



Interface description LAB system - TRIX

TRIX LAB connection

Auteur(s) J.W. van Markus
Datum september 2011
Ons kenmerk TRIX LAB connection v1.5h.doc
Status FINAL

Approved by:

Role	Name	Handtekening	Date
Opsteller	J.W. van Markus		
Opdrachtgever	R. Smeenk		
Beheerder	C. Veldhoven		
Projectleider	R. Smeenk		

Distribution:

- Approvers**
- Philips-LABOSYS**
- MIPS-GLIMS**
- Technidata-TDLIMS**
- MCS-PROMED**
- Isoft-LABZIS**
- SGS-CORTEX**
- SYSMEX-MOLIS**

	Index	page.
1	Goal	3
2	Process	3
3	Message definition	3
3.1	Find patient in TRIX (from LAB system)	3
3.2	Get Patient info	4
4	Browser	4
5	Security	4
6	Exception Handling	5
7	Example messages	6
	Appendix A – Communication Scheme	8
	Appendix B – Webservice definition (WSDL)	8

1 Goal

Connection between TRIX and the LAB system to exchange information about availability of a matching patient record in TRIX, and retrieval of name and address of a patient from the LAB system.

2 Process

When opening a patient record in the LAB system, a query to the TRIX system should be triggered to find patient related information regarding transfusion reactions.

This query should be done in the background when the patient record is being viewed by personnel.

When this query results in found patient records (single or multiple), a button should be enabled to view these search results. (Somehow the user should be alerted to the fact that a record has been found in TRIX, through blinking or any other means)

When TRIX is opened from the Lab system a list of patients will be displayed. The patient records can be viewed separately. There will be a number of buttons for actions.

The TRIX system is equipped with its own username/password security. To prevent secondary login the URL specifies a username and password. It should be possible to store this username and password to be used and specified on the URL which opens TRIX.

3 Message definition

3.1 Find patient in TRIX (from LAB system)

For the communication between the LAB system and the TRIX server a web service has been developed. The WSDL description file has been added in Appendix B. This service requires authentication as described in chapter 5.

3.1.1 URL

The URL to access the TRIX Webservice is:
`http://<trixserver>/trix/trix.nsf/WebServiceZoek`

3.1.2 Query "ZoekPatient"

The "ZoekPatient" has five parameter values:

- Version (currently, specify "1.0");
- Patient name of birth;
- Date of birth;
- BSN (if unknown or empty, specify "*");
- Patient id (for laboratory)

For an example see chapter 7.

3.1.3 Response “ZoekPatient”

The response from TRIX will be a standard Web Service response. The response will contain the amount of TRIX patients who match the given search query. For an example see chapter 7.

3.2 Get Patient info

For the communication between the LAB system and the TRIX server a web service has been developed. The WSDL description file has been added in Appendix B.

3.2.1 Query “getPatient”

The “getPatient” action initiated from TRIX has one parameter value:

- Patient ID;

This patient ID is the patient ID for the hospital where the patient is created in TRIX.

3.2.2 Response “getPatient”

The response from the Labsystem will be a standard Web Service response. The response will contain the following details about the patient:

BSN:	BSN number.
Eigennaam:	Last name.
EigennaamVoorvoegsels:	Last name prefixes.
Voorletters:	Initials.
Partnernaam:	Partner last name.
PartnernaamVoorvoegsels:	Partner last name prefixes.
Geslacht:	Gender (One letter, M or V).
Geboortedatum:	Date of birth (xsl:date, YYYY-MM-DD).
Adres:	Street address.
Postcode:	Zip code.
Woonplaats:	City.
Huisnummer:	Street address number.
HuisnummerToevoeging:	Number extension.

4 Browser

TRIX currently requires Internet Explorer 6.0 or later.

5 Security

For security the TRIX server (Domino) provides basic authentication (username and password). It should be possible for the LAB system to include these in the communication from the started browser window, so the user does not receive the logon prompt.

For the current implementation it is not needed to encrypt the data from the TRIX server to the LAB system or browser window. The TRIX server and the user's workstation will both be on the internal network of the Hospital.

If the future requires encryption this should be possible though, without much programming needs. The TRIX server can provide SSL encryption when needed, but it will not be used.

6 Exception Handling

The LAB system cannot access the TRIX server to send query:

The LAB system should inform the user that the TRIX server cannot be accessed and that an administrator should be contacted as soon as possible. (If possible the LAB system should also send an email message to a definable administrator (or administrator group))

The LAB system can send the query to the TRIX server, but there is no response, or the response is unreadable.

