



UID-Register

Webservice Schnittstelle 5.0

Versionierung

| Datum | Version | Autor | Bemerkung |
|----------|---------|----------------------|---------------------------------------|
| 01.03.18 | 0.1 | Stefan Schenkel, BFS | Initialversion für Partner Feedback |
| 06.06.18 | 0.2 | Stefan Schenkel, BFS | Anpassungen V5.0 |
| 27.06.18 | 0.3 | Stefan Schenkel, BFS | Feedback aus Review FRM eingearbeitet |
| 15.10.18 | 0.4 | Marc Froidevaux, BFS | Korrekturen |
| 14.11.19 | 0.5 | Marc Froidevaux, BFS | Anpassungen SSL-Zertifikate |

Inhalt

| | | |
|------------|--|-----------|
| 1 | UID-Webservice | 4 |
| 1.1 | Webservice-URL..... | 4 |
| 1.2 | WSDL | 4 |
| 1.3 | SSL-Zertifikate..... | 4 |
| 1.4 | Client-Zertifikate | 4 |
| 2 | Änderungen gegenüber Version 3.0 | 6 |
| 3 | Datenformat | 8 |
| 3.1 | Unternehmensdaten..... | 8 |
| 3.2 | UID Datentypen | 8 |
| 3.2.1 | uidEntityCreateRequest | 8 |
| 3.2.2 | uidEntityCreateResponse | 9 |
| 3.2.3 | uidEntityUpdateRequest | 9 |
| 3.2.4 | uidEntityUpdateResponse | 10 |
| 3.2.5 | uidEntityDeleteRequest | 10 |
| 3.2.6 | uidEntityDeleteResponse..... | 10 |
| 3.2.7 | uidEntityReactivateRequest..... | 11 |
| 3.2.8 | uidEntityReactivateResponse | 11 |
| 3.2.9 | uidRegisterDeregisterRequest..... | 11 |
| 3.2.10 | registerDeregisterResult | 12 |
| 3.2.11 | uidEntityGetDetailRequest..... | 13 |
| 3.2.12 | uidEntityGetDetailResponse | 14 |
| 3.2.13 | getInfoAboMessagesRequest..... | 14 |
| 3.2.14 | getInfoAboMessagesResponse | 14 |
| 3.2.15 | infoAboMessage | 15 |
| 3.2.16 | uidEntitySearchRequest | 15 |
| 3.2.17 | uidEntityPublicSearchRequest..... | 16 |
| 3.2.18 | uidEntitySearchParameters | 16 |
| 3.2.19 | uidEntityPublicSearchParameters | 18 |
| 3.2.20 | personNameType | 18 |
| 3.2.21 | addressSearchType | 19 |
| 3.2.22 | uidregInformationSearchType..... | 20 |
| 3.2.23 | uidregInformationPublicSearchType..... | 20 |
| 3.2.24 | commercialRegisterInformationSearchType..... | 21 |
| 3.2.25 | vatRegisterInformationSearchType | 21 |
| 3.2.26 | searchConfiguration | 22 |
| 3.2.27 | searchMode | 23 |
| 3.2.28 | uidEntitySearchResponse..... | 23 |
| 3.2.29 | uidEntitySearchResultItem..... | 23 |
| 3.2.30 | scheduledTaskItem | 24 |
| 4 | Public Services | 25 |
| 4.1 | Operationen | 25 |
| 4.1.1 | Beispieldaten beziehen | 25 |
| 4.1.2 | Suche nach UID-Einheiten..... | 25 |
| 4.1.3 | Detailldaten abfragen mit der UID | 26 |
| 4.1.4 | Validieren einer UID | 26 |
| 4.1.5 | Validieren einer MWST-Nummer | 26 |
| 5 | Partner-Services | 28 |
| 5.1 | Operationen..... | 28 |

| | | |
|------------|--|-----------|
| 5.1.1 | Beispieldaten beziehen..... | 28 |
| 5.1.2 | Suche nach UID-Einheiten..... | 28 |
| 5.1.3 | Schnelle Suche nach UID-Einheiten..... | 29 |
| 5.1.4 | UID-Einheiten registrieren..... | 29 |
| 5.1.5 | UID-Einheiten mutieren..... | 31 |
| 5.1.6 | UID-Einheit löschen | 32 |
| 5.1.7 | Gelöschte UID-Einheit reaktivieren..... | 32 |
| 5.1.8 | Gelöschte UID-Einheiten mutieren und reaktivieren | 33 |
| 5.1.9 | Als meldende UID-Stelle registrieren..... | 34 |
| 5.1.10 | Als meldende UID-Stelle deregistrieren..... | 34 |
| 5.1.11 | InfoAbo einer UID-Einheit abonnieren | 35 |
| 5.1.12 | InfoAbo einer UID-Einheit abbestellen..... | 35 |
| 5.1.13 | InfoAbo abfragen | 36 |
| 5.1.14 | Abfragen der Resultate von Hintergrundverarbeitungen | 37 |
| 5.1.15 | Datenabgleich mit Zefix anstossen | 38 |
| 6 | Terminierte Meldungen | 39 |
| 6.1 | Datenvalidierung | 39 |
| 6.2 | Rückmeldung | 39 |
| 6.3 | Terminierte Meldungen mutieren | 39 |
| 6.4 | Operationen für terminierte Meldungen | 39 |
| 6.4.1 | Mutation vorerfassen | 39 |
| 6.4.2 | Löschung vorerfassen..... | 40 |
| 6.4.3 | Registrierung vorerfassen..... | 40 |
| 6.4.4 | Deregistrierung vorerfassen..... | 40 |
| 6.4.5 | Pendente Meldungen abfragen..... | 41 |
| 6.4.6 | Pendente Meldungen löschen | 41 |
| 7 | Fehlermeldungen | 43 |
| 7.1 | SecurityFault | 43 |
| 7.2 | BusinessFault | 43 |
| 7.3 | DuplicateFault | 44 |
| 7.4 | InfrastructureFault | 45 |
| | Anhang A: Mappingtabelle | 46 |
| | Anhang B: Testen mit SoapUI | 48 |
| | Neues Projekt anlegen | 48 |
| | Credentials für Partner-Services mitgeben..... | 49 |

1 UID-Webservice

Der UID Webservice besteht aus den Public-Services und den Partner-Services.

In den *Public-Services* bietet das UID-Register Funktionen an, welche auch nicht registrierten Benutzern zur Verfügung stehen. Der Aufruf dieser Funktionen erfolgt demzufolge ohne Login.

Für die Nutzung der *Partner-Services* benötigen Sie einen Benutzer-Account mit entsprechenden Berechtigungen. Benutzer mit Webservice-Berechtigung erkennen Sie am Suffix „_sa“.

Falls Sie noch über keinen entsprechenden Benutzeraccount verfügen wenden Sie sich bitte an den UID Helpdesk uid@bfs.admin.ch

1.1 Webservice-URL

Testsystem

Public-Services <https://www.uid-wse-a.admin.ch/V5.0/PublicServices.svc>
Partner-Services <https://www.uid-wse-a.admin.ch/V5.0/PartnerServices.svc>

Produktivsystem

Public-Services <https://www.uid-wse.admin.ch/V5.0/PublicServices.svc>
Partner-Services <https://www.uid-wse.admin.ch/V5.0/PartnerServices.svc>

1.2 WSDL

Die WSDL zur Anbindung des Webservice kann durch Anfügen von „?wsdl“ automatisch generiert werden

Testsystem

Public-Services <https://www.uid-wse-a.admin.ch/V5.0/PublicServices.svc?wsdl>
Partner-Services <https://www.uid-wse-a.admin.ch/V5.0/PartnerServices.svc?wsdl>

Produktivsystem

Public-Services <https://www.uid-wse.admin.ch/V5.0/PublicServices.svc?wsdl>
Partner-Services <https://www.uid-wse.admin.ch/V5.0/PartnerServices.svc?wsdl>

1.3 SSL-Zertifikate

Die Kommunikation mit dem Webservice erfolgt über einen gesicherten https-Kanal. Die dazu verwendeten SSL-Zertifikate werden auf den Loadbalancern der Bundesverwaltung durch das Bundesamt für Informatik (BIT) installiert und verwaltet. Das BFS als Betreiber des UID-Registers hat keine Kontrolle über diese Zertifikate.

Bitte stellen Sie sicher, dass die entsprechenden Root-Zertifikate im Truststore Ihrer Anwendung installiert sind und jeweils rechtzeitig vor deren Ablaufdatum erneuert werden.

Die verwendeten Zertifikate sind in Ihrem Browser sichtbar.

1.4 Client-Zertifikate

Die UID Partnerservices können auch über den *sedex Webservice Proxy* aufgerufen werden. Aus diesem Grund wird beim Aufruf eines Services in der Domain www.uid-wse.admin.ch nach einem Client-Zertifikat gefragt. Clients welche sich ohne sedex WS Proxy mit den UID Webservices verbinden, kön-

nen diese Anfrage ignorieren und sich mit Username/Passwort anmelden. Alternativ können die Services auf dem Port 8443 aufgerufen werden, auf diesem Port wird keine Zertifikatsanfrage gemacht.

Anmerkung: Das UID-Register selber vergibt keine Client-Zertifikate. Diese Form der Authentifizierung ist ausschliesslich den sedex Kunden vorbehalten.

Weitere Informationen zum sedex WS Proxy finden Sie auf <http://www.sedex.ch> im Bereich *Synchron*.

2 Änderungen gegenüber Version 3.0

Der neue Webservice basiert auf dem aktuellen eCH-Standard eCH-0108 V 5.0 und dessen Referenzen (siehe Kapitel 3 Datenformat)

Die Version 4.0 wurde übersprungen, weil der Standard kurzfristig nochmal erweitert werden musste.

Housekeeping

- Mit der neuen Version der UID-Webservices, werden nicht mehr die forgiving Versionen der eCH-Standards verwendet.
- Die in der Vorgängerversion als obsolet gekennzeichnete Operation *GetInfoAboChanges* wird nicht mehr angeboten
- Der Legacy-Endpunkt *Service.svc* wird nicht mehr angeboten.

Änderungen an der Schnittstelle

- Die Parameter und Rückgabewerte aller Operationen sind neu in einer Request-/Response Struktur gewrappt. (z.B. *uidEntityCreateRequest*, *uidEntityCreateResponse*). Diese enthalten alle eine *extension* vom Typ *xs:anyType*, welche bei Bedarf für kurzfristige Erweiterungen oder partnerspezifische Zusätze verwendet werden kann.
- Da nicht mehr die forgiving Standards verwendet werden ist die UID neu ein Pflichtfeld. Um eine neue UID-Einheit zu erstellen muss daher beim *Create* als UID CHE-000.000.001 angegeben werden.
- Beim *Create* wird neu eine automatische Doublettenprüfung durchgeführt. Potentielle Duplikate werden als *DuplicateFault* zusammen mit eine *OverrideCode* zurückgemeldet. Mit dem *OverrideCode* kann die Doublettenprüfung übersteuert und der Eintrag der neuen Einheit forciert werden.
- Die *Delete* Operation verlangt neu nur noch die UID der zu löschenden UID-Einheit und den Lösungsgrund (bisher: komplette Organisationsdaten)
- Die Operation *GetInfoAboMessagesByType* wurde entfernt. An ihrer Stelle kann neu mit der Operation *GetInfoAboMessages* über den optionalen Filter *messageType* im *getInfoAboMessagesRequest* das gleiche Resultat erreicht werden.

Änderungen in der Datenstruktur

- Die Adressen werden neu als *eCH-0098:organisationAddressType* abgebildet (bisher: *eCH-0046:addressType*). Der neue Typ fasst alle im UID-Register geführten Adressfelder zusammen und enthält zusätzlich auch die [EGID](#). Die Felder *eCH-0108:organisationMunicipality*, *eCH-0108:cantonAbbreviationMain* und *eCH-0108:cantonAbbreviationAdditional* sind obsolet und wurden aus dem Standard entfernt.
- Die *addressCategories* wurden entsprechend ihrer Bedeutung umbenannt. Es gibt neu folgende Adresskategorien:
 - LEGAL (bisher: main)
 - BUR (bisher: additional)
 - POBOX (bisher: postbox)
- Über *eCH-0108:involvedPerson* können einer UID-Einheit neu Personendaten mit Name, Vorname, AHV Nummer, Geburtsdatum und Geschlecht zugewiesen werden. Der Standard sieht 0-n Personen vor. Im UID-Register wird *involvedPerson* bis auf weiteres

nur für den Inhaber/die Inhaberin von UID-Einheiten mit Rechtsform *Einzelunternehmen* genutzt werden.

Personendaten mit einer AHV Nummer werden mit UPI abgeglichen.

- Das bisher für das Geburtsdatum missbrauchte Feld `eCH-0098:foundationDate` steht neu wirklich für das Gründungsdatum des Unternehmens.
Das Gründungsdatum wird ausschliesslich vom UID-Register gesetzt und muss von den UID-Stellen nicht gemeldet werden.
- Bei den Informationsquellen (`eCH-0108:uidRegSources`) werden neu neben der UID auch Typ (zuständig, registriert) und Status (aktiv, `WaitingForSuccessor`) angezeigt
Zudem wird der abfragenden UID-Einheit auch eine allfällige passive Registrierung (`InfoAbo-Subscription`) aufgelistet.
- Über den neuen Typ `eCH-0108:groupRelationshipType` können nun Organisationstrukturen abgebildet werden. Es sind aktuell 3 Typen definiert:
HQ_BRANCH: Beziehung zwischen Hauptsitz und Zweigniederlassungen
HQ_BRANCH_INACTIVE: Beziehungen vom Hauptsitz zu nicht mehr aktiven Zweigniederlassungen
VATGROUP: Mehrwertsteuergruppen
- Bei UID-Einheiten die über einen LEI (Legal Entity Identifier) verfügen, werden der LEI und dessen Status angezeigt (`eCH-0108:leiRegisterInformation`)
- UID-Stellen sind neu am Flag `eCH-0108:uidregUidService` erkennbar.

