.5.3	Vorgehensweise	14
4.6	Analyseergebnisse einer Probe herunterladen (<i>getAnalysis</i>):	14
4.6.1	Input	14
4.6.2	Output	14
4.6.3	Vorgehensweise	15

4.7	Rücksetzantrag senden (<i>requestResetting</i>):	15
4.7.1	Input	15
4.7.2	Output	15
4.7.3	Vorgehensweise	15
4.8	Liste der Dateien einer Probe (<i>getSampleFileList</i>):	15
4.8.1	Input	15
4.8.2	Output	15
4.8.3	Vorgehensweise	16
4.9	Herunterladen einer Datei (<i>getSampleFile</i>)	16
4.9.1	Input	16
4.9.2	Output	16
4.9.3	Vorgehensweise	16
4.10	Hochladen einer Datei (<i>setSampleFile</i>)	16
4.10.1	Input	16
4.10.2	Output	16
4.10.3	Vorgehensweise	16
4.11	Herunterladen der PDF Probebegleitdaten (<i>getSampleReport</i>)	16
4.11.1	Input	16
4.11.2	Output	17
4.11.3	Vorgehensweise	17
4.12	Herunterladen der PDF Analysedaten (<i>getAnalysisReport</i>)	17
4.12.1	Input	17
4.12.2	Output	17
4.12.3	Vorgehensweise	17
4.13	Zuordnung Standorte auflisten (<i>getLocationAssignments</i>)	17
4.13.1	Input	17
4.13.2	Output	17
4.13.3	Vorgehensweise	18
4.14	Zuordnung Standort hinzufügen (<i>setLocationAssignment</i>)	18

4.14.1	Input	18
4.14.2	Output	18
4.14.3	Vorgehensweise	18
4.15	Zuordnung Standort löschen (<i>removeLocationAssignment</i>):	18
4.15.1	Input	18
4.15.2	Output	18
4.15.3	Vorgehensweise	18
5	Schnittstelle für Labore	19
5.1	Probenliste (<i>getSampleList</i>)	19
5.2	Probe herunterladen (<i>getSample</i>):	19
5.3	Probe Daten bestätigen (<i>confirmSampleData</i>)	19
5.4	Zustand ändern (<i>changeSampleState</i>)	19
5.5	Rücksetzantrag senden (<i>requestResetting</i>):	19
5.6	Analyseergebnisse einer Probe herunterladen (<i>getAnalysis</i>):	19
5.7	Labordaten hochladen (<i>setSampleLaboratoryData</i>):	19
5.7.1	Input	19
5.7.2	Output	20
5.7.3	Vorgehensweise	20
5.8	Laborprofil: Parameter / Unterparameter herunterladen (<i>getParameters</i>)	20
5.8.1	Input	20
5.8.2	Output	21
5.8.3	Vorgehensweise	21
6	Informative Funktionen	21
6.1	Futtermittelliste herunterladen (<i>getFeed</i>)	21
6.1.1	Input	21
6.1.2	Output	21
6.1.3	Vorgehensweise	21
6.2	Prüfparameter herunterladen (<i>getInspectionParameters</i>)	22
6.2.1	Input	22

6.2.2	Output	22
6.2.3	Vorgehensweise	22
6.3	Unterparameter herunterladen (<i>getSubParameters</i>)	22
6.3.1	Input	22
6.3.2	Output	22
6.3.3	Vorgehensweise	22
6.4	zugelassene Labore für Futtermittelmonitoring herunterladen	23
6.4.1	Input	23
6.4.2	Output	23
6.4.3	Vorgehensweise	23
7	Begriffs-/ Feldbeschreibung	23
8	Ihre Ansprechpartner zur Spezifikation	25

1 Fachlicher Rahmen

Um die technische Umsetzung der Schnittstelle zum Modul Futtermittelmonitoring der QS-Software-Plattform besser vornehmen zu können, wird einleitend die Fachlichkeit der Futtermittelmonitorings beschrieben. Diese stellt sich grob wie in der Abbildung 1 dargestellten Ablauf dar. Für eine detaillierte fachliche Beschreibung sei an dieser Stelle auf das Dokument Leitfaden Futtermittelmonitoring verwiesen.

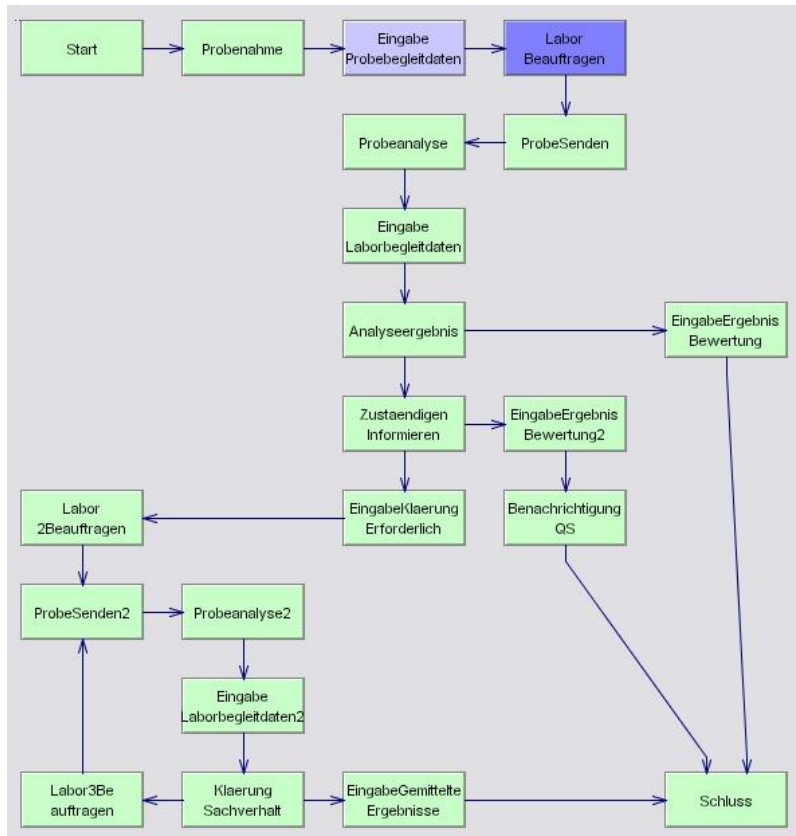


Abbildung 1 Ablauf des Futtermittelmonitorings

In der software-technischen Umsetzung des Prozesses des Futtermittelmonitorings startet dieser mit der Eingabe und Übermittlung der Probegleitdaten. Diese können von einem Probennehmer, einem Bündler bzw. Systemteilnehmer (dem eigentlichen Auftraggeber), einer Probenahmeinstitution (PNI) oder einem Labor (mit Probennehmern) eingegeben werden. Bei der Übergabe der Probegleitdaten an das Labor wechselt der Zustand der Probegleitdaten auf „Labor beauftragt“, erst dann sind die Probegleitdaten für das Labor sichtbar.

Das Labor seinerseits kann nach der Übergabe der Probegleitdaten nun die Laborbegleitdaten und das Analyseergebnis eintragen. Werden keine Überschreitungen von Rückstandshöchstmengen festgestellt, erhalten die Probegleitdaten den Zustand „Analyse abgeschlossen“, der Auftraggeber und QS haben nun Zugriff auf die Analyseergebnisse. Hier endet in diesem Fall der Prozess.

