

Schnittstellenbeschreibung

PhM Prescriber



Pharmed Solutions GmbH

Greifenseestrasse 15
8050 Zürich

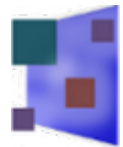
Email: info@pharmedsolutions.ch



In Zusammenarbeit mit

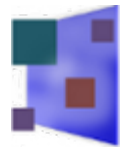
ywesee GmbH
Winterthurerstrasse 52
8006 Zürich

Email: zdavatz@ywesee.com



Inhalt

1	Einleitung	3
1.1	Hintergrund	3
1.2	Datenschutzkonzept	3
1.3	Funktionsweise der Schnittstelle	4
1.4	Technische Details	4
2	Technische Details „Rezept abholen“	5
2.1	Zusammenfassung	5
2.2	Details	5
2.2.1	Request	5
2.2.2	Response	6
2.2.3	Soap Faults	9
2.3	Test-Account	10
3	Technische Details „Rezept übermitteln“	11
3.1	Zusammenfassung	11
3.2	Details	11
3.2.1	Request	11
3.2.2	Response	15
3.2.3	Soap Faults	16
3.3	Test-Account	17
4	Beispielcode	18



1 Einleitung

1.1 Hintergrund

Medikationsfehler stellen auch heute noch ein erhebliches Risiko für Patienten dar und verursachen unnötiges Leid. Ziel der PhM Prescriber Schnittstelle ist deshalb, die Kommunikation zwischen verschreibendem Arzt und abgebender Apotheke zu verbessern, um ebensolche zu verhindern.

Einen ersten Schritt dazu haben wir bereits vorgenommen. Unsere Verschreibungssoftware PhM Prescriber unterstützt verschreibungspflichtige Ärzte bei der Rezeptausstellung und ermöglicht eine einfache und fehlerfreie Übermittlung der Rezepte an die Apotheken. Durch unseren innovativen Ansatz, *keinerlei Patientendaten im Netz zu speichern*, und der einfachen Handhabung der Software haben wir in kurzer Zeit bereits zahlreiche Ärzte überzeugen können, mit unserer Software zu arbeiten. Der Arzt druckt dabei das Rezept aus und gibt es unterschrieben dem Patienten mit. Die Apotheke kann über den aufgedruckten Strichcode die Medikamentendaten aus dem Netz abrufen.

Das vorliegende Dokument soll nun im Weiteren die Konzepte und technischen Grundlagen der Schnittstelle näher erläutern, um weitere Anwendungen einbinden zu können.

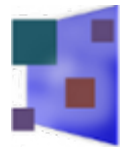
1.2 Datenschutzkonzept

Der Datenschutz nimmt mit zunehmender Digitalisierung ständig an Bedeutung zu, da es möglich wird, verschiedenste personenbezogene Daten zu speichern. Im Gesundheitsbereich sind diese Daten besonders sensibel, denn sie betreffen die Privatsphäre der betroffenen Personen in ihrem intimsten Bereich. Der Umgang mit personenbezogenen Daten ist im Datenschutzgesetz des Bundes gesetzlich geregelt, damit deren Integrität gesichert bleibt.

Unsere Lösung für die Datenschutzproblematik ist denkbar einfach: Wir speichern sensible Patientendaten gar nicht erst ab. Wo keine Daten sind, können diese auch nicht gestohlen oder missbraucht werden. Auf unseren Servern liegen deshalb ausschliesslich die Medikamentendaten (verschriebene Produkte, Verschreibungsdatum u.a.), die keinem Patienten zugeordnet werden können.

Einzig der Arzt und die Apotheke können die Patienten- und Medikamentendaten kombinieren.

Natürlich haben wir trotz dieses Konzeptes alle möglichen technischen und organisatorischen Vorkehrungen getroffen, um die Daten bestmöglich zu schützen. So kann zum Beispiel nur verschlüsselt mit unseren Servern kommuniziert werden. Wie aber leider das Beispiel des „HeartBleed-Bug“ gezeigt hat, ist es nie möglich, Daten zu 100% zu schützen, da jede Infrastruktur von Drittsoftware abhängig ist. Gerade deshalb erachten wir unser Konzept als erfolgsversprechend.