Änderungen in der Suche

- Die Suchparameter haben neu den Typ `uidwse:uidEntitySearchRequest` (bisher `eCH-0108:organisationType`). Dieser neue Typ erlaubt über einen XSD-Choice die Suche nach **UID** **oder** *OtherOrganisationId* (HR, LEI, GLN etc) **oder** *AVH-Nr* **oder** *freie Suche*. Damit ist die bisher implizit angewandte Suchlogik nun auch in der Schnittstelle abgebildet.
Die freie Suche erlaubt wie bisher die Suche nach beliebigen Kombinationen von Parametern.
- Bei der freien Suche ist neu für die Rechtsform und den Unternehmenstyp eine Mehrfachauswahl möglich.
- Es gibt neu eine *QuickSearch*, welche nur die Kernmerkmale (UID, Name, Hauptadresse, Status) zurückgibt. Dies bietet eine erhebliche Performanceverbesserung gegenüber der normalen Suche, welche für jeden Treffer die komplette Objekthierarchie aufbauen muss.
Die "normale" Search-Operation liefert weiterhin vollständige Datensätze im bisherigen Umfang zurück.
- Die vollständigen Daten eines Einzelrecords können bei Bedarf über die neue Operation *GetOrganisationDetails* abgefragt werden.
- Die Suche berücksichtigt zukünftig auf Wunsch auch frühere Firmennamen und Adressen.
Das Suchresultat zeigt immer die aktuellen Daten, Treffer die auf historischen Daten beruhen werden entsprechend markiert.
Diese Funktionalität wird beim Aufschalten des neuen Webservice noch nicht implementiert sein. Sie wird auf der Schnittstelle trotzdem bereitgestellt, so kann sie später im UID-Register aktiviert werden, ohne dass die Clients erneut angepasst werden müssen.
- Die Suche liefert neu den `uidEntitySearchResponse` mit einer Liste von `uidEntitySearchResultItem` zurück (bisher *ratedOrganisation*).

Neue Operationen

QuickSearch

Schnelle Suche die nur die Kernmerkmale zurückliefert

GetOrganisationDetails

Abfrage aller Daten anhand einer UID. Es können mehrere UIDs angegeben werden

Bem. vermutlich werden wir das aus Performancegründen auf 100-200 Einheiten pro Abfrage beschränken müssen.

UpdateAndReactivate

Ermöglicht es eine gelöschte UID-Einheit zu reaktivieren und gleichzeitig die Daten der Einheit zu ändern.

3 Datenformat

3.1 Unternehmensdaten

Um den Austausch von Unternehmensdaten zwischen Registern und ihren Partnern zu vereinheitlichen, wurden vom *Verein eCH* entsprechende Standards erarbeitet.

Die Basis für die Kommunikation mit dem UID Register bildet der Standard [eCH-0108: Datenstandard Unternehmens-Identifikationsregister \(UID-Register\)](#).

Für den UID-Webservice V5.0 gelten die folgenden Standards und Versionen

| Standard | Version | Inhalt |
|--------------------------|---------|--|
| eCH-0108 | 5.0 | Datenstandard UID-Register |
| eCH-0098 | 5.0 | Datenstandard Unternehmensdaten |
| eCH-0097 | 4.0 | Datenstandard Unternehmensidentifikation |
| eCH-0046 | 5.0 | Datenstandard Kontakt |
| eCH-0044 | 4.1 | Datenstandard Austausch von Personenidentifikationen |
| eCH-0010 | 7.0 | Datenstandard Postadresse für natürliche Personen, Firmen, Organisationen und Behörden |
| eCH-0008 | 3.0 | Datenstandard Staaten und Gebiete |
| eCH-0007 | 6.0 | Datenstandard Gemeinde |

Die Tabelle im *Anhang A*: Mappingtabelle zeigt die Zuordnung der eCH-Felder auf die UID Daten.

Aktuelle Informationen und Dokumentationen zum eCH-0108 und den darin referenzierten Standards sind auf der Webseite des für die Standards verantwortlichen Vereins eCH <http://www.ech.ch> zu finden.

3.2 UID Datentypen

Ergänzend zu den von eCH formulierten Unternehmensdaten, verwendet der UID Webservice eigene Datentypen, welche im Folgenden kurz beschrieben werden.

3.2.1 uidEntityCreateRequest

```
<xs:complexType name="uidEntityCreateRequest">  
  <xs:sequence>
```

```

<xs:element name="organisation" type="eCH-0108:organisationType" />
<xs:element name="duplicateOverrideCode" type="xs:token" minOccurs="0" />
<xs:element name="extension" type="xs:anyType" minOccurs="0" />
</xs:sequence>
</xs:complexType>

```

| Element | Datentyp | Kardinalität |
|-----------------------|---|--------------|
| | Beschreibung | |
| organisation | eCH-0108:organisationType | 1 |
| | Vollständige Organisationsdaten der neuen UID-Einheit mit UID = CHE-000.000.001 | |
| duplicateOverrideCode | xs:token | 0-1 |
| | Mit dem OverrideCode kann die Doublettenprüfung des UID-Registers übersteuert und der Eintrag der neuen UID-Einheit forciert werden | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.2 uidEntityCreateResponse

```

<xs:complexType name="uidEntityCreateResponse">
  <xs:sequence>
    <xs:element name="organisation" type="eCH-0108:organisationType" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

```

| Element | Datentyp | Kardinalität |
|--------------|--|--------------|
| | Beschreibung | |
| organisation | eCH-0108:organisationType | 1 |
| | Die vollständige UID-Einheit, wie sie nach dem Speichern im UID-Register abgelegt ist. In diesen Daten ist auch die neu zugewiesene UID enthalten. Falls der UID-Status auf „provisorisch“ gesetzt wurde, wird die meldende UID-Stelle per InfoAbo über das Resultat der Validierung informiert. | |
| extension | xs:anyType | 0-1 |
| | [optional] Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.3 uidEntityUpdateRequest

```

<xs:complexType name="uidEntityUpdateRequest">
  <xs:sequence>
    <xs:element name="organisation" type="eCH-0108:organisationType" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>

```

| Element | Datentyp | Kardinalität |
|--------------|---|--------------|
| | Beschreibung | |
| organisation | eCH-0108:organisationType | 1 |
| | Vollständige Organisationsdaten der zu ändernden UID-Einheit. | |
| extension | xs:anyType | 0-1 |

| | |
|--|--|
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen |
|--|--|

3.2.4 uidEntityUpdateResponse

```
<xs:complexType name="uidEntityUpdateResponse">
  <xs:sequence>
    <xs:element name="organisation" type="eCH-0108:organisationType" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|--------------|--|--------------|
| | Beschreibung | |
| organisation | eCH-0108:organisationType | 1 |
| | Die vollständige UID-Einheit, wie sie nach dem Speichern im UID-Register abgelegt ist. Falls der UID-Status auf „in Mutation“ gesetzt wurde, wird die meldende UID-Stelle per InfoAbo über das Resultat der Validierung informiert. | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.5 uidEntityDeleteRequest

```
<xs:complexType name="uidEntityDeleteRequest">
  <xs:sequence>
    <xs:element name="uid" type="eCH-0097:uidStructureType" />
    <xs:element name="deleteReason" type="eCH-0108:uidregLiquidationReasonType" />
    <xs:element name="uidReplacement" type="eCH-0097:uidStructureType" minOccurs="0" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|----------------|--|--------------|
| | Beschreibung | |
| uid | eCH-0097:uidStructureType | 1 |
| | UID der zu löschenden UID-Einheit | |
| deleteReason | eCH-0108:uidregLiquidationReasonType | 1 |
| | Löschungsgrund | |
| uidReplacement | eCH-0097:uidStructureType | 0-1 |
| | Massgebende UID einer gelöschten Doublette Diese Angabe ist Pflicht wenn der Lösungsgrund 9 (Doublette) angegeben wird. | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.6 uidEntityDeleteResponse

```
<xs:complexType name="uidEntityDeleteResponse">
  <xs:sequence>
    <xs:element name="organisation" type="eCH-0108:organisationType" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|--------------|--|--------------|
| | Beschreibung | |
| organisation | eCH-0108:organisationType | 1 |
| | Die vollständige UID-Einheit, wie sie nach dem Speichern im UID-Register abgelegt ist. Falls der UID-Status auf „in Mutation“ gesetzt wurde, wird die meldende UID-Stelle per InfoAbo über das Resultat der Validierung informiert. | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.7 uidEntityReactivateRequest

```
<xs:complexType name="uidEntityDeleteResponse">
  <xs:sequence>
    <xs:element name="organisation" type="eCH-0108:organisationType" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|-----------|--|--------------|
| | Beschreibung | |
| uid | eCH-0097:uidStructureType | 1 |
| | UID der zu reaktivierenden UID-Einheit | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.8 uidEntityReactivateResponse

```
<xs:complexType name="uidEntityReactivateRequest">
  <xs:sequence>
    <xs:element name="uid" type="eCH-0097:uidStructureType" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|-----------|--|--------------|
| | Beschreibung | |
| uid | eCH-0097:uidStructureType | 1 |
| | UID der zu reaktivierenden UID-Einheit. Falls der UID-Status auf „in Reaktivierung“ gesetzt wurde, wird die meldende UID-Stelle per InfoAbo über das Resultat der Validierung informiert. | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.9 uidRegisterDeregisterRequest

```
<xs:complexType name="uidRegisterDeregisterRequest">
  <xs:sequence>
    <xs:element name="uid" type="eCH-0097:uidStructureType" maxOccurs="unbounded" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
```

```
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|-----------|--|--------------|
| | Beschreibung | |
| uid | eCH-0097:uidStructureType | 1-n |
| | UID der UID-Einheit/en für welche die UID-Stelle registriert werden soll | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.10 registerDeregisterResult

Das *registerDeregisterResult* gilt auch für die Operationen Subscribe und Unsubscribe. Diese Struktur wird sowohl als Bestätigung bei einer eingehenden Meldung (Register etc.), wie auch bei der Abfrage der Resultate (GetRegisterResult etc.) verwendet.

```
<xs:complexType name="registerDeregisterResult">
  <xs:sequence>
    <xs:element name="taskId" type="xs:int" />
    <xs:element name="items" type="uid-wse:registerDeregisterItem" minOccurs="0" max-
Occurs="unbounded" />
    <xs:element name="remark" type="xs:string" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|---------|--|--------------|
| taskId | xs:int | 1 |
| | Id für spätere Abfrage der Resultate mittels der entsprechenden Get-Result Operation | |
| Items | registerDeregisterItem - Array | 0-1 |
| | Liste der verarbeiteten UID-Einheiten | |
| remark | xs:string | 0-1 |
| | Ergänzende Informationen in Textform (nur englisch). | |

3.2.10.1 registerDeregisterItem

Für jede verarbeitete UID-Einheit enthält das Resultat ein *registerDeregisterItem* mit der UID der Einheit sowie dem Status der Verarbeitung.

```
<xs:complexType name="registerDeregisterItem">
  <xs:sequence>
    <xs:element name="uid" type="eCH-0097:uidStructureType" />
    <xs:element name="status" type="uid-wse-shared:registerDeregisterStatus" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|---------|---|--------------|
| uid | eCH-0097:uidStructureType | 1 |
| | Die UID der zu verarbeitenden UID-Einheit | |
| status | registerDeregisterStatus | 1 |
| | Status der Verarbeitung | |

3.2.10.2 registerDeregisterStatus

Der *registerDeregisterStatus* gilt auch für die Operation *Subscribe* und *Unsubscribe*. Nachfolgende Status sind definiert:

| Wert | Beschreibung |
|-------------------------------------|--|
| AlreadyRegistered | Nur Register und Subscribe: Die UID-Stelle ist bei dieser UID-Einheit bereits registriert. |
| AlreadyRegisteredAsActiveUIDService | Nur Subscribe: Die meldende Stelle ist bei dieser UID-Einheit bereits als aktive UID-Stelle registriert und somit automatisch auch für das InfoAbo abonniert |
| DeletionRequired | Nur Deregister: Anstelle der gewünschten Deregistrierung muss eine Löschung beantragt werden. Dies ist der Fall wenn die meldende Stelle die zuständige UID-Stelle ist. |
| Duplicate | Die Einheit wurde in dieser Anfrage bereits verarbeitet. |
| InvalidParameter | Nur bei Registrierung mittels CSV auf dem WebGUI möglich: Das Format der angegebenen UID ist ungültig |
| NotRegistered | Nur Deregister und Unsubscribe: Die meldende Stelle war bei dieser Einheit nicht registriert. |
| ReactivationRequired | Nur Register und Subscribe: Die UID-Einheit ist nicht aktiv. Für eine Registrierung muss die Einheit vorher reaktiviert werden. |
| RegistrationNotAllowed | Nur Register: Diese Registrierung ist nicht zulässig. Dies ist der Fall wenn die meldende Stelle keine UID-Stelle ist oder versucht sich für sich selbst zu registrieren. |
| Success | Die Aufgabe wurde erfolgreich ausgeführt. |
| SystemError | Bei der Verarbeitung ist ein technisches Problem aufgetreten. Falls dieser Fehler wiederholt auftritt, melden Sie dies bitte dem UID Helpdesk |
| UidNotFound | Es wurde keine Einheit mit der angegebenen UID gefunden. |

3.2.11 uidEntityGetDetailRequest

```
<xs:complexType name="uidEntityGetDetailRequest">
  <xs:sequence>
    <xs:element name="uid" type="eCH-0097:uidStructureType" maxOccurs="unbounded" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|-----------|--|--------------|
| | Beschreibung | |
| uid | eCH-0097:uidStructureType | 1-n* |
| | UID der UID-Einheit/en von welchen die vollständigen Details geladen werden sollen * n ist aus Performancegründen auf 100 Einheiten begrenzt. Im Schema bleibt die Obergrenze mit "unbounded" offen um die Limite ohne Schemaanpassungen ändern zu können | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.12 uidEntityGetDetailResponse

```
<xs:complexType name="uidEntityGetDetailResponse">
  <xs:sequence>
    <xs:element name="organisation" type="eCH-0108:organisationType" maxOccurs="unbounded"
  />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|--------------|--|--------------|
| | Beschreibung | |
| organisation | eCH-0108:organisationType | 1-n |
| | Vollständige Details der abgefragten UID-Einheiten. | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.13 getInfoAboMessagesRequest

```
<xs:complexType name="getInfoAboMessagesRequest">
  <xs:sequence>
    <xs:element name="dateFrom" type="xs:date" />
    <xs:element name="dateTo" type="xs:date" />
    <xs:element name="messageType" type="eCH-0116:uidInfoAboMessageTypeType" minOccurs="0"
  maxOccurs="unbounded" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|-------------|--|--------------|
| | Beschreibung | |
| dateFrom | xs:date | 1 |
| | Untere Grenze der Abfrageperiode | |
| dateTo | xs:date | 1 |
| | Obere Grenze der Abfrageperiode | |
| messageType | eCH-0116:uidInfoAboMessageTypeType | 0-n |
| | Optionale Einschränkung auf einzelne Nachrichtentypen | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.14 getInfoAboMessagesResponse

```
<xs:complexType name="getInfoAboMessagesResponse">
  <xs:sequence>
    <xs:element name="infoAboMessage" type="uid-wse:infoAboMessage" maxOccurs="unbounded"
  />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|----------------|--|--------------|
| | Beschreibung | |
| infoAboMessage | infoAboMessage | 0-n |
| | Liste aller InfoAbo-Meldungen im abgefragten Zeitraum | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.15 infoAboMessage

Diese Struktur repräsentiert eine einzelne InfoAbo Meldung. Die InfoAbo-Meldung enthält immer den aktuellen Gesamtdatensatz der entsprechenden UID-Einheit. Jede Meldung ist über die messageId (UUID) eindeutig identifizierbar.

```
<xs:complexType name="infoAboMessage">
  <xs:sequence>
    <xs:element name="messageId" type="eCH-0058:messageIdType" />
    <xs:element name="reportingRegister" type="eCH-0097:uidStructureType" />
    <xs:element name="organisation" type="eCH-0108:organisationType" />
    <xs:element name="messageType" type="eCH-0116:uidInfoAboMessageTypeType" />
    <xs:element name="eventDate" type="eCH-0058:eventDateType" />
    <xs:element name="diff" type=" " minOccurs="0" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Beschreibung |
|-------------------|------------------------------------|--|
| messageId | eCH-0058:messageIdType | Eindeutiger Identifikator (UUID) |
| reportingRegister | eCH-0097:uidStructureType | UID der meldenden UID-Stelle |
| Organisation | eCH-0108:organisationType | Aktuelle Daten der geänderten UID-Einheit |
| messageType | eCH-0116:uidInfoAboMessageTypeType | Meldungstyp |
| eventDate | eCH-0058:eventDateType | Datum der Registrierung/Mutation im UID-Register |
| diff | | Auflistung der geänderten Felder mit vorher/nachher Werten |
| extension | Freitext | [optional] Platzhalter für allfällige Ad-Hoc Erweiterungen |

3.2.16 uidEntitySearchRequest

Die Suchanfrage in den PartnerServices erlaubt die Suche nach UID oder otherOrganisationId oder AHV-Nummer oder eine freie Suche.