Wird allerdings beispielsweise eine Überschreitung festgestellt, kann nach Rücksprache zwischen Labor und Auftraggeber ein zweites Labor zur Überprüfung des Analyseergebnisses eingeschaltet werden. In diesem Fall erhalten die Probegleitdaten den Zustand „Klärung erforderlich“.

Industry Parc GmbH

Der Auftraggeber hat nun wieder Zugriff auf seine Probebegleitdaten und kann ein zweites Labor mit einer weiteren Analyse der Probe beauftragen. Hier erhalten die Probebegleitdaten den Zustand „Labor beauftragt (Zweituntersuchung)“.

Das zweite Labor hat nun Zugriff auf die Probebegleitdaten und kann die Laborbegleitdaten und „Ergebnis liegt vor“ eintragen. Wird die Eingabe abgeschlossen, wird der Zustand „Ergebnis liegt vor“ für die Probebegleitdaten gesetzt.

Zwischen Auftraggeber und QS erfolgt nun eine Klärung der möglichen Diskrepanz zwischen dem ersten und zweiten Analyseergebnis und QS trägt ggfs. ein Analyseergebnis ein. Danach erhalten die Probebegleitdaten den Zustand „Durch QS abgeschlossen“.

Abbildung 2 stellt die einzelnen Zustandsübergänge der Probebegleitdaten noch einmal dar.

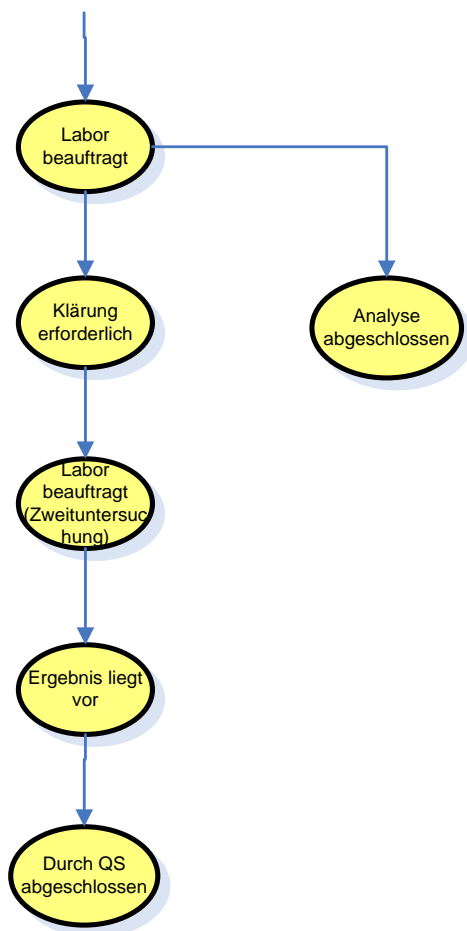


Abbildung 2 Zustandsübergangsdiagramm für die Probebegleitdaten

Um ein besseres Bild von den Abläufen bzw. Möglichkeiten der Benutzergruppen zu vermitteln, möchten wir auf die entsprechenden Anleitungen auf der [Hilfeseite der Plattform](#) unter „Futtermittelmonitoring“ hinweisen.

2 Unterteilung der Futtermittel-Schnittstellen in eine Auftraggeber und eine Labor-Schnittstelle:

Für Systemteilnehmer und Bündler (und die Zertifizierungsstellen für die Verwaltung ihrer Kleinstherzeuger), also für die Auftraggeber, und deren beauftragte Probenahmeinstitutionen (PNI) gibt es eine eigene Schnittstelle, während die Analyse-Labore über eine andere Schnittstelle auf die Softwareplattform zugreifen. So wird schon beim Aufruf der Schnittstelle erkannt, dass hier eine andere Benutzergruppe angesprochen wird und die unterschiedlichen Funktionalitäten sind sauber voneinander getrennt. Auch ist es nicht notwendig die WSDL Datei für die Auftraggeber neu zu gestalten, wenn eine Änderung bei den Laboren notwendig ist und umgekehrt. Des Weiteren werden Benutzergruppen nicht verleitet, Funktionen aufzurufen, die sie nicht nutzen können, da sie diese gar nicht erst zur Verfügung gestellt bekommen.

Die Auftraggeberschnittstelle kann mit folgenden Rollen benutzt werden: Bündler, Systemteilnehmer, Zertifizierungsstellen-Futtermitteladministrator und PNI-Administrator.

3 Allgemeine Richtlinie der Schnittstelle

Diese Webservice Schnittstelle befindet sich im selben technischen Umfeld, wie die anderen QS – Webservice Schnittstellen für Bündler oder Zertifizierungsstellen. Aus diesem Grund befindet sie sich ebenfalls hinter einem HTTPS Proxy Tunnel, um so eine Verschlüsselung der Nachrichten sicherzustellen. Außerdem muss jede Nachricht authentifiziert sein, indem im HTTP Header als User eine aktuell gültige UserID zusammen mit dem zugehörigen Passwort übertragen wird, welches auch verwendet wird, um sich auf der Webseite der Softwareplattform einzuloggen. An dieser Stelle werden dieselben Sicherheits-Mechanismen verwendet, wie sie auch bereits erfolgreich bei den anderen Schnittstellen eingesetzt werden. Aus diesem Grund wurde auch die Technik beibehalten, dass innerhalb der Nachricht zur Verifikation noch einmal die QS-ID des Unternehmens als AuthenticationID angegeben werden muss, bei dem der Anwender/ Nutzer angemeldet ist (eine PNI übermittelt also ihre eigene QS-ID und nicht die zum beprobten Standort passende Auftraggeber-QS-ID). So kann sichergestellt werden, dass der sich authentifizierende Benutzer aus demselben Unternehmen stammt, für das diese Anfrage gesendet wurde. Nur wenn diese übereinstimmt, erfolgt eine Auswertung der Anfrage. Serverseitig wird der Webservice in Java realisiert und zwar unter einem JBoss der Version 3.2.8 und es wird die Apache Axis Version 1.4.2_08-b03 verwendet, welches als Servlet im JBoss Container arbeitet. Axis unterstützt dabei die Standards SOAP 1.1, 1.2 und WSDL 1.1.

4 Schnittstelle für Auftraggeber (Systemteilnehmer, Bündler, Probenahmeinstitution)

Die zu dieser Beschreibung zugehörige WSDL Datei heißt QSFMonitoringOriginator.wsdl.