1.3 Funktionsweise der Schnittstelle

Die Schnittstelle ist als Webservice implementiert. Die Funktionsweise gestaltet sich sehr einfach:

1. Der Sender (der verschreibende Arzt oder die Apotheke) schickt eine Anfrage.
 - a. Beispiel Arzt: „Ich würde gerne ein Rezept übermitteln.“
 - b. Beispiel Apotheke: „Ich würde gerne das Rezept mit der Nummer 123 abholen.“
2. Der Webservice bearbeitet die Anfrage und sendet, falls der Sender authentifiziert wurde, eine Antwort zurück.
 - a. Beispiel Arzt: „Die Medikamentendaten wurden erfolgreich empfangen. Das anonyme Rezept hat die Nummer 124.“
 - b. Beispiel Apotheke: „Die Rezeptdaten sind im Anhang. <Anhang mit Medikamenten-Daten>“

Die PhM Prescriber Schnittstelle besitzt im Grunde genau diese zwei Anfragemöglichkeiten, es kann also entweder ein anonymes Rezept gesendet oder ein anonymes Rezept empfangen werden.

1.4 Technische Details

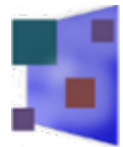
Die Schnittstelle basiert auf dem Webstandard SOAP. Die Beschreibung des Dienstes ist in der entsprechenden WSDL (Web Service Description Language) Datei enthalten und ermöglicht eine einfache Integration des Webservice mit verschiedensten Anwendungen. Die Schnittstelle ist im Übrigen auch WS-I compliant.

Sie können bei uns ein Konto beantragen, um die Schnittstelle zu testen. Des Weiteren können wir Sie mit Beispielcode für Java, C# und VB.NET unterstützen.

Die Schnittstelle bietet zwei Methoden an, die in den folgenden Kapiteln erläutert werden:

1. Ein Rezept abholen (Kapitel 2)
2. Ein Rezept übermitteln (Kapitel 3)

Die Schnittstelle lehnt sich im Übrigen an den **MedRx-Standard** an. Es gibt zusätzlich einige Anpassungen, die in unseren Augen die Schnittstelle einfacher gestalten.



2 Technische Details „Rezept abholen“

2.1 Zusammenfassung

Attribut	Wert
WSDL-Adresse	https://www.pharmedsolutions.ch/RezeptService/wsd
Methode	getRezept()
Input	GLN der Apotheke, Passwort, Secret, Rezeptnummer
Output	Rezept in XML-Format (tns:Rezept)

2.2 Details

Die Methode um ein anonymes Rezept abzuholen ist denkbar einfach: Die Apotheke sendet ihre EAN/GLN-Nummer mit dem dazugehörigen Passwort und die entsprechende Rezeptnummer, die im Regelfall auf dem Rezept aufgedruckt ist (Strichcode). Aus dem Verwaltungssystem der Apotheke wird das Secret eingespielen.

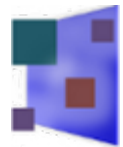
Als Antwort wird ein xml-codiertes Rezept retourniert.

2.2.1 Request

Element	Datentyp	Obligatorisch	Kommentar
GLN_EAN	xsd:integer	ja	EAN der anfragenden Apotheke
Password	xsd:string	ja	Zugehöriges Passwort
Secret	xsd:string	ja	Geheimes Passwort der 2-way authentication
RezeptID	xsd:string	ja	Nummer des Rezeptes (mit oder ohne „PhM“)

Xml-Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns1="https://www.pharmedsolutions.ch/RezeptService">
  <SOAP-ENV:Body>
    <ns1:getRezeptParameters>
      <GLN_EAN>7689827837382</GLN_EAN>
      <Password>skeidk?223._3ap023?</Password>
      <Secret>A1B2C3</Secret>
      <RezeptID>PhM 20127802620140606172440331</RezeptID>
    </ns1:getRezeptParameters>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