```
<xs:complexType name="uidEntitySearchRequest">
  <xs:choice>
    <xs:element name="uid" type="eCH-0097:uidStructureType" />
    <xs:element name="otherOrganisationId" type="eCH-0097:namedOrganisationIdType" />
    <xs:element name="vn" type="eCH-0108:vnType" />
    <xs:element name="uidEntitySearchParameters" type="uid-wse:uidEntitySearchParameters" />
  </xs:choice>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|---------|---------------------------|--------------|
| | Beschreibung | |
| uid | eCH-0097:uidStructureType | 0-1* |

| | | |
|---------------------------------|--|------|
| | Gezielte Suche nach einer UID | |
| otherOrganisationId | 0097:namedOrganisationIdType | 0-1* |
| | Suche nach einer OtherOrganisationId (CH.HR, WW.GLN etc) | |
| vn | eCH-0108:vnType | 0-1* |
| | Suche nach AHV Nummer. Nur möglich für UID-Stellen welche die AHV-Nummer verwenden dürfen. Andernfalls wird ein BusinessFault vom Typ <i>Unauthorized</i> geworfen. | |
| uidEntityPublicSearchParameters | uidEntitySearchParameters | 0-1* |
| | Parameter für freie Suche. Alle Felder innerhalb dieser Struktur sind optional und können beliebig kombiniert werden. Die ausgefüllten Felder werden UND verknüpft. | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

* Der xsl:choice erfordert einen der 4 Parameter

3.2.17 uidEntityPublicSearchRequest

Die Suchanfrage in den PublicServices erlaubt die Suche nach UID oder otherOrganisationId oder eine freie Suche.

```
<xs:complexType name="uidEntitySearchPublicRequest">
  <xs:choice>
    <xs:element name="uid" type="eCH-0097:uidStructureType" />
    <xs:element name="otherOrganisationId" type="eCH-0097:namedOrganisationIdType" />
    <xs:element name="uidEntitySearchParameters" type="uid-wse:uidEntityPublicSearchParameters" />
  </xs:choice>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|---------------------------------|---|--------------|
| | Beschreibung | |
| uid | eCH-0097:uidStructureType | 0-1* |
| | Gezielte Suche nach einer UID | |
| otherOrganisationId | 0097:namedOrganisationIdType | 0-1* |
| | Suche nach einer OtherOrganisationId (CH.HR, WW.GLN etc) | |
| uidEntityPublicSearchParameters | uidEntityPublicSearchParameters | 0-1* |
| | Parameter für freie Suche. Alle Felder innerhalb dieser Struktur sind optional und können beliebig kombiniert werden. Die ausgefüllten Felder werden UND verknüpft. | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

* Der xsl:choice erfordert einen der 3 Parameter

3.2.18 uidEntitySearchParameters

```
<xs:complexType name="uidEntitySearchParameters">
  <xs:sequence>
```

```

</xs:element name="organisationName" type="eCH-0097:organisationNameType" minOccurs="0"
/>
<xs:element name="personName" type="uid-wse:personNameType" minOccurs="0" />
<xs:element name="dateOfBirth" type="eCH-0098:datePartiallyKnownType" minOccurs="0" />
<xs:element name="address" type="uid-wse:addressSearchType" minOccurs="0" />
<xs:element name="legalForm" type="eCH-0097:legalFormType" minOccurs="0" maxOccurs="un-
bounded" />
<xs:element name="NOGACode" type="eCH-0098:NOGACodeType" minOccurs="0" />
<xs:element name="emailAddress" type="eCH-0046:emailAddressType" minOccurs="0" />
<xs:element name="internetAddress" type="eCH-0046:internetAddressType" minOccurs="0" />
<xs:element name="languageOfCorrespondance" type="eCH-0098:languageType" minOccurs="0"
/>
</xs:element name="uidregInformation" type="uid-wse:uidregInformationSearchType" minOc-
curs="0" />
<xs:element name="commercialRegisterInformation" type="uid-wse:commercialRegisterInfor-
mationSearchType" minOccurs="0" />
<xs:element name="vatRegisterInformation" type="uid-wse:vatRegisterInformationSe-
archType" minOccurs="0" />
</xs:sequence>
</xs:complexType>

```

| Element | Datentyp | Kardinalität |
|------------------------------------|---|--------------|
| | Beschreibung | |
| organisationName* | eCH-0097:organisationNameType | 0-1 |
| | Name der gesuchten UID-Einheit | |
| personName* | personNameType | 0-1 |
| | Name der gesuchten Person | |
| dateOfBirth | eCH-0098:datePartiallyKnownType | 0-1 |
| | Geburtsdatum der gesuchten Person | |
| Address | addressSearchType | 0-1 |
| | Adresse der gesuchten UID-Einheit | |
| legalForm | eCH-0097:legalFormType | 0-n |
| | Zu berücksichtigende Rechtsform/en | |
| NOGACode | eCH-0098:NOGACodeType | 0-1 |
| | | |
| emailAddress | eCH-0046:emailAddressType | 0-1 |
| | Suche nach einer Email-Adresse | |
| internetAddress | eCH-0046:internetAddressType | 0-1 |
| | Suche nach einer Internetadresse | |
| languageOfCorrespon- dance | eCH-0098:languageType | 0-1 |
| | | |
| uidregInformation | uidregInformationSearchType | 0-1 |
| | | |
| commercialRegisterIn- formation | commercialRegisterInformationSearchType | 0-1 |
| | | |
| vatRegisterInformation | vatRegisterInformationSearchType | 0-1 |
| | | |

* Die Kombination von Organisationname und Personname ist grundsätzlich möglich, wird aber in den meisten Fällen keinen Sinn machen. Hinter den beiden Feldern stecken unterschiedliche Suchalgorithmen. Für optimale Suchergebnisse empfehlen wir die beiden Parameter gezielt zu wählen.

3.2.19 uidEntityPublicSearchParameters

```
<xs:complexType name="uidEntityPublicSearchParameters">
  <xs:sequence>
    <xs:element name="organisationName" type="eCH-0097:organisationNameType" minOccurs="0"
  />
    <xs:element name="personName" type="uid-wse:personNameType" minOccurs="0" />
    <xs:element name="address" type="uid-wse:addressSearchType" minOccurs="0" />
    <xs:element name="legalForm" type="eCH-0097:legalFormType" minOccurs="0" maxOccurs="un-
bounded" />
    <xs:element name="uidregInformation" type="uid-wse:uidregInformationPublicSearchType"
minOccurs="0" />
    <xs:element name="commercialRegisterInformation" type="uid-wse:commercialRegisterInfor-
mationSearchType" minOccurs="0" />
    <xs:element name="vatRegisterInformation" type="uid-wse:vatRegisterInformationSearch-
Type" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|------------------------------------|---|--------------|
| | Beschreibung | |
| organisationName* | eCH-0097:organisationNameType | 0-1 |
| | Name der gesuchten UID-Einheit | |
| personName* | personNameType | 0-1 |
| | Name der gesuchten Person | |
| address | addressSearchType | 0-1 |
| | Adresse der gesuchten UID-Einheit | |
| legalForm | eCH-0097:legalFormType | 0-n |
| | Zu berücksichtigende Rechtsform/en | |
| uidregInformation | uidregInformationPublicSearchType | 0-1 |
| | | |
| commercialRegisterIn- formation | commercialRegisterInformationSearchType | 0-1 |
| | | |
| vatRegisterInformation | vatRegisterInformationSearchType | 0-1 |
| | | |

* Die Kombination von Organisationname und Personname ist grundsätzlich möglich, wird aber in den meisten Fällen keinen Sinn machen. Hinter den beiden Feldern stecken unterschiedliche Suchalgorithmen. Für optimale Suchergebnisse empfehlen wir die beiden Parameter gezielt zu wählen.

3.2.20 personNameType

```
<xs:complexType name="personNameType">
  <xs:sequence>
    <xs:element name="officialName" type="eCH-0108:baseNameType" />
    <xs:element name="firstName" type="eCH-0108:baseNameType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|--------------|--|--------------|
| | Beschreibung | |
| officialName | eCH-0108:baseNameType | 1 |
| | Nachname | |
| | Dies ist ein Pflichtfeld, die Suche nur nach Vorname ist nicht möglich | |

| | | |
|-----------|-----------------------|------|
| firstname | eCH-0108:baseNameType | 0-1* |
| | Vorname | |

3.2.21 addressSearchType

```

<xs:complexType name="addressSearchType">
  <xs:sequence>
    <xs:element name="addressLine1" type="eCH-0010:addressLineType" minOccurs="0"/>
    <xs:element name="addressLine2" type="eCH-0010:addressLineType" minOccurs="0"/>
    <xs:element name="street" type="eCH-0010:streetType" minOccurs="0"/>
    <xs:element name="houseNumber" type="eCH-0010:houseNumberType" minOccurs="0"/>
    <xs:element name="postOfficeBoxNumber" type="eCH-0010:postOfficeBoxNumberType" minOccurs="0"/>
    <xs:element name="town" type="eCH-0010:townType" minOccurs="0"/>
    <xs:choice>
      <xs:sequence>
        <xs:element name="swissZipCode" type="eCH-0010:swissZipCodeType" minOccurs="0" />
        <xs:element name="swissZipCodeAddOn" type="eCH-0010:swissZipCodeAddOnType" minOccurs="0"/>
        <xs:element name="municipalityId" type="eCH-0007:municipalityIdType" minOccurs="0"/>
        <xs:element name="cantonAbbreviation" type="eCH-0007:cantonAbbreviationType" minOccurs="0"/>
        <xs:element name="EGID" type="eCH-0098:EGIDType" minOccurs="0"/>
      </xs:sequence>
      <xs:element name="foreignZipCode" type="eCH-0010:foreignZipCodeType" minOccurs="0" />
    </xs:choice>
    <xs:element name="countryIdISO2" type="eCH-0010:countryIdISO2Type" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

```

| Element | Datentyp | Kardinalität |
|---------------------|----------------------------------|--------------|
| | Beschreibung | |
| addressLine1 | eCH-0010:addressLineType | 0-1 |
| | | |
| addressLine2 | eCH-0010:addressLineType | 0-1 |
| | | |
| street | eCH-0010:streetType | 0-n |
| | | |
| houseNumber | eCH-0010:houseNumberType | 0-1 |
| | | |
| postOfficeBoxNumber | eCH-0010:postOfficeBoxNumberType | 0-1 |
| | | |
| town | eCH-0010:townType | 0-1 |
| | | |
| swissZipCode | eCH-0010:swissZipCodeType | 0-1 |
| | | |
| swissZipCodeAddOn | eCH-0010:swissZipCodeAddOnType | 0-1 |
| | | |
| municipalityId | eCH-0007:municipalityIdType | 0-1 |
| | | |
| cantonAbbreviation | eCH-0007:cantonAbbreviationType | 0-1 |
| | | |

| | | |
|----------------|-----------------------------|-----|
| EGID | eCH-0098:EGIDType | 0-1 |
| foreignZipCode | eCH-0010:foreignZipCodeType | 0-1 |
| countryIdISO2 | eCH-0010:countryIdISO2Type | 0-1 |

3.2.22 uidregInformationSearchType

```
<xs:complexType name="uidregInformationSearchType">
  <xs:sequence>
    <xs:element name="uidregStatusEnterpriseDetail" type="eCH-0108:uidregStatusEnterprise-
DetailType" minOccurs="0" />
    <xs:element name="uidregPublicStatus" type="eCH-0108:uidregPublicStatusType" minOc-
curs="0" />
    <xs:element name="uidregOrganisationType" type="eCH-0108:uidregOrganisationTypeType"
minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="uidregUidService" type="xs:boolean" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|-----------------------------------|---|--------------|
| | Beschreibung | |
| uidregStatusEnterprise- Detail | eCH-0108:uidregStatusEnterpriseDetail- Type | 0-n |
| | Einschränkung auf einen oder mehrere UID-Status | |
| uidregPublicStatus | eCH-0108:uidregPublicStatusType | 0-1 |
| | Ermöglicht die Eingrenzung auf öffentliche/nicht öffentliche UID-Einheiten true = Es werden nur öffentliche UID-Einheiten gesucht false = Es werden nur nicht-öffentliche UID-Einheiten gesucht null = keine Einschränkung | |
| uidregOrganisationType | eCH-0108:uidregOrganisationTypeType | 0-n |
| | Einschränkung auf Unternehmenstyp/en | |
| uidregUidService | xs:boolean | 0-1 |
| | Ermöglicht die Separierung von UID-Einheiten und UID-Stellen true = Es werden nur UID-Stellen gesucht false = Es werden nur UID-Einheiten gesucht null = keine Einschränkung | |