The LAB system should retry at least 3 times regarding a definable wait time. After the TRIX server cannot respond in a normal way a message should be presented to the user to inform him of the problem. The administrator should be contacted and mailed like the previous exception.

The TRIX server does not respond on the provided URL from response

There should be a user instruction to ask for administrator help when this occurs. If possible, the LAB system should monitor the browser window for timeouts.

The TRIX server responds with an error on the provided URL from response

There should be a user instruction to ask for administrator help when this occurs. If possible, the LAB system should monitor the browser window for HTTP Error 400 and 500 messages.

The TRIX-server responds with a fault string due to an internal error on the query

For an example see chapter 7.

7 Example messages

Example request from labssystem to TRIX

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAPENV:Envelope xmlns:SOAPENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAPENV:Body>
<ns0:zoekPatient xmlns:ns0="urn:DefaultNamespace" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<versie xsi:type="xsd:string">1.0</versie>
<patientNaam xsi:type="xsd:string">Verheul</patientNaam>
<gebDat xsi:type="xsd:date">1967-04-02</gebDat>
<BSN xsi:type="xsd:string">194154372</BSN>
<pid xsi:type="xsd:string">2364532</pid>
</ns0:zoekPatient>
</SOAPENV:Body>
</SOAPENV:Envelope>
```

Example response from TRIX to labssystem

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<ns1:zoekPatientResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="urn:DefaultNamespace">
<zoekPatientReturn xsi:type="xsd:int">
1
</zoekPatientReturn>
</ns1:zoekPatientResponse>
</soapenv:Body>
</soapenv:Envelope>
```

Explanation response from TRIX to labssystem

In the response you see the number "1". TRIX processes the information sent in the request in three stages.

First on the national BSN-number, if no hit next on the patientnumber (pid) and (if no hit either) last on the combination of patientNaam (patient birthname) and gebDat (birthdate).

When no patient is found, TRIX will sent the number 0 as response. The receiving labssystem should do nothing.

When 1 patient is found TRIX will sent the number 1 as response, when two patients are found (possible on birthname and birthdate) the number 2 and so on.

When the response though is 1 or more the labssystem should present an alert (popup screen) indicating that TRIX found one ore more patients.

After the alert, there should be a TRIX-button present where the user can click on.
The url behind that button is specified in appendix A.

Example fault response from TRIX to labssystem

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Body>
<soapenv:Fault>
<faultcode>soapenv:Server.generalException</faultcode>
<faultstring>NotesException: De TR!X index is corrupt geraakt, neem direct contact
op met de TR!X helpdesk</faultstring>
<detail/>
</soapenv:Fault>
</soapenv:Body>
</soapenv:Envelope>
```

Example request from TRIX to labssystem to obtain NAW-data

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Body>
<ns0:TRIXService xmlns:ns0="urn:DefaultNamespace" SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<PID xsi:type="xsd:string">346287364</PID>
</ns0:TRIXService>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Waarbij TRIXService kan worden vervangen door uw eigen naam van uw web service.
Ook wordt het patient ID meegestuurd in de URL waarmee de service wordt aangeropen
bv:

http://labsteeem.ipadres/trix.service.cls?soap_method=patient&patientnr=346287364

Appendix A – Communication Scheme

The url to login to TRIX from the LAB system to view the patient record is:

<http://trix/names.nsf?login&username=%username%&password=%password%&redirectto=/trix/trix.nsf/wf-zoek?readform%26naam=%patientname%&26gebdat=%dateofbirth%&26bsn=%BSN%&26pid=%PID%>

The red and underlined items in the url have to be programmed flexible. The user must be able to specify and change these parameters.

%username% should be replaced by the Lab system with the TRIX username of the person logged in in the Lab system

%password% should be replaced by the Lab systems with corresponding TRIX password for the same user.

%patientname% should be replaced with the patient last name.

%dateofbirth% should be replaced with the patient date of birth in the format yyyy-mm-dd.

%BSN% should be replaced with the patient BSN (when available).

%PID% should be replaced with the patient hospital id

Appendix B – Webservice definition (WSDL)

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions targetNamespace="urn:DefaultNamespace"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:apachesoap="http://xml.apache.org/xml-soap"
xmlns:impl="urn:DefaultNamespace" xmlns:intf="urn:DefaultNamespace"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:wsd="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdsoap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <wsdl:types>
    <schema targetNamespace="urn:DefaultNamespace" xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://schemas.xmlsoap.org/soap/encoding/" />
      <complexType name="Patient">
        <sequence>
          <element name="instellingID" nillable="true" type="xsd:string"/>
          <element name="BSN" nillable="true" type="xsd:string"/>
          <element name="huisnummerToevoeging" nillable="true" type="xsd:string"/>
          <element name="huisnummer" nillable="true" type="xsd:string"/>
          <element name="plaats" nillable="true" type="xsd:string"/>
          <element name="postcode" nillable="true" type="xsd:string"/>
          <element name="straat" nillable="true" type="xsd:string"/>
          <element name="gebDat" nillable="true" type="xsd:date"/>
          <element name="geslacht" nillable="true" type="xsd:string"/>
          <element name="voorvoegselsPartner" nillable="true" type="xsd:string"/>
          <element name="naamPartner" nillable="true" type="xsd:string"/>
          <element name="initialen" nillable="true" type="xsd:string"/>
          <element name="voorvoegsels" nillable="true" type="xsd:string"/>
          <element name="naam" nillable="true" type="xsd:string"/>
          <element name="patientID" nillable="true" type="xsd:string"/>
          <element name="trixID" nillable="true" type="xsd:string"/>
          <element name="found" type="xsd:boolean"/>
        </sequence>
      </complexType>
    </schema>
  </wsdl:types>
  <wsdl:message name="zoekPatientResponse">
    <wsdl:part name="zoekPatientReturn" type="xsd:int"/>
  </wsdl:message>
  <wsdl:message name="zoekPatientByPIDRequest">
    <wsdl:part name="versie" type="xsd:string"/>
    <wsdl:part name="PID" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="zoekPatientQueryResponse">
    <wsdl:part name="zoekPatientQueryReturn" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="zoekPatientByPIDResponse">
    <wsdl:part name="zoekPatientByPIDReturn" type="impl:Patient"/>
  </wsdl:message>
  <wsdl:message name="zoekPatientQueryRequest">
    <wsdl:part name="versie" type="xsd:string"/>
    <wsdl:part name="patientNaam" type="xsd:string"/>
  </wsdl:message>