2.2.2 Response

Element	Datentyp	Obligatorisch	Kommentar
<i>Rezept</i>	<i>tns:Rezept</i>	Ja	Rezept mit allen benötigten Daten

tns:Rezept

Element	Datentyp	Obligatorisch	Kommentar
rezeptid	xsd:integer	Ja	Nummer des Rezeptes
<i>prescriptor</i>	<i>tns:Prescriptor</i>	ja	Angaben über den verschreibenden Arzt
<i>prescription</i>	<i>tns:Prescription</i>	ja	Rezeptdaten

tns:Prescriptor

Element	Datentyp	Obligatorisch	Kommentar
zsr_id	xsd:string	nein	ZSR des verschreibenden Arztes
ean_id	xsd:integer	nein	EAN des verschreibenden Arztes
last_name	xsd:string	nein	Familiennamen
first_name	xsd:string	nein	Vorname
title	xsd:string	nein	Titel
phone	xsd:string	nein	Telefonnummer
institution	xsd:string	nein	Institution
<i>address</i>	<i>tns:Address</i>	nein	Adresse des verschreibenden Arztes

tns:Address

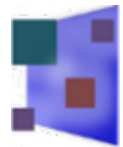
Element	Datentyp	Obligatorisch	Kommentar
street	xsd:string	nein	Strasse
pobox	xsd:string	nein	Hausnummer
zip	xsd:integer	nein	PLZ4 der Schweiz. Post
city	xsd:string	nein	Ortschaft

tns:Prescription

Element	Datentyp	Obligatorisch	Kommentar
date	xsd:date	ja	Verschreibungsdatum
accident_id	xsd:string	nein	Unfallnummer, falls Unfall
accident_date	xsd:date	nein	Unfalldatum, falls Unfall
<i>products</i>	<i>tns:ArrayOfProduct</i>	ja	Liste mit Verschreibungen

tns:ArrayOfProduct

Element	Datentyp	Obligatorisch	Kommentar
<i>item</i>	<i>tns:Product</i>	ja	Enthält die jeweiligen Verschreibungsinformationen



tns:Product

Element	Datentyp	Obligatorisch	Kommentar
line_id	xsd: integer	ja	Zeilennummer für Verschreibung
pharmacode	xsd: integer	ja	Pharmacode des Produktes
ean_id	xsd: integer	ja	EAN des Produktes
product_name	xsd: string	nein	Name des Produktes
prescriptor_qty	xsd: integer	nein	Anzahl Packungen (OP)
prescriptor_repetition_end	xsd: date	nein	Ende der Repetition, falls Dauerrezept
remark	xsd: string	nein	Bemerkung des Arztes (Einnahmeangabe, berücksichtigte Interaktionen, Produktname falls kein Pharmacode etc.)
substitution	xsd: boolean	nein	Substitution nicht erlaubt (sonst null)
<i>posologyRegular</i>	<i>tns:PosologyRegular</i>	nein	Einnahmeangaben

tns: PosologyRegular

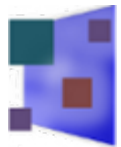
Element	Datentyp	Obligatorisch	Kommentar
qty_morning	xsd: double	nein	Quantität morgens
qty_midday	xsd: double	nein	Quantität mittags
qty_afternoon	xsd: double	nein	Quantität abends
qty_evening	xsd: double	nein	Quantität vor dem Schlafengehen

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ns1="https://www.pharmedsolutions.ch/RezeptService">
<SOAP-ENV:Body>

<ns1:Rezept>
  <rezeptid>20127802620140606172440331</rezeptid>

  <prescriptor>
    <zsr_id>D 04738.01</zsr_id>
    <ean_id>2333333333377</ean_id>
    <last_name>Strub</last_name>
    <first_name>Herbert</first_name>
    <title>Dr. med.</title>
    <phone>061 961 90 80</phone>
    <institution></institution>