3.2.23 uidregInformationPublicSearchType

```
<xs:complexType name="uidregInformationPublicSearchType">
  <xs:sequence>
    <xs:element name="uidregStatusEnterpriseDetail" type="eCH-0108:uidregStatusEnterprise-
DetailType" minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="uidregUidService" type="xs:boolean" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|-----------------------------------|---|--------------|
| | Beschreibung | |
| uidregStatusEnterprise- Detail | eCH-0108:uidregStatusEnterpriseDetail- Type | 0-n |
| | Einschränkung auf einen oder mehrere UID-Status | |

| | | |
|------------------|---|-----|
| uidregUidService | xs:boolean | 0-1 |
| | Ermöglicht die Separierung von UID-Einheiten und UID-Stellen true = Es werden nur UID-Stellen gesucht false = Es werden nur UID-Einheiten gesucht null = keine Einschränkung | |

3.2.24 commercialRegisterInformationSearchType

```
<xs:complexType name="commercialRegisterInformationSearchType">
  <xs:sequence>
    <xs:element name="commercialRegisterStatus" type="eCH-0108:commercialRegister-
StatusType" minOccurs="0" />
    <xs:element name="commercialRegisterEntryStatus" type="eCH-0108:commercialRegis-
terEntryStatusType" minOccurs="0" />
    <xs:element name="commercialRegisterEntryDate" type="xs:date" minOccurs="0" />
    <xs:element name="commercialRegisterLiquidationDate" type="xs:date" minOccurs="0" />
    <xs:element name="commercialRegisterEnterpriseType" type="eCH-0108:commercialRegis-
terEnterpriseTypeType" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|--|---|--------------|
| Beschreibung | | |
| commercialRegister- Status | eCH-0108:commercialRegisterStatusType | 0-1 |
| Status des Unternehmens im Handelsregister 1 = unbekannt (kommt im UID-Register nicht vor!) 2 = im HR eingetragen 3 = nicht im HR eingetragen null = keine Einschränkung | | |
| commercialRegisterEnt- ryStatus | eCH-0108:commercialRegisterEntryStatus- | 0-1 |
| Status des Eintrags im HR 1 = aktiv 2 = gelöscht 3 = provisorisch null = keine Einschränkung | | |
| commercialRegis- terEntryDate | eCH-0108:commercialRegisterEntryDate- | 0-1 |
| Einschränkung auf das Datum des Eintrags im HR | | |
| commercialRegisterLiq- uidationDate | eCH-0108:commercialRegisterLiquida- tionDateType | 0-1 |
| Einschränkung auf das Datum der Löschung aus dem HR | | |
| commercialRegisterEn- terpriseType | eCH-0108:commercialRegisterEnterprise- | 0-1 |
| Einschränkung auf Art des Unternehmenssitzes 1 = nur im HR eingetragene Hauptsitze 2 = nur im HR eingetragene Nebensitze null = keine Einschränkung | | |

3.2.25 vatRegisterInformationSearchType

```
<xs:complexType name="vatRegisterInformationSearchType">
  <xs:sequence>
    <xs:element name="vatStatus" type="eCH-0108:vatStatusType" minOccurs="0" />
    <xs:element name="vatEntryStatus" type="eCH-0108:vatEntryStatusType" minOccurs="0"/>
    <xs:element name="vatEntryDate" type="xs:date" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

```

<xs:element name="vatLiquidationDate" type="xs:date" minOccurs="0"/>
<xs:element name="uidVat" type="eCH-0097:uidStructureType" minOccurs="0"/>
</xs:sequence>
</xs:complexType>

```

| Element | Datentyp | Kardinalität |
|-----------------------------------|--|--------------|
| | Beschreibung | |
| vatStatus | eCH-0108:vatStatusType | 0-1 |
| | Einschränkung auf den Status des Unternehmens im MWST-Register 1 = unbekannt (kommt im UID-Register nicht vor!) 2 = nur im MWST-Register eingetragene UID-Einheiten 3 = nur nicht im MWST-Register eingetragene UID-Einheiten null = keine Einschränkung | |
| vatEntryStatus | eCH-0108:vatEntryStatusType | 0-1 |
| | Einschränkung auf den Status des Eintrags im MWST-Register 1 = nur bei der MWST aktive UID-Einheiten 2 = nur bei der MWST gelöschte UID-Einheiten null = keine Einschränkung | |
| vatEntryDate | eCH-0108:vatEntryDateType | 0-1 |
| | Einschränkung auf das Datum des Beginns der MWST Pflicht | |
| commercialRegisterLiquidationDate | eCH-0108:vatLiquidationDateType | 0-1 |
| | Einschränkung auf das Datum der Löschung aus dem MWST Register | |
| uidVat | eCH-0097:uidStructureType | 0-1 |
| | Suche nach MWST Nummer. Die MWST Nummer ist bei den meisten MWST pflichtigen UID-Einheiten identisch mit der UID. Bei Nebensitzen oder Mitgliedern von MWST Gruppen kann die uidVat aber von der UID abweichen. | |

3.2.26 searchConfiguration

Über die searchConfiguration können bei der Suche verschiedene Steuerparameter angegeben werden

```

<xs:complexType name="searchConfiguration">
  <xs:sequence>
    <xs:element name="searchMode" type="uid-wse-shared:searchMode" />
    <xs:element name="maxNumberOfRecords" type="xs:int" />
    <xs:element name="searchNameAndAddressHistory" type="xs:boolean" />
  </xs:sequence>
</xs:complexType>

```

| Element | Datentyp | Kardinalität |
|-----------------------------|--|--------------|
| searchMode | searchMode | 1 |
| | Der gewünschte Suchmodus. | |
| maxNumberOfRecords | xs:int | 1 |
| | Maximale Anzahl anzuzeigender Datensätze. Bei 0 wird das im UID-Register definierte Maximum (aktuell 200) angewendet. | |
| searchNameAndAddressHistory | xs:boolean | 1 |
| | Auf Wunsch wird nach früheren Namen und Adressen gesucht. Die Suche wird hierbei auf die Änderungshistorie der UID-Einheiten ausgeweitet, was entsprechende Performanceeinbußen mit sich bringt. Dieser Modus sollte daher nur bei Bedarf aktiviert werden. | |

3.2.27 searchMode

Es stehen verschiedene Suchmodi zur Auswahl

| Wert | Beschreibung |
|-------------|---|
| Auto | Das System entscheidet selber welche Suche die optimale ist (empfohlen) |
| Normal | Es wird eine „normale“ SQL Suche ausgeführt. |
| Fuzzy | Es wird eine <i>Ähnlichkeitssuche</i> ausgeführt Bei der <i>Ähnlichkeitssuche</i> werden die Suchparameter <i>Name</i> , <i>Strasse</i> , <i>PLZ</i> , <i>Ort</i> nach bestimmten Ähnlichkeitsregeln durchsucht. Allfällige weitere Parameter (z.B. Kanton, UID-Status etc.) werden anschliessend auf diesem Resultat als Filter angewendet. Siehe auch <i>5.1.2 Suche nach UID-Einheiten</i> |
| FuzzyPerson | Es wird eine auf Personennamen optimierte Ähnlichkeitssuche ausgeführt |

3.2.28 uidEntitySearchResponse

```
<xs:complexType name="uidEntitySearchResponse">
  <xs:sequence>
    <xs:element name="uidEntitySearchResultItem" type="uid-wse:uidEntitySearchResultItem"
minOccurs="0" maxOccurs="unbounded" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|----------------------------|--|--------------|
| | Beschreibung | |
| uidEntitySearchResultItems | uidEntitySearchResultItem | 0-n |
| | Liste der gefundenen UID-Einheiten | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.29 uidEntitySearchResultItem

```
<xs:complexType name="uidEntitySearchResultItem">
  <xs:sequence>
    <xs:element name="organisation" type="eCH-0108:organisationType" />
    <xs:element name="rating" type="xs:int" />
    <xs:element name="isHistoryMatch" type="xs:boolean" />
    <xs:element name="extension" type="xs:anyType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

| Element | Datentyp | Kardinalität |
|--------------|---|--------------|
| | Beschreibung | |
| organisation | eCH-0108:organisationType | 1 |
| | Unternehmensdaten der gefundenen UID-Einheit | |
| rating | xs:int / Wertebereich: 0 – 100 | 1 |
| | Die prozentuale Übereinstimmung mit den Suchparametern Bei Suchen die keine Ähnlichkeitssuche ausführen (Suche nach ID, searchMode=Normal) ist das Rating immer 100. | |

| | | |
|----------------|--|-----|
| isHistoryMatch | xs:boolean | 1 |
| | <i>true</i> gibt an, dass der Treffer auf historischen Daten basiert (alte Adresse, früherer Name). Wichtig: <i>organisation</i> enthält immer die aktuellen Daten, es werden nie historische Daten publiziert! | |
| extension | xs:anyType | 0-1 |
| | Platzhalter für Ad-Hoc Erweiterungen oder partnerspezifische Ergänzungen | |

3.2.30 scheduledTaskItem

Terminierte Meldungen werden in Form eines *scheduledTaskItems* ausgegeben. Diese Struktur wird sowohl als Bestätigung bei einer eingehenden Meldung, wie auch bei der Abfrage der pendenten Tasks verwendet.

```
<xs:complexType name="scheduledTaskItem">
  <xs:sequence>
    <xs:element name="taskId" type="xs:unsignedInt" />
    <xs:element name="taskType" type="uid-wse-shared:scheduledTaskType" />
    <xs:element name="scheduledDate" type="xs:date" />
  </xs:sequence>
</xs:complexType>
```

3.2.30.1 scheduledTaskType

Derzeit wird für 4 verschiedene Operationen eine Vorverfassung angeboten. Bei der Abfrage der pendenten Tasks wird dieser Typ zur einfacheren Identifizierung angegeben. Des Weiteren kann er zur selektiven Abfrage von pendenten Tasks verwendet werden.

Mögliche Werte (selbsterklärend):

- Register
- Deregister
- Update
- Delete

4 Public Services

In den Public Services bietet das UID-Register Funktionen an, welche auch nicht registrierten Benutzern zur Verfügung stehen. Der Aufruf dieser Funktionen erfolgt demzufolge ohne Login.

4.1 Operationen

4.1.1 Beispieldaten beziehen

Operation: GetOrganisationSample

Beschreibung

Diese Operation bildet keinen Prozess ab. Sie dient einzig dazu, einen fiktiven Datensatz im Format eCH-0108:organisationType zurückzuliefern. Dieser Datensatz enthält **alle** Felder aus dem UID-Register, welche öffentlich sichtbar sind. Diese Felder sind grösstenteils optional, können also bei einem echten Datensatz leer, resp. null, sein.

Parameter

keine

Rückgabewert

Beispieldatensatz im Format eCH-0108:organisationType

4.1.2 Suche nach UID-Einheiten

Operation: Search

Beschreibung

Die Suche im UID-Register, erlaubt die Suche nach UID-Einheiten mit einer beliebigen Kombination von Parametern. Das Suchresultat ist auf maximal 30 Datensätze beschränkt.

Wenn es sich bei einem der angegebenen Suchparameter um einen eindeutigen Identifikator handelt (UID, HR-Referenznummer, AHV-Nummer, usw.), werden die restlichen Parameter ignoriert.

Die Suche nach AHV-Nummer (vn) ist für anonyme Benutzer nicht erlaubt und wird mit einer Permission_Denied Exception quittiert.

Parameter

| Parameter | Datentyp | Beschreibung |
|------------------|------------------------------|---|
| searchParameters | uidEntityPublicSearchRequest | Die Suchparameter . |
| searchSettings | searchConfiguration | Die gewünschten Einstellungen für die Suche |

Rückgabewert

uidEntitySearchResponse

Mögliche Fehlermeldungen

- Data_validation_failed
- Request_limit_exceeded
- Permission_denied

Bemerkung: Bei der Suche gibt es kein Not_found. Wenn keine passenden UID-Einheiten gefunden werden, wird ein leeres Resultat zurückgegeben.

4.1.3 Detaildaten abfragen mit der UID

Operation: **GetByUID**

Beschreibung

Diese Operation bietet einen schnellen Zugriff für die Abfrage von Unternehmensdaten zu einer bekannten UID. Das Resultat ist analog zu einer Suche nach UID.

Parameter

| Parameter | Datentyp | Beschreibung |
|-----------|---------------------------|--------------------------------|
| uid | eCH-0097:uidStructureType | UID des gesuchten Unternehmens |

Rückgabewert

eCH-0108:organisationType Array

Mögliche Fehlermeldungen

- Data_validation_failed
- Request_limit_exceeded

4.1.4 Validieren einer UID

Operation: **ValidateUID**

Beschreibung

Diese Operation erlaubt es, die Gültigkeit einer UID zu prüfen, auch wenn die nötigen Berechtigungen zum Anzeigen der dazugehörigen Unternehmensdaten fehlen.

Eine UID ist gültig, wenn sie im UID-Register einem Unternehmen zugeordnet ist, also auch dann wenn dieses Unternehmen bereits erloschen ist.

Parameter

| Parameter | Datentyp | Beschreibung |
|-----------|----------|---------------------|
| uid | String | Die zu prüfende UID |

Rückgabewert

Boolean

Mögliche Fehlermeldungen

- Data_validation_failed
- Request_limit_exceeded

4.1.5 Validieren einer MWST-Nummer

Operation: **ValidateVatNumber**

Beschreibung

Diese Operation erlaubt es, die Gültigkeit einer MWST-Nummer zu prüfen.

Eine MWST-Nummer ist nur dann gültig, wenn sie aktiv ist.

Parameter

| Parameter | Datentyp | Beschreibung |
|-----------|----------|---|
| vatNumber | String | Die zu prüfende MWST-Nummer im Format CHE-123.456.789 oder CHE123456789 |

Rückgabewert

Boolean

Mögliche Fehlermeldungen

- Data_validation_failed
- Request_limit_exceeded

5 Partner-Services

5.1 Operationen

5.1.1 Beispieldaten beziehen

Operation: **GetOrganisationSample**

Beschreibung

Diese Operation bildet keinen Prozess ab. Sie dient einzig dazu, einen fiktiven Datensatz im Format eCH-0108:organisationType zurückzuliefern. Dieser Datensatz enthält **alle** Felder, welche im UID-Register gemapped sind. Die Daten sind nicht konsistent, das heisst die gesetzten Attribute werden in dieser Kombination in echt nicht vorkommen.

Parameter

keine

Rückgabewert

Beispieldatensatz im Format eCH-0108:organisationType

5.1.2 Suche nach UID-Einheiten

Operation: **Search**

Beschreibung

Die Suche im UID-Register, erlaubt die Suche nach UID-Einheiten mit einer eindeutigen ID oder mit einer beliebigen Kombination von Parametern. Das Suchresultat ist auf maximal 200 Datensätze beschränkt.

Wenn es sich bei einem der angegebenen Suchparameter um einen eindeutigen Identifikator handelt (UID, OtherOrganisationId, AHV-Nummer), haben die searchSettings keinen Einfluss.

Parameter

| Parameter | Datentyp | Beschreibung |
|------------------|------------------------|---|
| searchParameters | uidEntitySearchRequest | Die Suchparameter . |
| searchSettings | searchConfiguration | Die gewünschten Einstellungen für die Suche |

Rückgabewert

uidEntitySearchResponse

Mögliche Fehlermeldungen

- Permission_denied
- Schema_validation_failed
- Data_validation_failed

Bemerkung: Bei der Suche gibt es kein Not_found. Wenn keine passenden UID-Einheiten gefunden werden wird eine leere Liste zurückgegeben.

5.1.3 Schnelle Suche nach UID-Einheiten

Operation: QuickSearch

Beschreibung

Die Suche im UID-Register, erlaubt die Suche nach UID-Einheiten mit einer eindeutigen ID oder mit einer beliebigen Kombination von Parametern. Das Suchresultat ist auf maximal 200 Datensätze beschränkt.

Wenn es sich bei einem der angegebenen Suchparameter um einen eindeutigen Identifikator handelt (UID, OtherOrganisationId, AHV-Nummer), haben die searchSettings keinen Einfluss.

Parameter

| Parameter | Datentyp | Beschreibung |
|------------------|------------------------|---|
| searchParameters | uidEntitySearchRequest | Die Suchparameter . |
| searchSettings | searchConfiguration | Die gewünschten Einstellungen für die Suche |

Rückgabewert

uidEntitySearchResponse

Mögliche Fehlermeldungen

- Permission_denied
- Schema_validation_failed
- Data_validation_failed

Bemerkung: Bei der Suche gibt es kein Not_found. Wenn keine passenden UID-Einheiten gefunden werden wird eine leere Liste zurückgegeben.

5.1.4 UID-Einheiten registrieren

Operation: Create

Beschreibung

Ermöglicht es, ein Unternehmen neu im UID-Register einzutragen. Diese neue UID-Einheit erhält den Status „provisorisch“, bis die Neumeldung vom BFS geprüft und bestätigt wird.

Die bei einer Neuanmeldung anzugebenden Felder sind teilweise abhängig vom Typ der meldenden UID-Stelle. Eine Übersicht der Pflichtfelder finden Sie im *Anhang A: Mappingtabelle*

Falls Daten geschickt werden, für welche die meldende UID-Stelle keine Schreibberechtigung hat (z.B. Noga-Code), werden die entsprechenden Felder ohne Fehlermeldung ignoriert und der restliche Datensatz wird entsprechend den Berechtigungen übernommen.

Der eCH-Standard definiert die UID als Pflichtfeld. Da zum Zeitpunkt der Neuanmeldung noch keine UID vorhanden ist, muss als UID CHE-000.000.001 angegeben werden