4.1 Probenliste herunterladen: (*getSampleList*):

4.1.1 Input

AuthentifizierungsID, QSStandortnummer, Produktionsart, , Anfangsdatum für Probenahme, Enddatum für Probenahme, Anfangsdatum für letzte Änderung des Datensatzes, Enddatum für letzte Änderung des Datensatzes, IdZustand

Während die AuthenticationID in der Nachricht zur Verifikation gesendet werden muss, können die anderen Felder beliebig zur Filterung/ Selektion gefüllt werden.

authenticationID, locationNumber, productionType, sampleStateID samplingDateStart, samplingDateEnd, dateOfChangeStart, dateOfChangeEnd

4.1.2 Output

Eine Array-Liste aller Proben, die den Filterkriterien entsprechen: Dabei werden für einen Eintrag die folgenden Felder angezeigt:

ProbenID, Auftraggeber (hier wird Name und QSID entweder des Bündlers oder des Systemteilnehmers eingetragen), IdZustand, Probenahmedatum, Probenahme von und Probenahme bis, UserIDProbennehmer, Code-Nr. des Futtermittel, Name Futtermittel, KzÜberschreitungHGProbe, KzÜberschreitungRWProbe, KzKeineAnalyseVorhanden

sampleID, qSIDOriginator, sampleStateID, sampleDate, samplingFrom, samplingTo, samplerUserID, codeNrFM, nameFM, exceedanceOfMaximum, exceedanceOfReference, noAnalysisAvailable

Die Felder KzÜberschreitungHGProbe, KzÜberschreitungRWProbe werden nur befüllt, wenn die Analyse der Probe abgeschlossen ist und die Ergebnisse vorliegen (diese Ergebnisse werden für PNIs aber nicht befüllt und sie sind immer 0 für sie, da sie kein Recht haben, diese Information einzusehen)

Die Liste wird nach 500 Ergebnissen abgeschnitten, auch wenn mehr als 500 Ergebnisse vorliegen sollten. Falls mehr als 500 Ergebnisse ermittelt wurde, so wird das Flag Überlauf bzw. resultOverflow auf 1 gesetzt, um dies so anzuzeigen, dass noch mehr Ergebnisse vorliegen, aber diese nicht übertragen werden können.

4.1.3 Vorgehensweise

Mit den Inputdaten der Anfrage wird ein entsprechender Proben-Filter befüllt und daraus ergibt sich dann, welche Ergebnisse zurückgeliefert werden müssen. Alle Filterfelder sind über ein UND miteinander verknüpft, so dass jedes aktive Filterfeld zu einer weiteren Einschränkung des Ergebnisses führt, wobei aber jedes Filterfeld optional an oder abgeschaltet werden kann. Es werden also alle Proben zurückgeliefert, welche entweder die angegebene Standortnummer haben, die innerhalb des angegebenen Datumintervalls bzgl. ihres Probenahmedatums bzw. ihrer letzten Änderung eines ihrer Datumsfelder liegen und/oder die den geforderten Zustand haben. Bei den Datumsfelder ist es aber möglich, einen Null-Value in einem Anfang- oder Endefeld zu übertragen, was dann entsprechend bedeutet, dass ab- bzw. bis zu dem anderen Datum alle Proben gesucht werden sollen (wird auch das zweite Datums Feld leer gelassen, so wird diese Filtereinstellung ignoriert).

4.2 Probe herunterladen (*getSample*):

4.2.1 Input

AuthentifizierungsID, die eindeutige ProbenID der Probe (diese Nummer kann aus der Probenliste ausgelesen werden)

authenticationID, sampleID

4.2.2 Output

Es werden die bisher eingetragenen Daten zu der Probe selber zurückgesendet:

- ProbenID, IdProbenart, IdZustand, QsStandortNummer, QsStandortName, Standortadresse (Straße, PLZ, Ort), Produktionsart, QSID Labor1, QSID Labor 2, IdHerkunftsstaat, IdBundesland, UserId des Probennehmer und Name des Probennehmer (oder der PNI/ des Probenahme-Labors), Freitextfeld Probennehmer, Code-Nr. des Futtermittel und Futtermittelbezeichnung, Probenahme von und Probenahme bis, Herstellungsdatum BiProbenbezeichnung, Probenahmedatum, Probenmenge, KzÜberschreitungHGProbe, KzÜberschreitungRWProbe, Einheit, Artikelnummer, Handelsbezeichnung,

Industry Parc GmbH

QSID des Buendlers oder des Systemteilnehmer, Name Auftraggeber und Adresse Auftraggeber (Straße, PLZ, Ort), Bemerkungen oder Besonderheiten, Timestamp des ersten Öffnens durch das Labor

Sample = sampleID, sampleTypeID, qSIDOriginator, originatorName, shortOriginatorAdress = (street, zipcode, city), sampleStateID, locationNumber, produktionType, locationName, shortLocationAdress = (street, zipcode, city), originCountryID, stateID, samplerUserID, samplerFreeText, codeNrFM, nameFM, sampleDate, samplingFrom, samplingTo, productionDate, internalDescriptionName, sampleAmount, AmountUnit, articleNumber, tradeName, commentsOrSpecifics, openedOnDate, exceedanceOfMaximum, exceedanceOfReference

- Für jede Probe wird eine Liste der ausgewählten Prüfparameter angezeigt inkl. der Information, ob es sich dabei um ein Screening handelt (KzScreeningBeauftragt)

inspectionParameters = List of (codeNumber, screening, nameParameter,

- Falls bestimmte Unterparameter ausgewählt sein sollten, so werden diese ebenfalls in einer eigenen Liste aufgelistet, wobei auch hier das KzScreeningBeauftragt berücksichtigt werden kann.

subParameters= List of (casNumber, avvNumber, nameSubParameter, screening, inspectionParameter (nameParameter, codeNumber)),

- Die Analyseergebnis sowie die Laborbegleitdaten 1 ggf. die Laborbegleitdaten 2 werden entsprechend gefüllt, wenn sie zum einen abhängig vom jeweiligen Fall vorliegen und der Teilnehmer/Bündler/PNI das Recht haben, sie zu sehen.

laboratoriesData= [laboratory1QSID, laboratory1data, laboratory2QSID, laboratory2data, analysisdata]

- Falls es Bild- oder sonstige Bit-Dateien zu einer Probe gibt, so werden deren eindeutige ID-Nummer in einer Liste an diese Probe angefügt. Über diese ID kann diese Datei dann separat über eine andere Funktion heruntergeladen werden.

- Die Laborbegleitdaten bauen sich auf aus:

Probeneingangsdatum, Analyse von (tt.mm.jjjj), Analyse bis (tt.mm.jjjj), Name des Analytikers, Laborinterne Probennummer, Bemerkungsfeld, Bewertungsfeld, kz_Unterauftrag vergeben, Unterauftrag vergeben an

laboratory1QSID, laboratory1data = (sampleEntranceDate, analysisStart, analysisEnd, analyzerName, laboratorySampleNumber, comment, evaluation], ,subcontracting, subContractGivenTo

Ein Analyseergebnis baut sich auf aus einer Liste von folgenden Einzel-Daten für jeden Unterparameter auf:

CasNr, Unterparametername, Parameter (Name + ID), kzScreeningAnalysiert, Verwendete Methode, AvvDup Nr, Ergebniswert +Einheit, Bestimmungsgrenze +Einheit, Analysenspielraum +Einheit, Grenzwert +Einheit, QS-Richtwert +Einheit, KzÜberschreitungHGUnterparameter KzÜberschreitungRWUnterparameter KzKlaerung

analysisdata = list of {subParameter = { casNumber, avvNumber, nameSubParameter }, screening, inspectionParameter = (codeNumber + name), resultValue, resultUnit, determinationLimit, determinationUnit, analysisTolerance, limit, limitUnit, guidingValue, guidingUnit, resultFound, exceedanceOfMaximum, exceedanceOfReference, reviewRequired})

4.2.3 Vorgehensweise

Die ProbenID wird gesucht. Wird sie gefunden und darf diese von dem Teilnehmer/Bündler/PNI/Labor gesehen werden, so werden alle relevanten Daten aus der Datenbank entsprechend ausgelesen und zurückgesendet, wobei alle Regeln beachtet werden, welche Information von welchem Benutzer gesehen werden darf.