    <address>
      <street>Bachgasse</street>
      <pobox>12</pobox>
      <street_additional xsi:nil="true"/>
      <zip>4436</zip>
    </address>
  </prescriptor>
</ns1:Rezept>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



```
<city>Oberdorf BL</city>
  <country xsi:nil="true"/>
</address>

</prescriptor>

<prescription>
  <date>2014-04-17</date>
  <accident_id>asdfasdf</accident_id>
  <accident_date>2014-04-1</accident_date>

  <products>
    <item>
      <line_id>0</line_id>
      <pharmacode>2136399</pharmacode>
      <ean_id>7680542400907</ean_id>
      <prescriptor_qty>1</prescriptor_qty>
      <prescriptor_repetition_end xsi:nil="true"/>
      <remark>1-1-1-1</remark>
      <substitution xsi:nil="true"/>

      <posology>
        <qty_morning>1</qty_morning>
        <qty_midday>1</qty_midday>
        <qty_afternoon>1</qty_afternoon>
        <qty_evening>1</qty_evening>
      </posology>
    </item>

    <item>
      <line_id>1</line_id>
      <pharmacode>5212273</pharmacode>
      <ean_id>7680548240347</ean_id>
      <prescriptor_qty>1</prescriptor_qty>
      <prescriptor_repetition_end xsi:nil="true"/>
      <remark>asdf 1-1-1-1</remark>
      <substitution xsi:nil="true"/>
      <posology xsi:nil="true"/>
    </item>

    <item>
      <line_id>2</line_id>
      <pharmacode>2509110</pharmacode>
      <ean_id>7680553260231</ean_id>
      <prescriptor_qty>1</prescriptor_qty>
      <prescriptor_repetition_end xsi:nil="true"/>
      <remark>asdf</remark>
      <substitution xsi:nil="true"/>
      <posology xsi:nil="true"/>
    </item>
  </products>

</prescription>

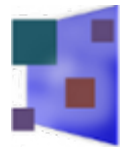
</nsl:Rezept>

</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```




2.2.3 Soap Faults

Fehlernummer	Fehlernachricht	Kommentar
900	Permission denied for RezeptID {RezeptID}.	Für den Testaccount sind nur gewisse RezeptIDs zulässig.
901	Unknown RezeptID	Zur RezeptID existiert kein Rezept. In diesem Fall ist Rücksprache mit dem Arzt zu nehmen (mögliche Fälschung).
902	Wrong credentials supplied or account inactivated.	Die Kontodaten sind nicht korrekt oder das Konto wurde gesperrt (Limit Rezeptabrufe oder Einlogversuche überschritten).
903	{Apotheke, in der das Rezept eingelöst wurde}	Falls das Rezept bereits in einer anderen Apotheke eingelöst wurde, wird der Name ebendieser angezeigt. Dies dient der Vorbeugung von Mehrfacheinlösungen.
904	Wrong RezeptID format.	Die Rezeptnummer hat ein falsches Format.



2.3 Test-Account

Um die Funktionalitäten der Schnittstelle zu testen, haben wir einen Test-Account mit limitiertem Zugang eingerichtet:

- GLN_EAN: 7689827837382
- Password: skeidk?223._3apO23?
- Secret: A1B2C3

Die Schnittstelle kann über einen Soap-Client angesteuert werden. Online-Beispiele wären:

- <http://wsdlbrowser.com/>
 - Eingabe WSDL: <https://www.pharmedolutions.ch/RezeptService/wsdl>
 - „getRezept()“ auswählen
 - Beispiel-XML von Kapitel 2.2.1 einfügen
- <http://www.soapclient.com/soaptest.html>
 - Eingabe WSDL: <https://www.pharmedolutions.ch/RezeptService/wsdl>
 - Parameter eingeben

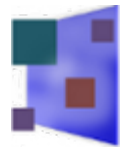
Ein Beispiel für ein grafisches Interface stellt unser Rezeptservice für Apotheken dar:

<https://www.pharmedolutions.ch/apotheken>

Einige interessante Rezeptnummern:

- PhM 20127802620140606172440331
- PhM 20127802620140606175549176
- PhM 20127802620140606175334011

Gerne stellen wir auf Anfrage auch Code für verschiedene Sprachen zur Verfügung, die das Ansteuern der Schnittstelle zeigen (Java, C#, VB.NET).



3 Technische Details „Rezept übermitteln“

3.1 Zusammenfassung

Attribut	Wert
WSDL-Adresse	https://www.pharmedsolutions.ch/RezeptServer/wsd
Methode	postPrescription
Input	Rezept in XML-Format (tns:Prescription)
Output	Rezeptnummer zum Aufdrucken als Strichcode

3.2 Details

Die Methoden um Rezepte zu übermitteln, haben wir so einfach wie möglich gestaltet. Neben den Rezeptdaten wird nur die ZSR-Nummer des verschreibenden Arztes übermittelt. Als Response wird die Rezeptnummer erhalten, die danach auf das Rezept aufgedruckt werden kann.

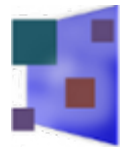