```
<eCH-0097:uid>  
  <eCH-0097:uidOrganisationIdCategorie>CHE</eCH-0097:uidOrganisationIdCategorie>  
  <eCH-0097:uidOrganisationId>000000001</eCH-0097:uidOrganisationId>  
</eCH-0097:uid>
```

Parameter

| Parameter | Datentyp | Beschreibung |
|---------------|------------------------|----------------------------|
| createRequest | uidEntityCreateRequest | Die Daten des Unternehmens |

Rückgabewert

uidEntityCreateResponse

Es wird die vollständige Einheit, wie sie nach dem Speichern im UID-Register abgelegt ist, zurückgegeben. In diesen Daten ist auch die neu zugewiesene UID enthalten.

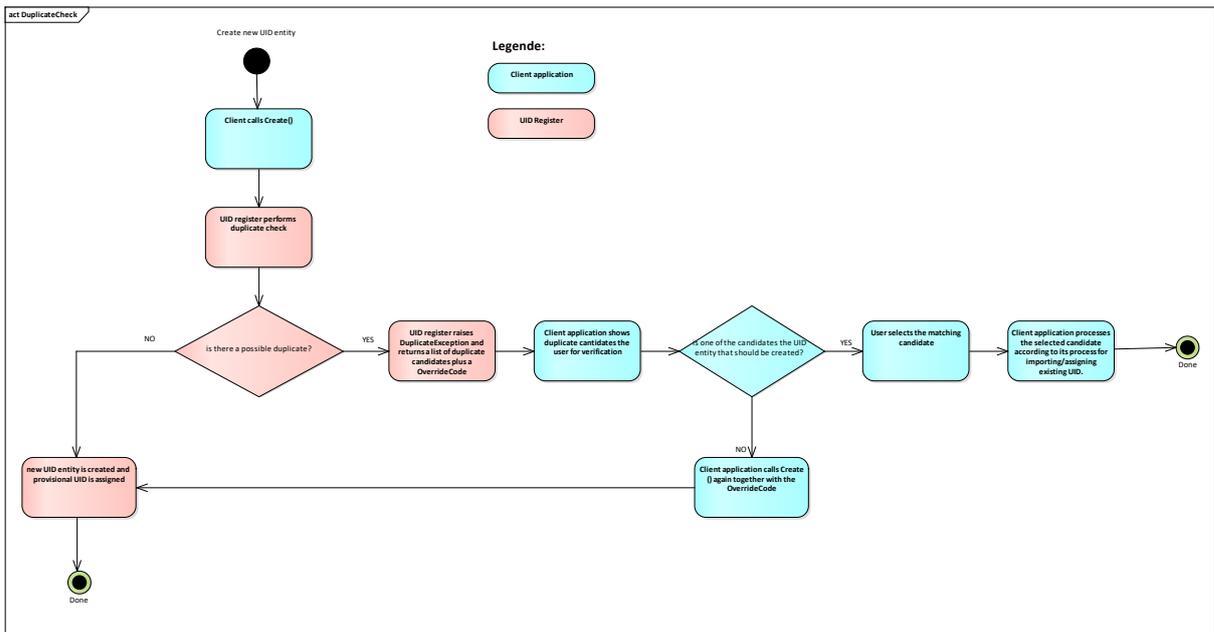
Falls der UID-Status auf „provisorisch“ gesetzt wurde, wird die meldende UID-Stelle per InfoAbo über die Validierung vom BFS informiert.

Mögliche Fehlermeldungen

- Permission_denied
- Schema_validation_failed
- Data_validation_failed
- Possible_duplicate
- Unauthorized

5.1.4.1 Doublettenprüfung

Bei jeder Neumeldung prüft das UID-Register die gesendeten Daten auf mögliche Duplikate. Falls es entsprechende Kandidaten gibt, wirft die Create Operation einen *DuplicateFault*. Dieser enthält eine Liste der möglichen Duplikate sowie einen OverrideCode um die Doublettenprüfung zu übersteuern.



5.1.4.2 OverrideCode

Der OverrideCode ist ein Hashcode, welcher aus den Daten der gemeldeten UID-Einheit berechnet wird. Der Code ist somit nur für genau diese Daten gültig und kann nicht für andere Meldungen verwendet werden. Der Code hat kein Verfalldatum.

5.1.4.3 Melden von Einheiten mit einer reinen Postfach-Adresse

Wenn eine Einheit nur über eine Postfach-Adresse verfügt, sind die Adressdaten wie folgt zu melden. Die Postfachnummer `postOfficeBoxNumber` kann dabei leer sein, dann ist eine postbox-Adresse nur

mit der PLZ anzugeben.

Das Postfach kann nicht in der main-Adresse übertragen werden. Die LEGAL-Adresse mit `swissZipCode` und `town` entspricht den Pflichtfeldern PLZ/Ort und kann nicht weggelassen werden.

Testfirma
Postfach 123
3000 Bern