```



```

<wsdl:part name="gebDat" type="xsd:date"/>
<wsdl:part name="BSN" type="xsd:string"/>
</wsdl:message>
<wsdl:message name="zoekPatientRequest">
  <wsdl:part name="versie" type="xsd:string"/>
  <wsdl:part name="patientNaam" type="xsd:string"/>
  <wsdl:part name="gebDat" type="xsd:date"/>
  <wsdl:part name="BSN" type="xsd:string"/>
  <wsdl:part name="pid" type="xsd:string"/>
</wsdl:message>
<wsdl:portType name="ZoekPortType">
  <wsdl:operation name="zoekPatient" parameterOrder="versie patientNaam gebDat BSN pid">
    <wsdl:input message="impl:zoekPatientRequest" name="zoekPatientRequest"/>
    <wsdl:output message="impl:zoekPatientResponse" name="zoekPatientResponse"/>
  </wsdl:operation>
  <wsdl:operation name="zoekPatientQuery" parameterOrder="versie patientNaam gebDat BSN">
    <wsdl:input message="impl:zoekPatientQueryRequest" name="zoekPatientQueryRequest"/>
    <wsdl:output message="impl:zoekPatientQueryResponse" name="zoekPatientQueryResponse"/>
  </wsdl:operation>
  <wsdl:operation name="zoekPatientByPID" parameterOrder="versie PID">
    <wsdl:input message="impl:zoekPatientByPIDRequest" name="zoekPatientByPIDRequest"/>
    <wsdl:output message="impl:zoekPatientByPIDResponse" name="zoekPatientByPIDResponse"/>
  </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ZoekPortSoapBinding" type="impl:ZoekPortType">
  <wsdlsoap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="zoekPatient">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="zoekPatientRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:DefaultNamespace" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="zoekPatientResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:DefaultNamespace" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="zoekPatientQuery">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="zoekPatientQueryRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:DefaultNamespace" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="zoekPatientQueryResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:DefaultNamespace" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
  <wsdl:operation name="zoekPatientByPID">
    <wsdlsoap:operation soapAction=""/>
    <wsdl:input name="zoekPatientByPIDRequest">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:DefaultNamespace" use="encoded"/>
    </wsdl:input>
    <wsdl:output name="zoekPatientByPIDResponse">
      <wsdlsoap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="urn:DefaultNamespace" use="encoded"/>
    </wsdl:output>
  </wsdl:operation>
</wsdl:binding>
<wsdl:service name="ZoekService">
  <wsdl:port binding="impl:ZoekPortSoapBinding" name="ZoekPort">
    <wsdlsoap:address location="http://localhost"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```