4.3 Probe hochladen: (*setSample*):

4.3.1 Input

AuthentifizierungsID, IdZustandNachher, bestaetigung, Probe = ProbenID, ProbenIDNeu, IdProbenart, QsStandortNummer, Produktionsart, Labor QS-ID1, Labor QS-ID2, IdHerkunftsstaat, UserId des Probennehmer (oder QS-ID PNI oder QS-ID Labor), Freitextfeld Probennehmer, Code-Nr. des Futtermittel, Herstellungsdatum, IdBundesland, BiProbenbezeichnung, Probenahmedatum, ProbenahmeOrt, Probenmenge, Einheit (kg), Artikelnummer, Handelsbezeichnung, QSID des Bündlers oder des Systemteilnehmers

authenticationID, sampleStateIDAfter, confirm, Sample = (siehe oben)

Eine Liste mit Prüfparametern (inkl. Screening), auf die untersucht werden sollen. Optional eine Liste mit Unterparametern, falls die Untersuchungen auf diese eingeschränkt werden sollen.

Wenn eine Probe an das Labor übergeben werden soll, sind neben der in jeder Nachricht zur Verifikation verpflichtenden QSID als AuthenticationID folgende Daten erforderlich:

ProbenID oder QsStandortNummer samt Produktionsart

Labor QS-ID1

UserId des Probennehmer (bzw. QS-ID PNI oder QS-ID Labor)

Code-Nr. des Futtermittel

Herstellungsdatum oder alternativ BiProbenbezeichnung

Probenahmedatum

IdHerkunftsstaat

IdBundesland

mind. ein Parameter bzw. Unterparameter

4.3.2 Output

Falls es beim Hochladen der Probe ein Problem gibt, so wird eine Exception geworfen und so auf den fehlerhaften Datenbestandteil hingewiesen. Sind alle Daten korrekt und können sie auf die Probe übertragen werden, so wird auf jeden Fall die ProbenID zurückübertragen (falls keine angegeben wurde, so wurde eine neue generiert) und außerdem der aktuelle Zustand der Probe an den Aufrufenden zurückübertragen. Wurde hingegen ein anderer IdZustandNach angegeben, als der in dem die Probe sich aktuell befindet, so wird als Ergebnis neben der ProbenID der IdZustandNach zurückgegeben, wenn der Übergang gültig ist (dies ist dieselbe Antwort, die bei einem entsprechenden Aufruf an die Übergangsfunktion zurückgegeben werden würde. Der gleiche Effekt kann erzielt werden, wenn der Wert von Bestätigen auf True gesetzt wird, da somit angezeigt wird, dass diese Probe bestätigt werden soll.

sampleResponse = sampleID, sampleStateID

4.3.3 Vorgehensweise

Es wird über die ProbenID nach der Probe gesucht. Wird über diese ProbenID keine Probe gefunden, so wird eine neue Probe mit den übertragenen Daten angelegt (dabei werden natürlich die Rechte

Industry Parc GmbH

beachtet, die für das Anlegen von Proben gelten). In diesem Fall einer Neuanlegung einer Probe wird zuerst auf die ProbenID zurückgegriffen, ist diese leer, wird die ProbenIDNeu verwendet, sollte diese aber auch leer sein, so wird dann erst vom System eine neue ProbenID generiert. In jedem Fall wird bei einer erfolgreichen Übertragung der Daten auf die Probe, die aktuell gültige ProbenID zurückübertragen. Wird die Probe über die ProbenID gefunden, so kann die aktuelle ProbenID durch die ProbenIDNeu ersetzt werden, wenn dieses Feld gefüllt sein sollte.

Wenn alle Bedingungen erfüllt sind, dass diese Daten übertragen werden können (insbesondere wenn die Probe noch in einem Zustand ist, in dem noch Daten übertragen werden können) und alle Werte auch korrekte und zulässige Einträge enthalten (hierbei wird z.B. auch überprüft, ob auf den gewählten Standort überhaupt ein Zugriffsrecht besteht), so werden diese Daten auf die Probe übertragen, wobei die übermittelten Daten alle alten Daten der Probe überschreiben. Das Feld LaborQS-ID2 wird nur dann ausgewertet, bevor die Probe zum 2ten Labor geschickt werden soll. In diesem Fall, wenn für die Probe also eine Klärung ansteht, werden die Daten in den anderen Feldern der Probe ignoriert und es wird nur die LaborQS-ID2 ausgewertet.

Nachdem die Probedaten korrekt eingetragen worden sind und es dabei zu keinen Exceptions kam, werden diese Daten in der Datenbank eingetragen und anschließend wird die Variable IdZustandNach ausgewertet: Über diesen IdZustandNach kann festgelegt werden, in welchen Folge-Zustand die Probe versetzt werden soll. Dafür muss auf das Zustandsdiagramm zurückgegriffen werden, um so zu erkennen, welche Folgezustände vom aktuellen Zustand der Probe aus zulässig sind. Wurde ein gültiger Übergang gewählt, so wird die Statusübergangsfunktion 4.5 aufgerufen und zwar mit dem aktuellen Zustand der Probe als Start- und dem Zustand aus idZustand als Ziel. Falls dieser Zustandsübergang erfolgreich war, so werden die ProbenID und der IdZustandNach zurückübertragen.

4.4 Probe Daten bestätigen (*confirmSampleData*)

4.4.1 Input

AuthentifizierungsID, ProbenID, IdZustandVorher

4.4.2 Output

Exceptions falls es zu einem Fehler bei der Änderung des Zustands kommt, z.B. der Zustandsübergang nicht erlaubt oder nicht vorgesehen ist, ein bestimmtes Pflicht-Datenfeld nicht die richtigen Daten enthält, damit der Übergang erfolgen kann, u.ä.

Falls die Übertragung erfolgreich war, so wird der IdZustandNach zurückgegeben.

sampleStateIDAfter

4.4.3 Vorgehensweise

Wenn der Benutzer Schreibe-Zugriff auf die Probe mit der ProbenID hat, so wird deren aktueller Zustand bestimmt. Stimmt dieser mit dem IdZustandVoher nicht überein, ist dies ein Fehler. Ist dies erfüllt, so wird die Probe in den nächsten Zustand übertrage. Es wird dabei aber immer überprüft, ob alle Pflichtfelder in der betreffenden Probe korrekt gefüllt sind (falls nicht, so wird eine Exception geworfen, in der das entsprechende Feld beanstandet wird). Sind alle diese Bedingungen erfüllt und es kam zu keiner Exception, so wird der Übergang durchgeführt und die Probe im neuen Zustand abgespeichert und der IdZustandNach zurückübertragen.

4.5 Zustand ändern (*changeSampleState*)

4.5.1 Input

AuthentifizierungsID, ProbenID, IdZustandVoher, IdZustandNachher

4.5.2 Output

Exceptions falls es zu einem Fehler bei der Änderung des Zustands kommt, z.B. der Zustandsübergang nicht erlaubt oder nicht vorgesehen ist, ein bestimmtes Pflicht-Datenfeld nicht die richtigen Daten enthält, damit der Übergang erfolgen kann, u.ä.

Falls die Übertragung erfolgreich war, so wird der IdZustandNach zurückgegeben.

sampleStateIDAfter

4.5.3 Vorgehensweise

Es passiert dasselbe wie bei der Funktion Probe Daten bestätigen in der aktuellen Funktion, d.h., dass die Variable IdZustandNachher nicht ausgewertet wird.