```
<eCH-0046:address>
  <eCH-0046:addressCategory>LEGAL</eCH-0046:addressCategory>
  <eCH-0010:town>Bern</eCH-0010:town>
  <eCH-0010:swissZipCode>3000</eCH-0010:swissZipCode>
  <eCH-0010:countryIdISO2>CH</eCH-0010:countryIdISO2>
</eCH-0046:address>
<eCH-0046:address>
  <eCH-0046:otherAddressCategory>POBOX</eCH-0046:otherAddressCategory>
  <eCH-0010:postOfficeBoxNumber>123</eCH-0010:postOfficeBoxNumber>
  <eCH-0010:town>Bern</eCH-0010:town>
  <eCH-0010:swissZipCode>3000</eCH-0010:swissZipCode>
  <eCH-0010:countryIdISO2>CH</eCH-0010:countryIdISO2>
</eCH-0046:address>
```

5.1.5 UID-Einheiten mutieren

Operation: Update

Beschreibung

Ermöglicht die Daten einer bestehenden UID-Einheit zu mutieren. Die Identifikation der UID-Einheit erfolgt ausschliesslich anhand der UID `eCH-0098:organisationIdentification/eCH-0097:uid`. Je nachdem welche Felder mutiert werden, erhält die UID-Einheit den UID-Status `eCH-0108:uidregStatusEnterpriseDetail` „in Mutation“, bis die Änderungen vom BFS geprüft und bestätigt sind.

Die bei einer Mutation anzugebenden Felder sind teilweise abhängig vom Typ der meldenden UID-Stelle. Eine Übersicht der Pflichtfelder finden Sie im *Anhang A: Mappingtabelle*. Es ist immer der vollständige Datensatz zu übertragen, nicht nur die zu ändernden Felder.

Falls Daten geschickt werden, für welche die meldende UID-Stelle keine Schreibberechtigung hat, werden die entsprechenden Felder ohne Fehlermeldung ignoriert und der restliche Datensatz wird entsprechend den Berechtigungen übernommen.

Parameter

| Parameter | Datentyp | Beschreibung |
|---------------|------------------------|----------------------------|
| updateRequest | uidEntityUpdateRequest | Die Daten des Unternehmens |

Rückgabewert

uidEntityUpdateResponse

Als Rückgabewert wird die vollständige UID-Einheit, wie sie nach dem Speichern im System abgelegt ist, zurückgegeben. Falls der UID-Status auf „in Mutation“ gesetzt wurde, wird die meldende UID-Stelle per InfoAbo über die Validierung vom BFS informiert.

Mögliche Fehlermeldungen

- Permission_denied
- Schema_validation_failed
- Data_validation_failed
- Possible_duplicate

- No_changes
- Unauthorized
- Not_found

5.1.6 UID-Einheit löschen

Operation: Delete

Beschreibung

Ermöglicht eine UID-Einheit aus dem UID-Register zu löschen. Die Identifikation der UID-Einheit erfolgt ausschliesslich anhand der UID `eCH-0098:organisationIdentification/eCH-0097:uid`.

Wenn als Lösungsgrund 9 (Dublette) angegeben wird, ist im Feld `uidReplacement` zwingend die UID der bestehenden UID-Einheit anzugeben.

Die zu löschende UID-Einheit erhält den Status „in Mutation“, bis die Löschung vom BFS geprüft und bestätigt wird.

Sonderfall 1: Irrtümliche Neumeldung

Falls eine UID-Stelle nach einer Neumeldung (Create) bemerkt, dass es sich bei dieser Meldung um einen Fehler handelt, kann sie dies bis zum Ende des Arbeitstages durch eine Löschung mit Lösungsgrund 8 (Fehler) selbständig korrigieren. Die noch provisorische UID-Einheit erhält in diesem Fall ohne weitere Prüfung den Status "annulliert" und wird damit im UID-Register nicht mehr angezeigt.

Sonderfall 2: Reaktivierung rückgängig machen

Einzelne UID-Stellen benötigen für ihre internen Prozesse eine aktive UID. Aus diesem Grund müssen teilweise gelöschte UID-Einheiten kurzfristig reaktiviert werden.

Solche temporären Reaktivierungen können bis zum Ende des Arbeitstages durch eine Löschung wieder rückgängig gemacht werden. Die UID-Einheit wird ohne weitere Prüfung wieder in den Zustand vor der Reaktivierung zurückversetzt.

Parameter

| Parameter | Datentyp | Beschreibung |
|---------------|------------------------|------------------------|
| deleteRequest | uidEntityDeleteRequest | Daten für die Löschung |

Rückgabewert

uidEntityDeleteResponse

Es wird die vollständige UID-Einheit, wie sie nach dem Speichern im UID-Register abgelegt ist, zurückgegeben. Falls der UID-Status auf „in Mutation“ gesetzt wurde, wird die meldende UID-Stelle per InfoAbo über die Validierung vom BFS informiert.

Mögliche Fehlermeldungen

- Permission_denied
- Schema_validation_failed
- Data_validation_failed
- No_changes
- Not_found

5.1.7 Gelöschte UID-Einheit reaktivieren

Operation: Reactivate

Beschreibung

Ermöglicht eine gelöschte UID-Einheit zu reaktivieren. Die Identifikation der UID-Einheit erfolgt ausschliesslich anhand der UID `eCH-0098:organisationIdentification/eCH-0097:uid`. Die zu reaktivierende UID-Einheit erhält den Status „in Reaktivierung“, bis die Reaktivierung vom BFS geprüft und bestätigt wird.

Parameter

| Parameter | Datentyp | Beschreibung |
|-------------------|----------------------------|------------------------------------|
| reactivateRequest | uidEntityReactivateRequest | UID der zu reaktivierenden Einheit |

Rückgabewert

uidEntityReactivateResponse

Es wird die vollständige UID-Einheit, wie sie nach dem Speichern im UID-Register abgelegt ist, zurückgegeben. Die meldende UID-Stelle wird per InfoAbo über die Validierung vom BFS informiert.

Mögliche Fehlermeldungen

- Permission_denied
- Data_validation_failed
- Possible_duplicate
- Unauthorized
- Not_found

5.1.8 Gelöschte UID-Einheiten mutieren und reaktivieren

Operation: **UpdateAndReactivate**

Beschreibung

Ermöglicht es, eine gelöschte UID-Einheit zu reaktivieren und gleichzeitig die Daten zu mutieren. Die Identifikation der UID-Einheit erfolgt ausschliesslich anhand der UID `eCH-0098:organisationIdentification/eCH-0097:uid`. Die zu reaktivierende UID-Einheit erhält den Status „in Reaktivierung“, bis die Reaktivierung vom BFS geprüft und bestätigt wird.

Die bei einer Mutation anzugebenden Felder sind teilweise abhängig vom Typ der meldenden UID-Stelle. Es ist immer der vollständige Datensatz zu übertragen, nicht nur die zu ändernden Felder.

Falls Daten geschickt werden, für welche die meldende UID-Stelle keine Schreibberechtigung hat, werden die entsprechenden Felder ohne Fehlermeldung ignoriert und der restliche Datensatz wird entsprechend den Berechtigungen übernommen.

Parameter

| Parameter | Datentyp | Beschreibung |
|---------------|------------------------|----------------------------|
| updateRequest | uidEntityUpdateRequest | Die Daten des Unternehmens |

Rückgabewert

uidEntityUpdateResponse

Es wird die vollständige UID-Einheit, wie sie nach dem Speichern im UID-Register abgelegt ist, zurückgegeben. Die meldende UID-Stelle wird per InfoAbo über die Validierung vom BFS informiert.

Mögliche Fehlermeldungen

- Permission_denied
- Schema_validation_failed
- Data_validation_failed

- Possible_duplicate
- No_changes
- Unauthorized
- Not_found

5.1.9 Als meldende UID-Stelle registrieren

Operation: Register

Beschreibung

Ermöglicht einer UID-Stelle, sich bei UID-Einheiten als *UID-Stelle mit umfassenden Pflichten* zu registrieren. Mit der Registrierung wird die UID-Stelle bei den registrierten UID-Einheiten als *Informationsquelle* eingetragen. Mit der Registrierung wird automatisch auch das InfoAbo der registrierten UID-Einheiten abonniert, die UID-Stelle wird somit fortan vom System über alle Mutationen an der registrierten UID-Einheit informiert (siehe Kapitel 5.1.13 InfoAbo abfragen)

Eine nähere Beschreibung des Registrieren/Deregistrieren Prozesses findet sich im Anwenderhandbuch auf www.uid.ch

Passive Registrierung: siehe 5.1.11 InfoAbo einer UID-Einheit abonnieren

Parameter

| Parameter | Datentyp | Beschreibung |
|-----------------|------------------------------|--|
| registerRequest | uidRegisterDeregisterRequest | Liste der UID für welche die meldende Stelle registriert werden soll |

Rückgabewert

registerDeregisterResult

Wenn *uidList* weniger als 100 Einträge enthält, wird die Anfrage direkt verarbeitet. Bei grösseren Listen wird die Anfrage gespeichert und später in einem Hintergrundjob verarbeitet. In diesem Fall enthält *registerDeregisterResult* nur eine *taskId*. Mit dieser kann später das Resultat über die Operation *GetRegisterResult* abgefragt werden.

Mögliche Fehlermeldungen

- Permission_denied
- Data_validation_failed

5.1.10 Als meldende UID-Stelle deregistrieren

Operation: Deregister

Beschreibung

Ermöglicht einer UID-Stelle, sich von registrierten UID-Einheiten abzumelden. Die UID-Stelle wird dabei aus den Informationsquellen entfernt und erhält in der Folge keine Notifikationen zu den entsprechenden UID-Einheiten mehr

Parameter

| Parameter | Datentyp | Beschreibung |
|-------------------|------------------------------|--|
| deregisterRequest | uidRegisterDeregisterRequest | Liste der UID für welche die meldende Stelle deregistriert werden soll |

Rückgabewert

registerDeregisterResult

Wenn *uidList* weniger als 100 Einträge enthält, wird die Anfrage direkt verarbeitet. Bei grösseren Listen wird die Anfrage gespeichert und später in einem Hintergrundjob verarbeitet. In diesem Fall enthält *registerDeregisterResult* nur eine *taskId*. Mit dieser kann später das Resultat über die Operation *GetDeregisterResult* abgefragt werden.

Mögliche Fehlermeldungen

- *Permission_denied*
- *Data_validation_failed*

5.1.11 InfoAbo einer UID-Einheit abonnieren

Operation: **Subscribe**

Beschreibung

Erlaubt es UID-Stellen und ausgesuchten UID-Einheiten, sich für das InfoAbo einer UID-Einheit zu abonnieren. Die Abonnie rung ist nach aussen nicht sichtbar.

Parameter

| Parameter | Datentyp | Beschreibung |
|-------------------------|-------------------------------------|--|
| <i>subscribeRequest</i> | <i>uidRegisterDeregisterRequest</i> | Liste der UID für welche das InfoAbo abonniert werden soll |

Rückgabewert

registerDeregisterResult

Wenn *uidList* weniger als 100 Einträge enthält, wird die Anfrage direkt verarbeitet. Bei grösseren Listen wird die Anfrage gespeichert und später in einem Hintergrundjob verarbeitet. In diesem Fall enthält *registerDeregisterResult* nur eine *taskId*. Mit dieser kann später das Resultat über die Operation *GetSubscribeResult* abgefragt werden.

Mögliche Fehlermeldungen

- *Permission_denied*
- *Data_validation_failed*

5.1.12 InfoAbo einer UID-Einheit abbestellen

Operation: **Unsubscribe**

Beschreibung

Erlaubt es den abonnierten UID-Stellen und UID-Einheiten, sich vom InfoAbo einer UID-Einheit abzumelden. UID-Stellen welche als *UID-Stelle mit umfassende Pflichten* registriert sind, können das InfoAbo nicht gesondert abbestellen.

Parameter

| Parameter | Datentyp | Beschreibung |
|---------------------------|-------------------------------------|---|
| <i>unsubscribeRequest</i> | <i>uidRegisterDeregisterRequest</i> | Liste der UID für welche das InfoAbo abbestellt werden soll |

Rückgabewert

registerDeregisterResult

Wenn *uidList* weniger als 100 Einträge enthält, wird die Anfrage direkt verarbeitet. Bei grösseren Listen wird die Anfrage gespeichert und später in einem Hintergrundjob verarbeitet. In diesem Fall enthält *registerDeregisterResult* nur eine *taskId*. Mit dieser kann später das Resultat über die Operation *GetUnsubscribeResult* abgefragt werden.

Mögliche Fehlermeldungen

- *Permission_denied*
- *Data_validation_failed*

5.1.13 InfoAbo abfragen

Operation: **GetInfoAboMessages**

Beschreibung

Die *GetInfoAboMessages* Operationen, ermöglichen es UID-Stellen und berechtigten UID-Einheiten, ihre InfoAbo-Meldungen aus einem bestimmten Zeitraum abzufragen. Die Meldungen richten sich nach der InfoAbo-Konfiguration der abfragenden Stelle. Das heisst also, es werden nicht alle Mutationen im UID-Register angezeigt, sondern nur die an den registrierten/abonnierten UID-Einheiten. Die Abfrage kann bis zu 60 Tage in die Vergangenheit gemacht werden. Es werden maximal 10'000 Meldungen zurückgegeben. Falls für eine UID-Einheit im abgefragten Zeitraum mehrere Meldungen existieren, wird nur die neuste Meldung zurückgeliefert. Die Unternehmensdaten in einer InfoAbo-Meldung enthalten immer die zum Zeitpunkt der Abfrage gültigen Daten der UID-Einheit. Es können keine historischen Daten abgefragt werden.

BEISPIEL

Eine Einheit wurde in KW13 mutiert, was als *Meldung A* festgehalten wurde. In KW15 erfolgte eine erneute Mutation, welche zu *Meldung B* führte.

In KW16 werden folgende Abfragen gemacht:

1. Abfrage der KW13
-> enthält in diesem Fall die *Meldung A*, die enthalten Unternehmensdaten entsprechen aber dem aktuellen Stand, enthalten also auch bereits die Mutation aus KW15.
2. Abfrage der KW14
-> enthält keine Meldungen
3. Abfrage der KW15
-> enthält die *Meldung B*.
4. Abfrage über dem ganzen Zeitraum KW13-KW16
-> enthält ebenfalls nur die *Meldung B*.