Damit eine Integration mit der MedRx-Schnittstelle von der OFAC AG möglich ist (60% aller Apotheken sind daran angeschlossen), müssen gewisse Patientendaten erfasst werden. **Diese Daten werden direkt verschlüsselt an die OFAC weitergeleitet und zu keinem Zeitpunkt bei uns gespeichert. Die OFAC AG ist berechtigt, Patientendaten zu speichern und dafür vom Bund zertifiziert.**

Wichtig erscheint die Bemerkung, dass die Objekte des Kapitels 2 *nicht identisch* mit jenen dieses Kapitels sind, obwohl die Namen zum Teil ähnlich lauten.

3.2.1 Request

tns:Prescription

Element	Datentyp	Obligatorisch	Kommentar
software	xsd:integer	ja	Von uns zugewiesene Softwarenummer
password	xsd:string	ja	Zur Softwarenummer zugehöriges Passwort
zsr_id	xsd:string	ja	ZSR des verschreibenden Arztes (Format: „X123456“ oder „X 1234.56“)
date	xsd:date	ja	Verschreibungsdatum des Rezepts
accident_id	xsd:string	nein	Unfallnummer, falls Unfall
accident_date	xsd:date	nein	Unfalldatum, falls Unfall
<i>products</i>	<i>tns:ArrayOfProduct</i>	ja	Liste mit verschriebenen Produkten
<i>patient</i>	<i>tns:Patient</i>	ja	Patientendaten, die für die MedRx-Integration von OFAC nötig sind.



tns:ArrayOfProduct

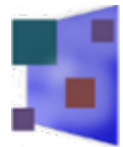
Element	Datentyp	Obligatorisch	Kommentar
<i>item</i>	<i>tns:Product</i>	ja	Jeweilige Verschreibungs- informationen pro Produkt

tns:Product

Element	Datentyp	Obligatorisch	Kommentar
pharmacode	xsd: integer	ja/nein	Pharmacode des Produktes: obligatorisch, falls kein EAN
ean_id	xsd: integer	nein/ja	EAN des Produktes: obligatorisch, falls kein Pharmacode
prescriptor_qty	xsd: integer	nein	Anzahl Packungen (OP), Default ist 0
prescriptor_repetition_end	xsd: date	nein	Ende der Repetition, falls Repetition
remark	xsd: string	nein	Bemerkung des Arztes (Einnahmeangabe, berücksichtigte Interaktionen, Produktname, falls kein Pharmacode etc.)
posology	xsd: string	nein	Einnahmeangaben im Format x-x-x-x (z.B. 1-0.5-2-0)

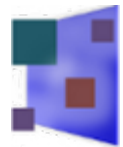
tns:Patient

Element	Datentyp	Obligatorisch	Kommentar
last_name	xsd: string	ja	Familienname des Patienten
first_name	xsd: string	ja	Vorname des Patienten
birth_date	xsd: date	ja	Geburtsdatum des Patienten
covercard	xsd: string	nein	Versichertenkarte- Nummer des Patienten (optional)



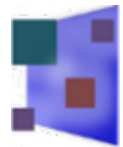
Wichtige Hinweise zum Request:

- Die Posology wird nur sauber geparkt und an die Apotheke übermittelt, falls sie im Format w-x-y-z übermittelt wird, wobei {w,x,y,z} xsd:double sind.
 - Beispiel: 0.5-1-0-1
- Die Posology kann auch im remark-Element mitgegeben werden, da sie automatisch geparkt wird.
- Damit eine Substitution verboten wird, muss der Substring „ **sic** „, „ **sic.** „ oder „ **sic!** „ (die Gross-/Kleinschreibung wird ignoriert) im remark-Element vorhanden sein.
- Ist weder ein Pharmacode noch EAN bekannt, sollte für Pharmacode/EAN „111“ eingesetzt werden und der Produktname im remark-Element angegeben werden.
- Die Authentifizierung wird mit interessierten Softwarehäusern bilateral geregelt.
- Alle Patientendaten werden verschlüsselt an den OFAC-pool geleitet und zu keinem Zeitpunkt auf unserer Infrastruktur gespeichert.



xml-Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="https://www.pharmedsolutions.ch/RezeptServer">
  <SOAP-ENV:Body>
    <ns1:Prescription>
      <software>1</software>
      <password>jdUikd88_kdl?=kdM</password>
      <zsr_id>B 2780.26</zsr_id>
      <date>2014-06-11</date>
      <accident_id>1093.2390 AB</accident_id>
      <accident_date>2014-06-01</accident_date>
      <products>
        <item>
          <pharmacode>123423</pharmacode>
          <ean_id>7601028923892</ean_id>
          <prescriptor_qty>1</prescriptor_qty>
          <prescriptor_repetition_end>2015-05-01</prescriptor_repetition_end>
          <remark>Bei Bedarf. Sic!</remark>
          <posology>1-3-2-1</posology>
        </item>
        <item>
          <ean_id>760102890001</ean_id>
          <prescriptor_qty>5</prescriptor_qty>
          <remark>1-4-3-2, Nach dem Essen</remark>
        </item>
      </products>
      <patient>
        <last_name>Aebi</last_name>
        <first_name>Markus</first_name>
        <birth_date>1955-05-15</birth_date>