In zukünftigen Versionen wird die folgende Vorgehensweise implementiert sein: Anschließend wird geprüft, ob im Zustandsübergangsdiagramm ein Übergang nach IdZustandNach vorgesehen ist. Ist dies nicht der Fall, so wird eine entsprechende Exception geworfen. Falls dieser Zustand im Diagramm vorgesehen ist und der Übergang korrekt und möglich ist, so wird dieser Übergang ausgewählt.

4.6 Analyseergebnisse einer Probe herunterladen (*getAnalysis*):

4.6.1 Input

AuthentifizierungsID, ProbenID (diese Nummer kann aus der Probenliste ausgelesen werden)

4.6.2 Output

es werden nur die Analyseergebnisse der Probe zurückgesendet:

- ProbenID, IdZustand

authenticationID, sampleID

- Die Analyseergebnis sowie die Laborbegleitdaten 1 ggf. die Laborbegleitdaten 2 werden entsprechend gefüllt, wenn sie zum einen abhängig vom jeweiligen Fall vorliegen und der Teilnehmer/Bündler/PNI das Recht haben, sie zu sehen.

- Die Laborbegleitdaten bauen sich auf aus:

Probeneingangsdatum, Analyse von (tt.mm.jjjj), Analyse bis (tt.mm.jjjj), Name des Analytikers, Laborinterne Probennummer, Bemerkungsfeld, Bewertungsfeld, kz_Unterauftrag vergeben, Unterauftrag vergeben an

LaboratoryData = sampleEntranceDate, analysisStart, analysisEnd, analyzer, laboratorySampleNumber, subcontracting, subContractGivenTo, comment, evaluation

- Ein Analyseergebnis baut sich auf aus einer Liste folgender Einzel-Daten für jeden Unterparameter auf:

, CasNr, Unterparametername, Parameter (Name + ID), kzScreeningAnalysiert, Verwendete Methode, AvvDup Nr, Ergebniswert + Einheit, Bestimmungsgrenze + Einheit, Analysenspielraum + Einheit, Grenzwert + Einheit, QS-Richtwert + Einheit, KzÜberschreitungHGUnterparameter, KzÜberschreitungRWUnterparameter, KzKlaerung, Ergebnis liegt vor

analysisdata = list of {subParameter = { casNumber, avvNumber, nameSubParameter, screening, inspectionParameter (nameParameter, codeNumber), methodeName, resultValue, resultUnit, determinationLimit, determinationUnit, analysisTolerance,

analysisTolerance Unit, limit, limitUnit, guidingValue, guidingUnit, resultFound, exceedanceOfMaximum, exceedanceOfReference, reviewRequired})

4.6.3 Vorgehensweise

Die Funktion kann nicht von PNIs aufgerufen werden und ergibt sofort eine Exception. Die Id wird gesucht und es wird entschieden, ob diese von dem Teilnehmer/Bündler/ Labor gesehen werden darf oder nicht. Es werden alle relevanten Daten aus der Datenbank entsprechend ausgelesen und zurückgesendet, wobei alle Regeln beachtet werden, welche Information von welchem Benutzer/Labor gesehen werden dürfen.

4.7 Rücksetzantrag senden (*requestResetting*):

4.7.1 Input

AuthentifizierungsID, ProbenID, Begründung (Freitext)

authenticationID, sampleID, commentar

4.7.2 Output

IdZustand

sampleStateID

4.7.3 Vorgehensweise

Falls kein Rücksetzantrag für diese Probe gestellt werden kann (weil z.B. dazu keine Recht besteht oder kein Zugriff auf diese Probe mehr besteht), so wird eine Exception geworfen. Falls der Rücksetzantrag erfolgreich war und er an QS gesendet wurde, so wird die ProbeID zusammen mit dem aktuellen Zustand der Probe zurückgesendet. Es erfolgt kein automatischen Zurücksetzen der Probe, sondern QS entscheidet über den jeweiligen Fall.

4.8 Liste der Dateien einer Probe (*getSampleFileList*):

4.8.1 Input

AuthentifizierungsID, ProbenID

authenticationID, sampleID

4.8.2 Output

Eine Array-Liste aller Dateien, die dieser Probe zugeordnet sind. Zu jeder Datei wird aber nur ihr Dateiname und die zugehörige, eindeutige fileID übermittelt. So kann über die Funktion Herunterladen mit Hilfe der DateiID die entsprechende Datei heruntergeladen werden.

Dateiname, dateiID

filename, fileID

4.8.3 Vorgehensweise

Alle Dateien, die dieser Probe zugeordnet worden sind, werden in dem Array aufgelistet und zurückgesendet

4.9 Herunterladen einer Datei (*getSampleFile*)

4.9.1 Input

AuthentifizierungsID, ProbenID, DateiID

authenticationID, sampleID, fileID

4.9.2 Output

BitStream der Datei

Ein im SOAP Antwortstream verpackter Bit-stream enthält dann die Daten der Datei, die zu dieser ProbenID gehört und die unter der eindeutigen Datei-ID auf dem Server gespeichert ist.

file

4.9.3 Vorgehensweise

Über die ProbenID wird die Probe gesucht und durch die dateiID wird die Datei eindeutig bestimmt, die zu übertragen ist.

4.10 Hochladen einer Datei (*setSampleFile*)

4.10.1 Input

AuthentifizierungsID, ProbenID, Dateiname, BitStream der Datei

authenticationID, sampleID, filename, file

4.10.2 Output

Wenn die Datei aus welchen Gründen auch immer nicht gespeichert und hochgeladen werden konnte, so wird eine diesbezügliche Exception geworfen, die den Fehler charakterisiert. Ist der Upload erfolgreich, wird die eindeutigen ID der Datei zurückgegeben.

fileID

4.10.3 Vorgehensweise

Die Probe wird gesucht, die Berechtigung wird gecheckt und gegebenenfalls wird der BitStream als Datei zu dieser Probe gespeichert. Die dafür generierte eindeutige ID wird an den Aufrufenden zurückgesendet.

4.11 Herunterladen der PDF Probebegleitdaten (*getSampleReport*)

4.11.1 Input

AuthentifizierungsID, ProbenID

authenticationID, sampleID

4.11.2 Output

Es wird eine entsprechende PDF Datei mit den Probegleitdaten erzeugt, diese wird in einen SOAP Antwortstream verpackt und dann genauso an den Sender zurückgesendet, wie dies beim Dateiversand erfolgt.

file

4.11.3 Vorgehensweise

Analoge Umsetzung wie oben (4.8) beschrieben.

4.12 Herunterladen der PDF Analysedaten (*getAnalysisReport*)

4.12.1 Input

AuthentifizierungsID, ProbenID

authenticationID, sampleID

4.12.2 Output

Es wird eine entsprechende PDF Datei mit den Analyseergebnissen erzeugt, diese wird in einen SOAP Antwortstream verpackt und dann genauso an den Sender zurückgesendet, wie dies beim Dateiversand erfolgt.

file

4.12.3 Vorgehensweise

Analoge Umsetzung wie oben (4.8) beschrieben.