Die Nachrichtentypen *MutationConfirmed* und *MutationRejected* sind auch dann enthalten, wenn deren Versand in der InfoAbo Konfiguration unterdrückt ist. Falls diese Meldungen auch in der Webservice-Abfrage nicht gewünscht sind, können sie über den optionalen Filter *messageTypes* im *getInfoAboRequest* ausgefiltert werden.

Eine nähere Beschreibung des InfoAbo findet sich im Anwenderhandbuch auf www.uid.ch

Parameter

| Parameter | Datentyp | Beschreibung |
|--------------------------|----------------------------------|-------------------------------|
| <i>getInfoAboRequest</i> | <i>getInfoAboMessagesRequest</i> | Parameter der InfoAbo Abfrage |
| | | |

Rückgabewert

infoAboMessage Array

Mögliche Fehlermeldungen

- Permission_denied
- Invalid_date

Die Abfrage erfolgt in der Form `messageDate >= dateFrom AND messageDate < dateTo`.

Beispiel: Lückenlose stündliche Abfrage

dateFrom = 2015-10-20T13:00:00 dateTo = 2015-10-20T14:00:00
 dateFrom = 2015-10-20T14:00:00 dateTo = 2015-10-20T15:00:00
 dateFrom = 2015-10-20T15:00:00 dateTo = 2015-10-20T16:00:00
 usw.

Beispiel: Lückenlose tägliche Abfrage :

dateFrom = 2015-10-20T00:00:00 dateTo = 2015-10-21T00:00:00
 dateFrom = 2015-10-21T00:00:00 dateTo = 2015-10-22T00:00:00
 usw.

Serverzeit ist UTC+01:00

Operation: GetInfoAboMessageByUUID

Beschreibung

Gezielte Abfrage einer InfoAbo-Meldung anhand der eindeutigen messageId.

Parameter

| Parameter | Datentyp | Beschreibung |
|-----------|----------|--------------------------------|
| uuid | GUID | UUID der gewünschten Nachricht |

Rückgabewert

infoAboMessage

Mögliche Fehlermeldungen

- Permission_denied
- Not_found

5.1.14 Abfragen der Resultate von Hintergrundverarbeitungen

**Operation: GetRegisterResult
 GetDeregisterResult
 GetSubscribeResult
 GetUnsubscribeResult**

Beschreibung

Bei den Operationen *Register*, *Deregister*, *Subscribe* und *Unsubscribe* werden nur Anfragen mit weniger als 100 Datensätzen synchron verarbeitet. Grössere Datenmengen werden zwischengespeichert und von einem Hintergrundjob verarbeitet. Die genannten Operationen liefern in dem Fall nur eine *taskId* zurück. Mit dieser taskId und den Get-Result Operationen, können die Resultate später ausgelesen werden.

Parameter

| Parameter | Datentyp | Beschreibung |
|-----------|----------|-------------------------------------|
| taskId | int | Die taskId des gewünschten Auftrags |

Rückgabewert

registerDeregisterResult

Mögliche Fehlermeldungen

- Permission_denied
- Not_found
- Not_ready

5.1.15 Datenabgleich mit Zefix anstossen

Operation: SyncDataFromZefix

Diese Operation kann ausschliesslich vom EHRA genutzt werden!

Beschreibung

Diese Operation ermöglicht es dem EHRA, im UID-Register einen Datenabgleich mit dem Zefix anzustossen. Es kann zu diesem Zweck eine Liste mit 1-n UID geliefert werden.

Die Verarbeitung erfolgt asynchron, d.h. die Daten werden zwischengespeichert und von einem Hintergrundjob verarbeitet.

Für die gemeldeten UID-Einheiten, werden über den Zefix Webservice die aktuellen Daten abgerufen und in das UID-Register übernommen. Via InfoAbo werden anschliessend die registrierten UID-Stellen über die Mutationen informiert.

Parameter

| Parameter | Datentyp | Beschreibung |
|-----------|---------------------------------|--|
| uidList | eCH-0097:uidStructureType Array | Liste der UID, welche mit dem Zefix synchronisiert werden soll |

Rückgabewert

Boolean (nur „true“ möglich, als Bestätigung, dass die Daten erfolgreich entgegen genommen wurden. Fehler werde durch Faults angezeigt)

Mögliche Fehlermeldungen

- Permission_denied
- Data_validation_failed
- Application_error

6 Terminierte Meldungen

Mit den terminierten Meldungen bietet das UID-Register die Möglichkeit, Meldungen zur Ausführung an einem bestimmten Datum vor zu erfassen. Terminierte Meldungen werden vom UID-Register entgegen genommen und am gewünschten Tag im Namen des Auftraggebers ausgeführt.

6.1 Datenvalidierung

Bei Auftragserteilung, werden die eingehenden Daten nur auf ihre Konformität gemäss eCH und die Vollständigkeit (Pflichtfelder) geprüft.

Da die Validierung der UID Geschäftsregeln Abhängigkeiten zum aktuellen Zustand einer Einheit hat, erfolgt diese erst wenn die Meldung verarbeitet wird.

6.2 Rückmeldung

Die Rückmeldung bei terminierten Meldungen erfolgt in Form des InfoAbo. Das InfoAbo umfasst jedoch derzeit nur die in eCH-0116 definierten Meldungen „Mutation bestätigt“ und „Mutation abgelehnt“. Diese werden versandt, nachdem die gemeldeten Daten vom BFS validiert wurden.

Bis auf weiteres erfolgt keine Rückmeldung um die Ausführung am vorgegebenen Datum zu bestätigen oder dabei aufgetretene Fehler zu melden. Im Fehlerfall wird jedoch die UID-Hotline informiert, welche die nötigen Schritte einleitet und die meldende Stelle nötigenfalls kontaktiert.

6.3 Terminierte Meldungen mutieren

Es ist derzeit nicht möglich vorerfasste Meldungen zu mutieren. Um Änderungen an einer terminierten Meldung vorzunehmen, muss die pendente Meldung gelöscht und mit den geänderten Daten neu aufgegeben werden.

6.4 Operationen für terminierte Meldungen

6.4.1 Mutation vorerfassen

Operation: UpdateScheduled

Beschreibung

Siehe 5.1.5 UID-Einheiten mutieren

Parameter

| Parameter | Datentyp | Beschreibung |
|---------------|------------------------|--|
| updateRequest | uidEntityUpdateRequest | Die geänderten Daten des Unternehmens |
| scheduledDate | Date | Datum an welchem die Mutation ausgeführt werden soll |

Rückgabewert

scheduledTaskItem

Mögliche Fehlermeldungen

- Permission_denied
- Schema_validation_failed
- Data_validation_failed

- Invalid_date
- Not_found

6.4.2 Löschung vorerfassen

Operation: DeleteScheduled

Beschreibung

Siehe 5.1.6 UID-Einheit löschen

Parameter

| Parameter | Datentyp | Beschreibung |
|---------------|------------------------|--|
| deleteRequest | uidEntityDeleteRequest | Die Daten des zu löschenden Unternehmens |
| scheduledDate | Date | Datum an welchem die Löschung ausgeführt werden soll |

Rückgabewert

scheduledTaskItem

Mögliche Fehlermeldungen

- Permission_denied
- Schema_validation_failed
- Data_validation_failed
- Invalid_date
- Not_found

6.4.3 Registrierung vorerfassen

Operation: RegisterScheduled

Beschreibung

Siehe 5.1.9 Als meldende UID-Stelle registrieren

Parameter

| Parameter | Datentyp | Beschreibung |
|---------------|---------------------------------|--|
| uidList | eCH-0097:uidStructureType Array | Liste der UID für welche die meldende Stelle deregistriert werden soll |
| scheduledDate | Date | Datum an welchem die Registrierung ausgeführt werden soll |

Rückgabewert

scheduledTaskItem

Mögliche Fehlermeldungen

- Permission_denied
- Data_validation_failed
- Invalid_date

6.4.4 Deregistrierung vorerfassen

Operation: DeregisterScheduled

Beschreibung

Siehe 5.1.10 Als meldende UID-Stelle deregistrieren

Parameter

| Parameter | Datentyp | Beschreibung |
|-------------------|------------------------------|--|
| deregisterRequest | uidRegisterDeregisterRequest | Liste der UID für welche die meldende Stelle deregistriert werden soll |
| scheduledDate | Date | Datum an welchem die Deregistrierung ausgeführt werden soll |

Rückgabewert

scheduledTaskItem

Mögliche Fehlermeldungen

- Permission_denied
- Data_validation_failed
- Invalid_date

6.4.5 Pendente Meldungen abfragen

Operation: **GetAllScheduledTasks**
GetScheduledTasksByType

Beschreibung

Terminierte Meldungen welche noch nicht ausgeführt wurden, können jederzeit abgefragt werden. Es stehen 2 Operationen zur Auswahl, wobei bei *GetScheduledTaskByType* die Daten nach Type eingeschränkt werden können.

Parameter (nur GetScheduledTasksByType)

| Parameter | Datentyp | Beschreibung |
|-----------|--------------------------|-------------------------------|
| type | <i>scheduledTaskType</i> | Typ der gewünschten Meldungen |

Rückgabewert

scheduledTaskItem Array

Mögliche Fehlermeldungen

- Permission_denied
- Data_validation_failed
- Not_found

6.4.6 Pendente Meldungen löschen

Operation: **CancelScheduledTask**

Beschreibung

Solange eine terminierte Meldung noch nicht ausgeführt wurde, kann sie jederzeit wieder gelöscht werden. Die dazu nötige *taskId* wird bei Erfassen der Meldung angegeben, oder kann durch Abfrage der pendenten Meldungen ermittelt werden.

Parameter

| Parameter | Datentyp | Beschreibung |
|-----------|------------|------------------------------|
| taskId | <i>int</i> | Id der zu löschenden Meldung |

Rückgabewert

Boolean

Es gibt nur die Rückmeldung mit *true*, wenn die Löschung erfolgreich war. Fehler werden mit entsprechenden Fehlermeldungen angezeigt.

Mögliche Fehlermeldungen

- Permission_denied
- Not_found

7 Fehlermeldungen

Fehlermeldungen werden dem Client in Form einer generischen FaultException zurückgemeldet. Es existieren 4 Fault-Typen :

- SecurityFault
- BusinessFault
- DuplicateFault
- InfrastructureFault

Jeder Fault verfügt über folgende Parameter

| Parameter | Beschreibung |
|--------------|---|
| Operation | Name der ausgeführten Operation, z.B. <i>Duplicate check, Data validation etc.</i> |
| Error | Fehlercode , z.B. <i>Not_found</i> |
| ErrorDetails | Weitergehende Informationen zum aufgetretenen Fehler, z.B. <i>The account is locked</i> |

Die Fehlermeldungen werden ausschliesslich in Englisch zurückgegeben. Die ErrorDetails richten sich an die Entwickler und sind eher technisch formuliert.

7.1 SecurityFault

SecurityFaults können im Zusammenhang mit der Anmeldung oder den Berechtigungen auftreten. Folgende Fehlercodes sind definiert:

| Fehlercode | Mögliche Ursache |
|-------------------|---|
| Login_failed | Username/Passwort inkorrekt Account ist gesperrt User hat keine Webservice Berechtigung |
| Permission_denied | Der Benutzer ist für die gewählte Operation nicht berechtigt |

7.2 BusinessFault

Unter die Kategorie BusinessFault fallen alle Fehler und Probleme im Zusammenhang mit den Daten. Geprüft werden die Konformität gemäss eCH-0108, Vollständigkeit der Pflichtfelder, Existenz von Duplikaten sowie die Berechtigung zum Erfassen von UID-Einheiten mit dem angegebenen Typ

| Fehlercode | Mögliche Ursache |
|------------------------|---|
| Data_validation_failed | Die Daten entsprechen nicht den Vorgaben des UID-Registers, z.B. <ul style="list-style-type: none">- Fehlende Pflichtfelder- Angaben liegen ausserhalb des Wertbereichs- Unzulässige Statusänderung |
| Invalid_date | Ein Datum liegt ausserhalb des gültigen Bereichs |
| No_changes | Die gemeldeten Daten sind identisch mit denen im UID-Register. Die Meldung wird verworfen. Es wird <u>kein</u> Historyeintrag erstellt und es werden <u>keine</u> Folgeprozesse (Validierung, Infoabo) ausgelöst. |
| Not_found | Die angegebene UID-Einheit wurde nicht gefunden Der angegebene Task wurde nicht gefunden |

| Fehlercode | Mögliche Ursache |
|--------------------------|---|
| Not_ready | Der abgefragte Task wurde noch nicht verarbeitet |
| Possible_duplicate | Der zu erfassende Datensatz könnte bereits existieren. <i>candidates enthält eine Liste mit den möglichen Dubletten</i> |
| Request_limit_exceeded | Bei den Public Services besteht die Möglichkeit, die Anzahl Requests die ein Client pro Minute absetzen kann zu begrenzen. Falls diese Begrenzung eingeschaltet ist und die Limit erreicht wird, wird diese Fehlermeldung ausgegeben. <i>errorDetail</i> enthält dabei nähere Angaben zur gesetzten Limite. |
| Schema_validation_failed | Die Daten sind nicht eCH-konform <i>errorDetail zeigt eine detaillierte Auflistung der Validierungsfehler</i> |
| Unauthorized | Der Benutzer ist nicht autorisiert nach AHV-Nummer suchen. Die meldende Stelle ist nicht autorisiert eine Einheit eines bestimmten Typs zu bearbeiten/löschen Die meldende Stelle ist die zuständige UID-Stelle und ist daher nicht autorisiert zu deregistrieren. |

Permission denied vs Unauthorized

Permission und Authorization sind eigentlich Synonyme. Im UID-Register unterscheiden wir die beiden Begriffe wie folgt:

Eine Permission-wird auf Stufe einer einzelnen Operation vergeben. So hat ein User die Permission für Search, Register etc.

Die Authorization wird für Einschränkungen innerhalb einer Operation vergeben. So kann ein User die Permission haben eine neue Einheit zu kreieren, ist dabei aber nicht autorisiert Einheiten vom Typ "Juristische Person" zu melden. Oder er hat die Permission zu suchen, ist aber nicht autorisiert nach AHV-Nummer zu suchen.

7.3 DuplicateFault

Dieser Fehlertyp tritt auf, wenn beim Melden einer neuen UID-Einheit möglich Duplikate festgestellt werden. Die potentiellen Duplikate werden als *candidates* im Format *duplicateCandidateType* zurückgegeben. Zusätzlich liefert der Fault einen *duplicateOverrideCode*. Mit diesem Code kann die Doublettenprüfung übersteuert und der Eintrag der neuen Einheit forciert werden.