      </patient>
    </ns1:Prescription>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```



3.2.2 Response

tns:PrescriptionResponse

Element	Datentyp	Obligatorisch	Kommentar
PrescriptionID	xsd:string	ja	Rezeptnummer zum Aufdrucken

Xml-Beispiel:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns1="https://www.pharmedsolutions.ch/" >
  <SOAP-ENV:Body>
    <ns1:PrescriptionResponse>
      <PrescriptionID>PhM 20127802620140611112305218</PrescriptionID>
    </ns1:PrescriptionResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3.2.2.1 Aufzudruckender Barcode

Der erhaltene PrescriptionID sollte nach Möglichkeit mit dem Präfix „PhM „ als *Code 128* (<http://de.wikipedia.org/wiki/Code128>) auf das Rezept aufgedruckt werden.

Der *Zeichensatz B* bietet sich an, da nur Alphanumerische Zeichen in der PrescriptionID enthalten sind.

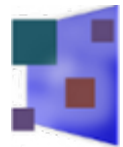
Der aufgedruckte Strichcode sollte nicht breiter als 6.5cm sein, um die Lesbarkeit für alle Scanner zu garantieren, andererseits auch nicht zu klein, damit die Balken sauber gedruckt werden können. Wir empfehlen deshalb eine Breite zwischen 6.2 und 6.4cm.

Für die verschiedenen Entwicklungssprachen sind freie Bibliotheken verfügbar, die den Strichcode automatisch generieren.

Beispiel generiert mit „Barcode Image Generation Library“ für .NET (<http://www.codeproject.com/Articles/20823/Barcode-Image-Generation-Library>)

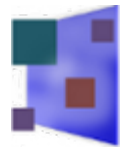


PhM 20127802620140611112305218



3.2.3 Soap Faults

Fehlernummer	Fehlernachricht	Kommentar
900	Invalid ZSR-format supplied (required: Z 1234.56).	Die ZSR hat das falsche Format.
901	Invalid prescription date format.	Das Verschreibungsdatum hat das falsche Format.
902	Invalid accident_date format.	Das Unfalldatum hat das falsche Format.
903	Patient last_name missing.	Der Familienname des Patienten fehlt.
904	Patient first_name missing.	Der Vorname des Patienten fehlt.
905	Invalid patient birth_date format.	Das Geburtsdatum des Patienten hat das falsche Format.
906	Invalid Covercard format supplied.	Die Covercardnummer hat ist nicht numerisch oder hat nicht 20 Ziffern.
907	No products supplied.	Es wurden keine Produkte übermittelt.
908	No pharmacode nor ean_id supplied.	Der Pharmacode oder der EAN-Code fehlt
909	Invalid prescriber_repetition_end format.	Das Repetitionsdatum hat das falsche Format.
910	Invalid posology format.	Die Posologie hat das falsche Format.
911	Physicians EAN not registered at BAG.	Die EAN des Arztes ist nicht im Medizinalregister des Bundes vorhanden.
912	Invalid credentials supplied.	Die Einlogdaten sind nicht korrekt.
913	Daily post limit exceeded.	Das Tageslimit für Rezept-Uploads wurde überschritten (gilt pro Arzt).
999	Wrong ZSR for test account supplied.	Für den Test-Account ist nur die angegebene ZSR erlaubt.



3.3 Test-Account

Die Schnittstelle kann mit den folgenden Benutzerdaten angesteuert werden:

- software: 1
- password: jdUikd88_kd1?=kdM
- zsr_id: B 2780.26

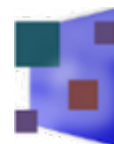
Die Schnittstelle kann über einen Soap-Client angesteuert werden. Online-Beispiele wären:

- <http://wsdlbrowser.com/>
 - Eingabe WSDL: <https://www.pharmedsolutions.ch/RezeptServer/wsdl>
 - „postPrescription()“ auswählen
 - Beispiel-XML von Kapitel 3.2.1 einfügen
- <http://www.soapclient.com/soaptest.html>
 - Eingabe WSDL: <https://www.pharmedsolutions.ch/RezeptServer/wsdl>
 - Parameter eingeben
 - Hinweis: Hier müssen leider auch die optionalen Parameter angegeben werden, damit der Service funktioniert.

Über den Test-Account können Rezepte im korrekten Format hochgeladen werden, entsprechend den Spezifikationen von Kapitel 3.2.1.

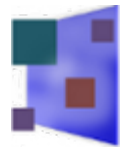
Der Test-Account erlaubt das Hochladen von maximal 90 Rezepten pro Tag. Bei einer Überschreitung des Limits nehmen Sie bitte Kontakt mit uns auf, damit wir den Account wieder freigeben können.

Zusätzlich funktioniert der Test-Account nur für die angegebene ZSR.



4 Beispielcode

Gerne senden wir Ihnen auf Anfrage Codebeispiele für Java, C# und VB.NET.
Angehängt ein Beispiel für C#, nachdem der Webservice importiert wurde.



```
try
{
    //Kreiere neues RezeptService Objekt
    RezeptPortTypeClient RezeptService = new RezeptPortTypeClient();

    //Kreiere neues Request-Objekt
    getRezeptRequest RezeptRequest = new getRezeptRequest();

    //Kreiere neues Parameter Objekt für den RezeptRequest
    getRezeptParameters RezeptParameters = new getRezeptParameters();

    //Weise den 3 Parametern ihre Werte zu
    RezeptParameters.GLN_EAN = tboGLN.Text;
    RezeptParameters.Password = tboPasswort.Text;
    RezeptParameters.RezeptID = tboRezeptID.Text;

    //Weise dem RezeptRequest seine Parameter zu
    RezeptRequest.getRezeptParameters = RezeptParameters;

    //Kreiere neues Rezept-Objekt, das nach dem Konsumieren des Service
    alle Daten enthält
    Rezept Rezept = new Rezept();

    //Konsumiere den Service
    RezeptService.Open();
    Rezept = RezeptService.getRezept(RezeptParameters);

    //Der RezeptService gibt nun ein Rezept-Objekt zurück.
    //Das Rezept-Objekt enthält als Eigenschaften ("Properties) alle
    Knoten des im WSDL definierten Rezept-Objekt (complex type "Rezept")
    //Die Rezeptdaten können über die Eigenschaften des Rezeptobjekts
    bequem ausgelesen werden

    //Lese das Rezept-Objekt aus
    MessageBox.Show("RezeptID: " + Rezept.rezeptid);
    MessageBox.Show("Arzt: " + Rezept.prescriptor.first_name + ", " +
    Rezept.prescriptor.last_name);

    //Loop über alle Produkte des Rezept

    foreach (Product Product in Rezept.prescription.products)
    {
        MessageBox.Show("(" + Product.line_id + ") Anwendung: " +
        Product.remark);
    }

    //Soap Faults (Fehlermeldungen des Soap Servers) können mittels
    System.ServiceModel.FaultException direkt abgefangen werden
}
catch (System.ServiceModel.FaultException ex)
{
    MessageBox.Show("Soap Fault" + '\n' + ex.Message);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message + '\n' + ex.StackTrace);}
}
```