4.13 Zuordnung Standorte auflisten (*getLocationAssignments*)

4.13.1 Input

AuthentifizierungsID, UserIDZuordnung, QSID

authenticationID, userID, QSID

Als Filter muss eine gültige UserIDProbennehmer oder UserIdStandortAdministrator im Feld UserIDZuordnung angegeben werden oder aber eine gültige QSID eines Labors oder einer PNI. Es dürfen dabei aber nicht beide Felder gleichzeitig gefüllt sein. Entweder das eine oder das andere.

4.13.2 Output

Es wird eine Liste von StandortNummern + Produktionsart, PLZ, Ort zurückgegeben, für die der entsprechende User (also der Probennehmer bzw. der Standortadministrator) bzw. das Labor oder die PNI aktuelle eine Zuordnung hat.

locationNumber, productiontype, street, zipcode, city

4.13.3 Vorgehensweise

Alle Standorte, die diesem User/Labor/PNI zugeordnet sind, werden aufgelistet und ihre Standortnummer und aktuell angemeldete Produktionsart, PLZ, Ort werden zurückübertragen.

4.14 Zuordnung Standort hinzufügen (*setLocationAssignment*)

4.14.1 Input

AuthentifizierungsID, UserID, QSID, Standortnummer, Produktionsart

authenticationID, userID, qSID, locationNumber, productionType

4.14.2 Output

Ok, falls die Übertragung geglückt ist, ansonsten eine entsprechende Exception.

sucess

4.14.3 Vorgehensweise

Es wird über die UserID der entsprechende User = Probennehmer/Standortadministrator gesucht bzw. über die QSID das entsprechende Labor oder PNI. Wird er/es gefunden, so wird nach dem Standort gesucht. Wird auch er beim entsprechenden Auftraggeber gefunden, so wird eine Zuordnung dazwischen hergestellt. Falls diese bereits existiert, geschieht nichts.

4.15 Zuordnung Standort löschen (*removeLocationAssignment*):

4.15.1 Input

AuthentifizierungsID, UserID, QSID, Standortnummer, Produktionsart

authenticationID, userID, qSID, locationNumber, productionType

4.15.2 Output

„Ok“, falls die Lösung erfolgreich war, ansonsten wird eine entsprechende Exception generiert.

sucess

4.15.3 Vorgehensweise

Es wird über die UserID entweder der Probennehmer oder der Standortadministrator bzw. über die QSID das Labor oder die PNI gesucht. Wird er/sie gefunden, so wird für diesen User/Labor/PNI nach seiner Zuordnung zu dem angegebenen Standort gesucht. Wird diese Zuordnung gefunden, so wird sie gelöscht. Falls diese nicht existiert, wird eine entsprechende Exception geworfen, die signalisiert, dass diese Zuordnung bisher überhaupt nicht gespeichert war.

5 Schnittstelle für Labore

Die zu dieser Beschreibung zugehörige WSDL Datei heißt *QSFMonitoringLaboratory.wsdl*

5.1 Probenliste (*getSampleList*)

Für diese Funktion wird auf die entsprechende Funktionalität der Schnittstelle wie oben unter 4.1 zurückgegriffen. Die entsprechende Funktion wird hier mit derselben Signatur noch mal angeboten und beide Aufrufe werden an eine gemeinsame Funktion weitergeleitet, so dass beide identische Ergebnisse liefern.

5.2 Probe herunterladen (*getSample*):

Auch hier wird dieselbe Funktion verwendet, die schon oben unter 4.2 beschrieben wurde. Bei einem ersten Download eines Datensatzes durch ein Labor, der im Zustand Labor beauftragt ist, wird der Timestamp des ersten Öffnens durch das Labor gefüllt und der entsprechende Analytiker wird eingetragen und zwar desjenigen Analytiklers über dessen Login dieser Webserviceaufruf authentifiziert worden ist.

Es werden natürlich nur die Labordaten zurückgeliefert, auf die das Labor einen Zugriff hat.

5.3 Probe Daten bestätigen (*confirmSampleData*)

Es wird auf dieselbe Funktion zurückgegriffen, die schon oben unter 4.4 beschrieben wurde. Dabei sind die möglichen Übergänge entsprechend eingeschränkt auf die Zustände die für ein Labor relevant sind.

5.4 Zustand ändern (*changeSampleState*)

Auch hier wird dieselbe Funktion verwendet, die schon oben unter 4.5 beschrieben wurde. Dabei sind die möglichen Übergänge entsprechend eingeschränkt auf die Zustände die für ein Labor relevant sind

5.5 Rücksetzantrag senden (*requestResetting*):

Es wird dieselbe Funktionalität, die unter 4.7 beschrieben wurde, eingesetzt.

5.6 Analyseergebnisse einer Probe herunterladen (*getAnalysis*):

Auch hier wird dieselbe Funktion verwendet, die schon oben unter 4.6 beschrieben wurde.

5.7 Labordaten hochladen (*setSampleLaboratoryData*):

5.7.1 Input

AuthentifizierungsID, ProbenID, IdZustandNach, bestaetigen

authenticationID, sampleID, sampleStateIDAfter, confirm

laborQSID, Probeneingangsdatum, Analyse von (tt.mm.jjjj), Analyse bis (tt.mm.jjjj), UserID des Analytiklers, kz_Unterauftrag vergeben, Unterauftrag vergeben an , Laborinterne Probennummer

LaboratoryData = laboratoryID, sampleEntranceDate, analysisStart, analysisEnd, analyzer, laboratorySampleNumber, subcontracting, subContractGivenTo comment, evaluation

Industry Parc GmbH

Eine Liste aller Analyseergebnisse, wobei jede Datensatz aus folgenden Werten besteht: CasNr, Parameter-ID, KzScreeningAnalysiert, Ergebniswert +Einheit, Bestimmungsgrenze +Einheit, Analysenspielraum +Einheit, Grenzwert +Einheit, QS-Richtwert +Einheit, KzKlaerung, ErgebnisLiegtVor

analysisdata = list of {subParameter = { casNumber, avvNumber, nameSubParameter , screening, inspectionParameter (nameParameter, codeNumber), methodeName, resultValue, resultUnit, determinationLimit, determinationUnit, analysisTolerance, limit, limitUnit, guidingValue, guidingUnit, resultFound, reviewRequired})

5.7.2 Output

Es wird die ProbenID und IdZustandNach übermittelt, sowie das gesamte Analyseergebnis, was für diese Probe eingetragen worden ist, wobei so die KzÜberschreitungHGUnterparameter- und KzÜberschreitungRWUnterparameter erstgestellt werden.

analysisResponse = sampleID, sampleStateID, analysisdata = list of {subParameter = { casNumber, avvNumber, nameSubParameter , inspectionParameter (nameParameter, codeNumber), methodeName, screening, resultValue, resultUnit, determinationLimit, determinationUnit, analysisTolerance, limit, limitUnit, guidingValue, guidingUnit, resultFound, exceedanceOfMaximum, exceedanceOfReference, reviewRequired})

5.7.3 Vorgehensweise

Die Probe wird nach den üblichen Berechtigungsregeln gesucht und es wird über die CasNr das entsprechende Laborergebnis des Labor 1 oder Labor 2 gesucht. Wird es gefunden, so werden die Daten überschrieben, sofern dafür das Recht besteht, ansonsten wird ein neues Ergebnis angelegt.

Wird sie gefunden und der Zugriff ist auf diese Daten erlaubt, und sind außerdem die übermittelten Daten korrekt, so überschreiben sie die bisher eingetragenen Daten. Es werden natürlich nur die Laborbegleitdaten/ Laborergebnisse gefüllt, die zu dem Labor passen. Auch ist es nur für eingeloggte LaborAdministratoren möglich andere Analytiker einzutragen, während eingeloggte Analytiker nur sich selber bestimmen können.

Falls bei einem Eintrag ein Fehler auftauchen sollte, so wird in der Exception die zugehörige CasNr und der Name des Unterparameters genannt, um so den Eintrag eindeutig identifizieren zu können.

Es ist nicht notwendig alle Felder auszufüllen oder alle Daten in einem Schritt zu übertragen. Sobald in dieser Nachricht Werte gefunden werden, so überschreiben die aktuellen Werte die alten Daten. Da keine Analyseergebnisse oder sonstige Daten gelöscht werden können, ist diese Vorgehensweise ausreichend.

Falls das Labor2 Analyse - Daten hochlädt, so werden alle Daten in den Datenfeldern wie Analyseergebnis / Resultvalue oder Richtwert / guidingValue ignoriert und nur das Feld ErgebnisLiegtVor / resultFound wird ausgewertet und eingetragen.

Dies gilt in analoger Weise bei Labor1 für einen Unterparameter, bei dem das Labor KzKlaerung markiert hat.

Falls ein Nachfolge-Zustand idZustandNach angegeben worden sein sollte oder falls der Wert von bestaetigen bzw. confirm auf True stehen sollte, so wird dies genauso behandelt, wie dies bei anderen Hochladen erfolgt, indem der Übergang im Zustandsübergangdiagramm gesucht wird bzw. der nächste folgende Zustand ausgewählt wird. Auch erfolgt ein Übergang natürlich nur dann, wenn alle Daten vollständig vorliegen. Falls der Übergang erfolgreich ist, werden die ProbenID und der IdZustandNach zurückübermittelt. Siehe auch Beschreibung unter Laborbegleitdaten hochladen

5.8 Laborprofil: Parameter / Unterparameter herunterladen (*getParameters*)

5.8.1 Input

authenticationID

5.8.2 Output

Eine Liste aller aktuell Parameter/ Unterparameter die im Laborprofil vorhanden sind.

laboratoryName, laboratoryQS-ID,

parameterList = inspectionParameters list of { codeNumber, screening, parameterName, unit}

subParameters = list of [casNumber, avvNumber, subParameterName, screening, inspectionParameter = (codeNumber, nameParameter), determinationLimit, determinationUnit, analysisTolerance, methode]

5.8.3 Vorgehensweise

Aus dem Laborprofil werden alle Parametern / Unterparametern übermittelt, die für das Labor relevant sind.

6 Informative Funktionen

Die zu dieser Beschreibung zugehörige WSDL Datei heiß QSFMonitoringOpen.wsdl

Für diese Funktionen wird keine Authentifizierung benötigt und die Übertragung erfolgt ohne HTTPS, da hier öffentlich zugängliche Informationen abgefragt werden.

6.1 Futtermittelliste herunterladen (*getFeed*)

6.1.1 Input

Dieser Aufruf kann optional mit einem Datumsfeld befüllt werden, das einen Filter darstellt, dass nur Futtermittel aufgelistet werden sollen, die ab diesem Datum hinzugefügt oder verändert worden sind

after

6.1.2 Output

Es wird eine Liste von Futtermitteln zurückgegeben, welches für jeden Eintrag die folgenden Felder hat:

Bezeichnung, Code-Nr., KzEinzelfuttermittel

feedList = List of { codeNrFM, description, feedMaterial }

6.1.3 Vorgehensweise

Es wird über alle Futtermittel gegangen und die entsprechenden Daten werden in die Liste übertragen. Im Feld KzEinzelfuttermittel wird gespeichert, ob es sich bei diesem Futtermittel um ein Einzelfutter handelt. Falls der Datumsfilter aktiviert wurde, werden natürlich nur solche Futtermittel in die Liste aufgenommen, die AN oder NACH diesem Datum geändert oder hinzugefügt worden sind. Ohne Datumsfilter werden alle Futtermittel ausgewählt.

6.2 Prüfparameter herunterladen (*getInspectionParameters*)

6.2.1 Input

Optional kann der Aufruf mit einem Datumfeld befüllt werden, das dann einen Filter aktiviert, der nur Prüfparameter durchläßt, die ab diesem Datum hinzugefügt oder verändert worden sind.

after

6.2.2 Output

Es wird eine Liste von Prüfparametern zurückgegeben, welches für jeden Eintrag die folgenden Felder hat:

Prüfparametername, Code-Nr., KzScreening, IdEinheit

inspectionParameterList = List of { codeNumber, screening, nameParameter, }

6.2.3 Vorgehensweise

Es wird über alle Prüfparameter gegangen und die entsprechenden Daten werden in die Liste übertragen. Das Feld KzScreening gibt dabei an, ob für diesen Prüfparameter auch ein Screening möglich ist oder nicht. In der IdEinheit wird die Standard-Maßeinheit für diesen Prüfparameter gespeichert. Falls der Datumsfilter aktiviert wurde, werden natürlich nur solche Prüfparameter in die Liste aufgenommen, die AN oder NACH diesem Datum geändert oder hinzugefügt worden sind. Ohne Datumsfilter werden alle Prüfparameter ausgewählt.

6.3 Unterparameter herunterladen (*getSubParameters*)

6.3.1 Input

Im Aufruf kann ein Datumfeld optional mit übermittleit werden, welches dann einen Filter befüllt, der nur solche Unterparameter durchläßt, die ab diesem Datum hinzugefügt oder verändert worden sind.

after

6.3.2 Output

Es wird eine Liste von Unterparametern (Wirkstoffen) zurückgegeben, welches für jeden Eintrag die folgenden Felder hat:

Unterparametername, CAS-Nummer. AVV-DüP, Prüfparameter

subParameterList = list of { casNumber, avvNumber, nameSubParameter, screening, inspectionParameter = (codeNumber, nameParameter)}

6.3.3 Vorgehensweise

Es wird über alle Unterparameter gegangen und die entsprechenden Daten werden in die Liste übertragen. Falls der Datumsfilter aktiviert wurde, werden natürlich nur solche Unterparameter in die Liste aufgenommen, die AN oder NACH diesem Datum geändert oder hinzugefügt worden sind. Ohne Datumsfilter werden alle Unterparameter ausgewählt.

6.4 zugelassene Labore für Futtermittelmonitoring herunterladen

6.4.1 Input

Es sind keine Inputdaten notwendig.

6.4.2 Output

Es wird eine Liste aller aktuell zugelassenen Labor für das Futtermittelmonitoring erstellt mit folgenden Einträgen:

QSID, Name, PLZ, Ort, Telefon, „aktive“ Parameter/ Unterparameter (aus dem Laborprofil)

laboratoryQS-ID, laboratoryName, country, adress = {street, zipcode, city} telephone...

parameterList = inspectionParameters list of { codeNumber, screening, nameParameter, }

subParameters = list of [casNumber, avvNumber, nameSubParameter, screening, inspectionParameter = (codeNumber, nameParameter), determinationLimit, determinationUnit, analysisTolerance, analysisToleranceUnit, methode]

6.4.3 Vorgehensweise

Alle zum Zeitpunkt der Abfrage zugelassenen Labore werden bestimmt und ihre jeweilige QS-ID und der Name des Labors samt Adressdaten und Telefonnummer werden übermittelt. Dabei wird eine Liste aller aktiven Parametern / Unterparametern übermittelt, die das Labor aktuell anbietet.

7 Begriffs-/ Feldbeschreibung

KzÜberschreitungHGProbe: Kennzeichnung einer Probe mit einer Überschreitung eines Höchstgehaltes/ Grenzwertes bei mind. einem Parameter/ Unterparameter.

KzÜberschreitungRWProbe: Kennzeichnung einer Probe mit einer Überschreitung eines (QS-)Richtwertes bei mind. einem Parameter/ Unterparameter.

KzÜberschreitungHGUnterparameter: Kennzeichnung eines Unterparameters bei dem eine Überschreitung eines Höchstgehaltes/ Grenzwertes erfolgt ist.

KzÜberschreitungRWUnterparameter: Kennzeichnung eines Unterparameters bei dem eine Überschreitung eines (QS-)Richtwertes erfolgt ist.

Produktionsart: Code der Produktionsart (z.B. 1001 = Rindermast oder 72 = Einzelfuttermittelhersteller)

Anfangsdatum für Probenahme, Enddatum für Probenahme: Hier kann optional vom Auftraggeber für die Beauftragung des Probennehmers oder der PNI ein Zeitintervall eingetragen werden, in dem die Probe möglichst gezogen werden soll.

Anfangsdatum für letzte Änderung des Datensatzes, Enddatum für letzte Änderung des Datensatzes: Hier kann optional ein Zeitintervall für das Datum der letzten Änderung bei (Proben-)Datensätzen abgefragt werden, um alle entsprechenden Datensätze zu identifizieren.

IdZustand: Dies bezeichnet einen Zustand, den eine Probe annehmen kann, der im Zustandsdiagramm definiert ist

IdZustandNach: Dies bezeichnet den Zustand, in den die Probe versetzt werden soll. Dieser Zustand muss im Zustandsdiagramm definiert sein und er muss vom aktuellen Zustand aus erreichbar sein. Außerdem muß natürlich das Recht bestehen, diesen Übergang durchführen zu dürfen.

CasNr: Dies ist die eindeutige Identifikationsnummer, um einen Unterparameter zu identifizieren. Diese Nummer wird der AvvDup Nr vorgezogen.

AvvDup Nr: Dies ist eine alternative Nummer, um einen Unterparameter zu identifizieren.

Industry Parc GmbH

Parameter-ID: Dies ist die eindeutige Nummer, um einen Prüfparameter zu identifizieren.

kzScreeningBeauftragt: Hier ist in den Probabegleitdaten die Beauftragung der Untersuchung als Screening gemeint.

kzScreeningAnalysiert: Bei den Laborergebnissen wird hiermit übertragen, wenn ein Screening-Verfahren verwendet wurde.

KzKlaerung: Dies ist das Flag, was vom Labor1 aktiviert werden soll, wenn das Ergebnis der Analyse eine Klärung bedarf.

ProbenahmeOrt: Vordef. Möglichkeiten, z.B. „Lager“.

IdZustandVon: Dies bezeichnet den aktuellen Ursprungs-Zustand der Probe, von dem aus die Transition erfolgen soll. (Dieser übermittelte Zustand muß mit dem tatsächlichen Zustand der Probe übereinstimmen, da nur so sichergestellt werden kann, dass die richtige Transition durchgeführt werden soll)

IdZustandNach: Dies bezeichnet den Ziel-Zustand, in den die Probe durch die Transition gebracht werden soll.

Verwendete Methode: Die vom jeweiligen Labor für jeden Parameter/ Unterparameter verwendete Methode wird von QS für das Labor hinterlegt (Laborprofil) und als Information mit den Analyseergebnissen ausgegeben.

Herstellungsdatum: Herstellungsdatum des Futtermittels (tt.mm.jjjj)

BiProbenbezeichnung: Betriebsinterne (Proben-)Bezeichnung (Freitext-Feld).
Zum Zweck der Rückverfolgbarkeit und Identifizierbarkeit muss entweder das Herstellungsdatum oder die betriebsinterne Bezeichnung angegeben werden.

Probenahmedatum: Datum der Probenziehung (tt.mm.jjjj)

IdHerkunftsstaat: Code des Landes, aus dem das Futtermittel stammt (ISO 3166 Ländercode; z.B.: 276 = Deutschland)

IdBundesland: Code des Bundeslandes (falls Herkunftsstaat 276 = Deutschland); (01 = Schleswig-Holstein);(02 = Hamburg);(03 = Niedersachsen);(04 = Bremen);(06 = Hessen);(07 = Rheinland-Pfalz);(08 = Baden-Württemberg);(09 = Bayern);(10 = Saarland);(12 = Brandenburg);(13 = Mecklenburg-Vorpommern);(14 = Sachsen);(15 = Sachsen-Anhalt);(16 = Thüringen);

Ergebniswert: Zahl mit Kommatrennung oder "n.n" (nicht nachweisbar), "n.b."(nicht bestimmbar), "n.a." (nicht analysiert)

Einheit(en): Code für die Maßeinheit; (ppm = Milligramm/Kilogramm (mg/kg); ppb = Mikrogramm/Kilogramm (µg/kg); ppt = Nanogramm/Kilogramm (ng/kg)

Analysespielraum: Der Analysespielraum ist für die Berechnung, ob eine Überschreitung des eingegebenen Grenz- oder (QS-)Richtwertes vorliegt, zwingend erforderlich. Bei Übergabe einer Zahl mit einem Prozentzeichen (%) wird von der Anwendung vom „Ergebniswert“ ausgehend gerechnet, ansonsten wird eine absolute Zahl zugrunde gelegt.

Grenzwert: Zahl mit Kommatrennung; für (Unterparameter mit Höchstgehalt ist die Eingabe bei Eintrag eines Messwertes (Zahl) zwingend erforderlich.

Industry Parc GmbH

Richtwert: Zahl mit Kommatrennung; für (Unterparameter mit QS-Richtwert ist die Eingabe bei Eintrag eines Messwertes (Zahl) zwingend erforderlich.

Bestimmungsgrenze: Zahl mit Kommatrennung; wird aus dem hinterlegten Laborprofil hinzugefügt, falls keine Einträge vorgenommen werden.

Proben-ID: Der erste Teil der Proben-Identifikationsnummer (ID) besteht aus der QS-Standortnummer des beprobten Standortes. Der zweite Teil (max. 10 Stellen) kann individuell vergeben werden, muss aber eindeutig sein und wird beim Upload überprüft. (z.B.: F00012345-0047110815)
Wird in diesem Feld keine Proben-ID eingetragen, vergibt die Datenbank automatisch eine Nummer.

IdProbenart: Unterscheidung zwischen „regulärer“ und „freiwilliger“ Probe. (1 = Reguläre Probe (QS-Kontrollplan); 2 = Freiwillige Probe)

8 Ihre Ansprechpartner zur Spezifikation

QS Qualität und Sicherheit GmbH

Herr Adam Nowroth

Tel. +49 (0) 228 35068-191

adam.nowroth@q-s.de

Frau Dr. Weber-Jonkheer

Tel. +49 (0) 228 35068-190

ragnhild.weber-jonkheer@q-s.de