```
<xs:complexType name="duplicateFault">
  <xs:sequence>
    <xs:element name="operation" type="xs:token" />
    <xs:element name="error" type="xs:token" />
    <xs:element name="errorDetail" type="xs:token" />
    <xs:element name="candidate" type="uid-wse:duplicateCandidateType" maxOccurs="un-
bounded" />
    <xs:element name="duplicateOverrideCode" type="xs:token" />
  </xs:sequence>
</xs:complexType>
```

Der duplicateFault enthält neben den 3 allgemeinen Fault-Parametern die beiden folgenden Felder:

| Element | Datentyp | Kardinalität |
|-----------------------|--------------------------------|--------------|
| | Beschreibung | |
| candidate | duplicateCandidateType | 1-n |
| | Liste der möglichen Duplikate. | |
| duplicateOverrideCode | token | 1 |

| |
|---|
| Code zum Übersteuern der Doublettenprüfung. Siehe Ablaufdiagramm im Kapitel 5.1.4.1 Doublettenprüfung |
|---|

```

<xs:complexType name="duplicateCandidateType">
  <xs:sequence>
    <xs:element name="rating" type="xs:int" />
    <xs:element name="uid" type="eCH-0097:uidStructureType" />
    <xs:element name="organisationName" type="eCH-0097:organisationNameType" />
    <xs:element name="organisationAdditionalName" type="eCH-0097:organisationName-
Type" minOccurs="0" />
    <xs:element name="street" type="eCH-0010:streetType" />
    <xs:element name="swissZipCode" type="eCH-0010:swissZipCodeType" />
    <xs:element name="town" type="eCH-0010:townType" />
    <xs:element name="uidregStatusEnterpriseDetail" type="eCH-0108:uidregStatusEn-
terpriseDetailType" />
    <xs:element name="uidregOrganisationType" type="eCH-0108:uidregOrganisationType-
Type" />
  </xs:sequence>
</xs:complexType>

```

7.4 InfrastructureFault

Dieser Fehlertyp tritt auf, wenn eine Anfrage aus technischen Gründen nicht verarbeitet werden kann. Ein Application_Error kann bei jeder Operation auftreten.

| Error | Mögliche Ursache |
|-------------------|--|
| Application_error | Netzwerkfehler Out-of-memory DB-Log voll <i>Er werden keine detaillierten Systeminformationen an den Client weitergegeben</i> |

Anhang A: Mappingtabelle

Zur besseren Übersicht wurde der XPath-Pfad gekürzt. **<root>** entspricht `//eCH-0108:organisationRoot/eCH-0108:organisation`

| Label WebGUI | eCH XPath | ech-Type | |
|-------------------------------------|--|-----------------------------------|------|
| Stammdaten | | | |
| UID | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0097:uid | uidStructureType | 0097 |
| Name * | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0097:organisationName | organisationNameType | 0097 |
| Zusätzlicher Name | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0097:organisationAdditionalName | organisationNameType | 0097 |
| Übersetzung | <root>/eCH-0108:uidregInformation/eCH-0108:uidregTranslation | Token, maxLength 255 | xsd |
| Gründungsdatum | <root>/eCH-0108:organisation/eCH-0098:foundation/eCH-0098:foundationDate | datePartiallyKnownType | 0098 |
| UID Status detailliert | <root>/eCH-0108:uidregInformation/eCH-0108:uidregStatusEnterpriseDetail | uidregStatusEnterpriseDetailsType | 0108 |
| Daten veröffentlichen | <root>/eCH-0108:uidregInformation/eCH-0108:uidregPublicStatus | uidregPublicStatusType | 0108 |
| Löschung im UID-Register | <root>/eCH-0108:organisation/eCH-0098:liquidation/eCH-0098:uidregLiquidationDate | uidregLiquidationReasonType | 0108 |
| Löschungsgrund | <root>/eCH-0108:uidregInformation/eCH-0108:uidregLiquidationReason | uidregLiquidationReasonType | 0108 |
| UID der Nachfolge-Einheit | <root>/eCH-0108:uidregInformation/eCH-0108:uidReplacement | uidStructureType | 0097 |
| Adressen | | | |
| Art der Adresse | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:addressCategory | freeCategoryTextType | 0046 |
| Strasse | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:street | streetType | 0010 |
| Hausnummer/-zusatz | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:houseNumber | houseNumberType | 0010 |
| c/o Adresse | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:addressLine1 | addressLineType | 0010 |
| AddressLine2 (z.B. für "Gebäude A") | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:addressLine2 | addressLineType | 0010 |
| Postfach | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:postOfficeBoxNumber | postOfficeBoxNumberType | 0010 |
| PLZ * | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:swissZipCode | swissZipCodeType | 0010 |
| PLZ Zusatz | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:swissZipCodeAddOn | swissZipCodeAddOnType | 0010 |
| PLZ Ausland | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:foreignZipCode | foreignZipCodeType | 0010 |
| Ort * | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:town | townType | 0010 |
| Kanton * | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:cantonAbbreviation | cantonAbbreviationType | 0007 |
| Gemeinde Nr. | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:municipalityId | municipalityIdType | 0007 |
| EGID | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0098:EGID | EGIDType | 0098 |
| Land * | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0098:address/eCH-0010:countryIdISO2 | countryIdISO2Type | 0010 |
| Kategorisierung | | | |
| Kategorie der UID-Einheit * | <root>/eCH-0108:uidregInformation/eCH-0108:uidregOrganisationType | uidregOragnisationTypeType | 0108 |
| NOGA | <root>/eCH-0108:organisation/eCH-0098:nogaCode | nogaCodeType | 0098 |
| Rechtsform | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0097:legalForm | legalFormType | 0097 |
| Wirtschaftliche Tätigkeit | <root>/eCH-0108:organisation/eCH-0098:uidBrancheText | uidBrancheTextType | 0098 |
| Weitere Identifikationsnr. | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0097:OtherOrganisationId/* | organisationIdCategoryType | 0097 |
| Beteiligte Personen | | | |
| Rolle | <root>/eCH-0108:involvedPerson/eCH-0108:role | roleType | 0108 |
| Vorname | <root>/eCH-0108:involvedPerson/eCH-0108:firstName | baseNameType | 0108 |

| | | | |
|--------------------------------------|---|-----------------------------------|------|
| Nachname | <root>/eCH-0108:involvedPerson/eCH-0108:officialName | baseNameType | 0108 |
| Geburtsdatum | <root>/eCH-0108:involvedPerson/eCH-0108:dateOfBirth | datePartiallyKnownType | 0098 |
| AHV Nummer | <root>/eCH-0108:involvedPerson/eCH-0108:vn | vnType | 0108 |
| Geschlecht | <root>/eCH-0108:involvedPerson/eCH-0108:gender | sexType | 0044 |
| Kommunikation | | | |
| Bevorzugte Sprache * | <root>/eCH-0108:organisation/eCH-0098:languageOfCorrespondance | languageType | 0098 |
| Email | <root>/eCH-0108:organisation/eCH-0098:contact/eCH-0046:email/eCH-0046:emailAddress | emailAddressType | 0046 |
| Webseite | <root>/eCH-0108:organisation/eCH-0098:contact/eCH-0046:internet/eCH-0046:internetAddress | internetAddressType | 0046 |
| Informationssquellen | | | |
| Art der Beziehung | <root>/eCH-0108:uidregInformation/eCH-0108:uidregSource/eCH-0108:relationType | xs:token | |
| UID der Quelle | <root>/eCH-0108:uidregInformation/eCH-0108:uidregSource/eCH-0108:uid | uidStructureType | 0097 |
| Datum der Registrierung | <root>/eCH-0108:uidregInformation/eCH-0108:uidregSource/eCH-0108:registrationDate | xs:date | |
| Status der Registrierung | <root>/eCH-0108:uidregInformation/eCH-0108:uidregSource/eCH-0108:registrationStatus | xs:token | |
| MWST Daten ¹ | | | |
| MwSt. Nummer | <root>/eCH-0108:vatRegisterInformation/eCH-0108:uidVat | uidStructureType | 0097 |
| Begin der MwSt. Pflicht ² | <root>/eCH-0108:vatRegisterInformation/eCH-0108:vatEntryDate | vatEntryDate | 0108 |
| Ende der MwSt. Pflicht | <root>/eCH-0108:vatRegisterInformation/eCH-0108:vatLiquidationDate | vatLiquidationDate | 0108 |
| MwSt. Status | <root>/eCH-0108:vatRegisterInformation/eCH-0108:vatEntryStatus | vatEntryStatusType | 0108 |
| Mwst Gruppe | Tbd groupRelationship | | |
| HR Daten ³ | | | |
| HR Referenznummer ⁴ | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0097:OtherOrganisationId/eCH-0097:organisationIdCategory[text()='CH.HR']/following-sibling::* /text() | organisationIdCategoryType | 0097 |
| HR-Status | <root>/eCH-0108:commercialRegisterInformation/eCH-0108:commercialRegisterEntryStatus | commercialRegisterEntryStatusType | 0108 |
| Übersetzung | <root>/eCH-0108:commercialRegisterInformation/eCH-0108:commercialRegisterNameTranslation | organisationNameType | 0097 |
| Eintrag HR | <root>/eCH-0108:commercialRegisterInformation/eCH-0108:commercialRegisterEntryDate | commercialRegisterEntryDateType | 0108 |
| Löschung HR | <root>/eCH-0108:commercialRegisterInformation/eCH-0108:commercialRegisterLiquidationDate | commercialRegisterLiquidationDate | 0108 |
| Art des Sitzes ⁴ | <root>/eCH-0108:commercialRegisterInformation/eCH-0108:commercialRegisterEnterpriseType | | |
| UID des Hauptsitzes | Tbd groupRelationship | | |
| LEI Daten ⁵ | | | |
| LEI | <root>/eCH-0108:organisation/eCH-0098:organisationIdentification/eCH-0097:OtherOrganisationId/eCH-0097:organisationIdCategory[text()='WW.LEI']/following-sibling::* /text() | organisationIdCategoryType | 0097 |
| InitialRegistrationDate | <root>/eCH-0108:leiRegisterInformation/eCH-0108:initialRegistrationDate | xs:date | |
| NextRenewalDate | <root>/eCH-0108:leiRegisterInformation/eCH-0108:expirationDate | xs:date | |
| Status | <root>/eCH-0108:leiRegisterInformation/eCH-0108:registrationStatus | token | |

* Pflichtfelder

¹ MWST Daten können nur von der ESTV gemeldet werden

² Zusätzliches Pflichtfeld für die ESTV

³ HR Daten können nur von den kantonalen Handelsregistern gemeldet werden

⁴ Zusätzliches Pflichtfeld für Handelsregister

⁵ LEI Daten können nur von LEI-Schweiz gemeldet werden



Anhang B: Testen mit SoapUI

Das Tool SoapUI, welches auch in einer Freeware Version erhältlich ist, hat sich zum Quasi-Standard für das Testen von SOAP Webservices gemauert. Das Tool kann über die Webseite des Herstellers heruntergeladen werden <http://www.soapui.org/>

Als Schnelleinstieg zeigt das folgende Kapitel einige Tipps und Beispiele für den Umgang mit dem SoapUI. Für weiterführende Information bieten die Webseite des Herstellers und das Internet zahllose Tutorials und Anleitungen.

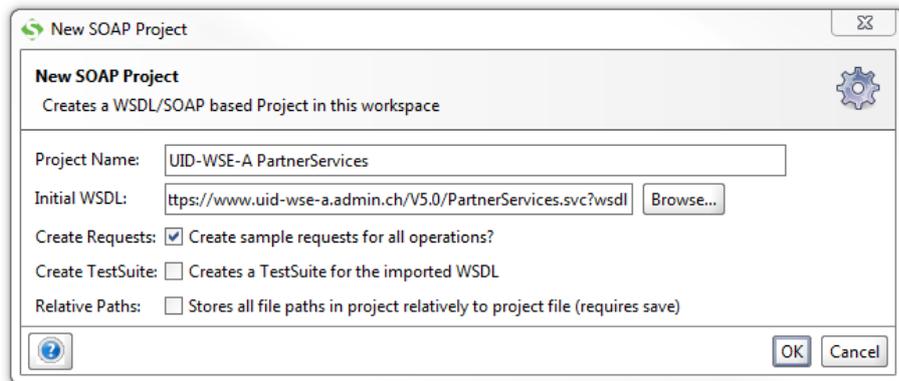
Neues Projekt anlegen

Um ein neues Projekt zu starten benötigen Sie die URL zum WSDL des Webservices (siehe Abschnitt 1.2 WSDL) und im Falle der Partner-Services einen gültigen UID Webservice-Account.

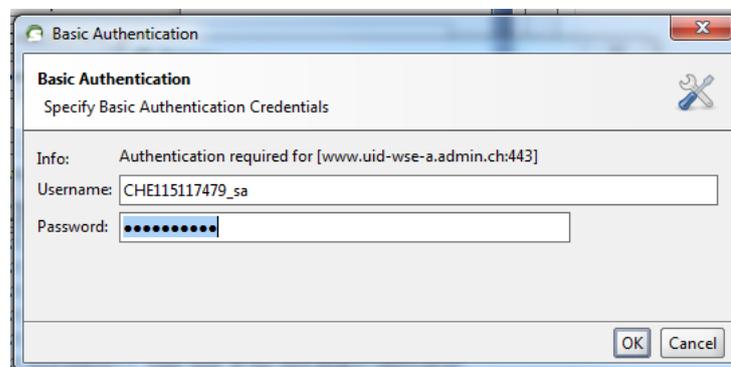
Starten Sie über *File -> New SOAP Project* den Project-Wizard.

Geben Sie dem Projekt einen Namen und kopieren Sie die URL des WSDL ins Feld *Initial WSDL/WADL*.

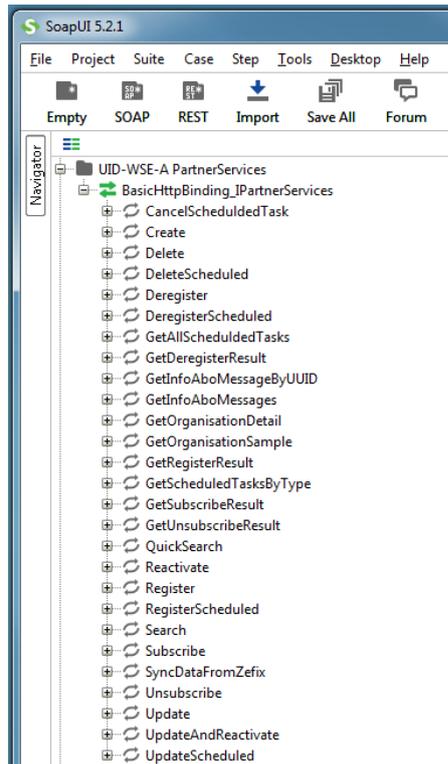
Die darunterliegenden Checkboxes können Sie auf den vorgegebenen Werten belassen,



Nach dem Klicken auf OK verlangt SoapUI nach den Credentials, geben Sie hier Username und Passwort ihres Webservice-Accounts ein.



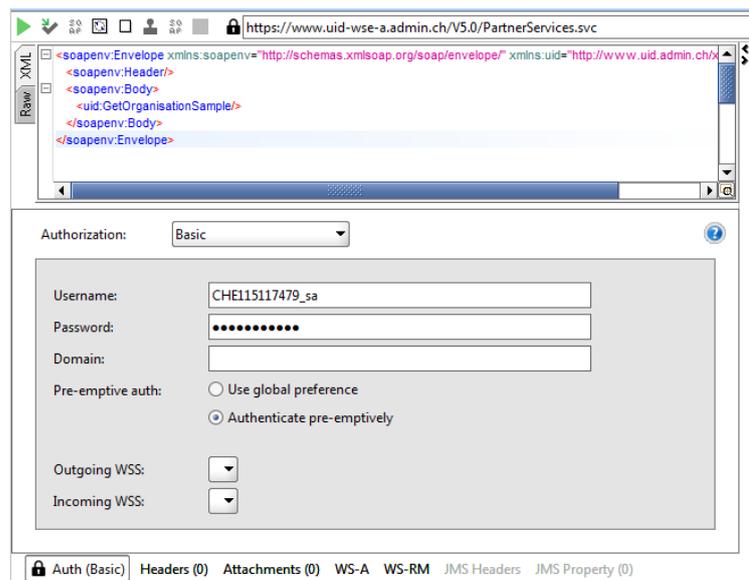
Das neu erstellte Projekt erscheint nun links im Navigator



Credentials für Partner-Services mitgeben

Der Aufruf von Operationen in den Partner-Services verlangt nach einem Benutzernamen und einem Passwort. Die entsprechenden Eingabefelder befinden sich unterhalb des Request-Fensters und sind standardmässig ausgeblendet. Über den Tab *Auth* können sie ein- und ausgeblendet werden.

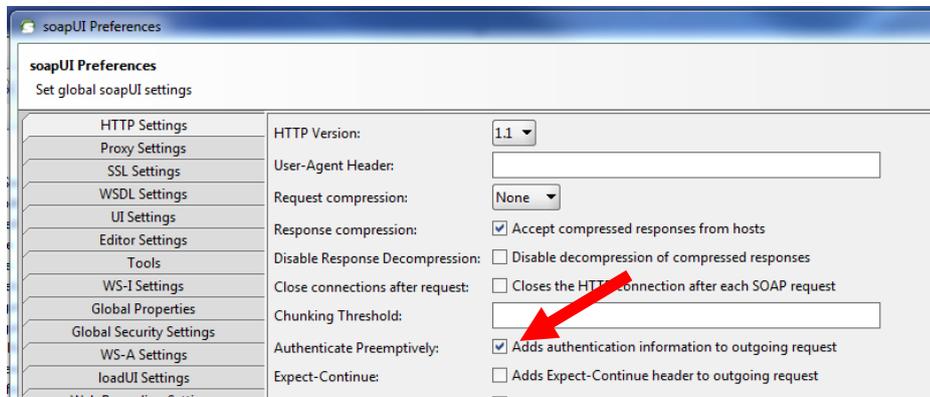
Wählen Sie als Authorization den Typ *Basic* und geben Sie Username und Passwort Ihres Webservice Account ein.



Im Gegensatz zu einer Webseite wird in soapUI nicht mit einem Login-Dialog nachgefragt, falls diese Angaben fehlen. Daher müssen die Credentials bei jedem Aufruf präventiv mitgegeben werden. Dies ist auf 2 Arten möglich:

Die erste Möglichkeit besteht darin, wie im obigen Screenshot "Authenticate pre-emptively" anzuwählen

Alternativ kann die präventive Anmeldung in den Einstellungen als Standard aktiviert werden:



TIP: XML-Validierung

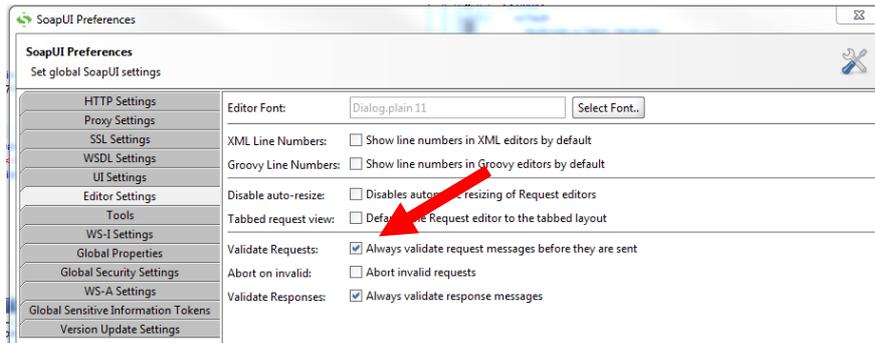
Bei manuellen Tests mit soapUI ist ein Tippfehler schnell geschehen. Das kann zu auf den ersten Blick unerklärlichen Fehlermeldungen führen.

Beispiel: Im folgenden Search-Request wurde <uidOrganisationId> fälschlicherweise mit grossem "ID" geschrieben. Das führt dazu, dass der numerische Teil der UID auf dem Server nicht gemappt werden kann und die UID dadurch unvollständig ist.

```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:uid="http://www.uid.admin.ch/xmlns/uid-wse">
  <soapenv:Header/>
  <soapenv:Body>
    <uid:Search>
      <uid:searchParameters>
        <ns:uid>
          <ns1:uidOrganisationIdCategorie>CHE</ns1:uidOrganisationIdCategorie>
          <ns1:uidOrganisationID>115117479</ns1:uidOrganisationID>
        </ns:uid>
      </uid:searchParameters>
      <uid:config>
        <ns2:searchMode>Auto</ns2:searchMode>
        <ns2:maximumOfRecords>10</ns2:maximumOfRecords>
        <ns2:searchNameAndAddressHistory>false</ns2:searchNameAndAddressHistory>
      </uid:config>
    </uid:Search>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<?xml version='1.0' encoding='UTF-8'>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/">
  <s:Body>
    <s:Fault>
      <faultcode>s:Client</faultcode>
      <faultstring xml:lang="de-CH">Application_error</faultstring>
      <detail>
        <infrastructureFault xmlns="http://www.uid.admin.ch/xmlns/uid-wse" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema-instance">
          <operation xmlns="http://www.uid.admin.ch/xmlns/uid-wse-shared/2">Search</operation>
          <error xmlns="http://www.uid.admin.ch/xmlns/uid-wse-shared/2">Application_error</error>
          <errorDetail xmlns="http://www.uid.admin.ch/xmlns/uid-wse-shared/2">The operation failed [CHE is not a valid UID]</errorDetail>
        </infrastructureFault>
      </detail>
    </s:Fault>
  </s:Body>
</s:Envelope>
```

Um solche Tippfehler zu erkennen empfiehlt es sich, in soapUI die XML Validierung einzuschalten.



soapUI gibt dann eine im Normalfall selbsterklärende Fehlermeldung aus:

