Quick start guide for preparing the X-Road XForms complex service
Version 2.0

1. Table of contents

2. Introduction

The aim of this guide is to give a brief overview of how to create a complex service from two simple services. The service can be used in MISP2. We will look at some simpler XForms forms and the important XForms elements of the simple service beforehand to get to know the semantics of the language.

The guide is meant for developers who are developing the X-Road service in the MISP2 application. MISP2 uses an X-Road request presentation based on XForms technology. XForms is a form description language, an offspring of HTML forms in XHTML 2.0.

3. XForms

XForms is a text-formatting language used to collect input data from web forms.
XForms uses XML for data definition and HTML or XHTML for data display. XForms separates the logic of data from its presentation. This way, the data is defined on its own, independently from the user's activity.

3.1 History

XForms 1.0 (*Third Edition*) was released on 29 October 2007. The original XForms specification became an official W3C Recommendation on 14 October 2003. The version 1.1, which introduced a number of improvements, reached the same status on 20 October 2009. The latest XForms version is 2.0 (as at 25 September 2017)

3.2 Advantages and disadvantages

Advantages of XForms

- Model-View-Controller (MVC) architecture
- Declarative programming – easier to learn, maintain, debug
- The user interface has many options for solving complex issues (e.g. dates, numbers, and ranges)
- Compatibility with other XML standards (CSS, CML, Schema, XPath)
- Less Javascript
- Localization

Disadvantages of XForms

- Low browser support
- Slow with large forms
    - Large = 200–300 fields
- Few supporting materials

3.3 XForms function and what does it consist of

The goal of XForms is to collect data in forms. In order to use XForms, the XForms elements have to be declared with the namespace `xmlns:xforms="http://www.w3.org/2002/xforms"`. XForms framework is divided into two parts:

- XForms model
- XForms user interface

3.3.1 XForms model

XForms model is used for describe data. The data template is an XML document.

```
<xforms:model>
    <xforms:instance xmlns="">
        <person>
            <fname/>
            <lname/>
        </person>
    </xforms:instance>
    <xforms:submission id="form1" action="submit.asp" method="get"/>
</xforms:model>
```

In the example above, we define inside the `<instance>` element our XML template for data to be collected and in the `<submission>` element, we describe how to submit the data. Inside the Submission, we establish the request identifier (`id="form1"`), the URL to where we submit data (`action="submit.asp"`) and the method that we use to submit data (`method="get"`).

If our XML template looks initally like this:

```
<person>
    <fname/>
    <lname/>
</person>
```

Then after collecting data it could look like this:

```
<person>
    <fname>John</fname>
    <lname>Smith</lname>
</person>
```

3.3.2 XForms user interface

XForms user interface is used to display and input data. The user interface elements of XForms are called controls (*input controls*).

```
<xforms:input ref="fname">
    <xforms:label>First Name</xforms:label>
</xforms:input>
<xforms:input ref="lname">
    <xforms:label>Last Name</xforms:label>
</xforms:input>
<xforms:submit submission="form1">
    <xforms:label>Submit</xforms:label>
</xforms:submit>
```

Here, we use two `<input>` elements to collect input data. Attributes **ref="fname"** and **ref="lname"** point to the `<fname/>` and `<lname/>` elements of our data structure.

In the `<Submit>` control, the **submission="form1"** attribute refers to the `<submission>` element in the XForms model. A submit element is usually displayed as a button.

With XForms, every input control has a required `<label>` element.

3.3.3 Container

As the XForms document does not exist, we have to move it inside another XML document, e.g. XHTML.

```
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:xforms="http://www.w3.org/2002/xforms">
    <head>
        <xforms:model>
            <xforms:instance xmlns="">
                <person>
                    <fname/>
                    <lname/>
                </person>
            </xforms:instance>
            <xforms:submission id="form1" action="submit.asp" method="get"/>
        </xforms:model>
    </head>
    <body>
        <xforms:input ref="fname">
            <xforms:label>First Name</xforms:label>
        </xforms:input>
        <xforms:input ref="lname">
            <xforms:label>Last Name</xforms:label>
        </xforms:input>
        <xforms:submit submission="form1">
            <xforms:label>Submit</xforms:label>
        </xforms:submit>
    </body>
</html>
```

Visually it looks like this:

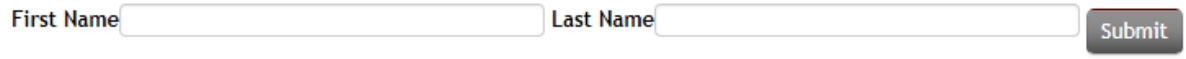


Figure1 XForms example

Additional information about the XForms model and the user interface controls can be found on:
https://www.w3.org/TR/2003/REC-xforms-20031014/index.html#contents
https://en.wikibooks.org/wiki/XForms
http://w3schools.sinsixx.com/xforms/default.asp.htm

3.4 XPath

XPath is a W3C standard syntax for defining parts of XML documents. XPath uses path expressions to identify nodes in an XML document.

This XPath expression:

```
/person/fname
```

addresses the 'fname' node in the XML document:

```
<person>
    <fname>Hege</fname>
    <lname>Refsnes</lname>
</person>
```

Additional information about XPath functions can be found on the following link: https://www.w3schools.com/xml/xsl_functions.asp.

3.5 Let us get to know the code

3.5.1 Example 1 – Welcome

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:xforms="http://www.w3.org/2002/xforms">
<head>
<xforms:model>
    <xforms:instance xmlns="">
        <person xmlns=""> <givenName /> </person>
    </xforms:instance>
</xforms:model>
</head>
<body>
    <h1>Hello</h1>
    <p>
        <xforms:input ref="/person/givenName" incremental="true">
            <xforms:label xml:lang="et">Sisestage oma nimi: </xforms:label>
            <xforms:label xml:lang="en">Enter your name: </xforms:label>
        </xforms:input>
    </p>
    <p>
        <xforms:output
            value="if (normalize-space(/person/givenName) = '') then '' else
concat('Hello ', /person/givenName, '!')" />
    </p>
</body>
</html>
```

Form overview:

- The file begins with an XML declaration that can be excluded.

- Then follows an `<html>` tag with the namespaces used in the file.

- `<head>` content differs from regular HTML code, as we define the model (`<xforms:model>`) where we can create our XForms data structure (`<xforms:instance>`) whose content is an XML object.

- `<body>` section shows both the HTML code and the XForms user interface controls. It begins with a heading, which is inside a regular `<h1>` tag, as the default namespace of the file is the XHTML namespace. Inside the `<p>` tag you can see the `<xforms:input>` control, which we can use to set the value of the respective XML node (XPath) determined in the **ref** attribute. Attribute **incremental="true"** determines the automatic update of a value. `<xforms:label>` is similar to the HTML `<label>` element, which can be used to display explanatory text in front of the input. In the second paragraph, we will display, using an output, a `<xforms:output>` control, whose **value** attribute is also XPath. A condition has been added that if the XPath value is empty or the input is blank, the response will not be displayed. Otherwise, "Hello {input}!" will be displayed.

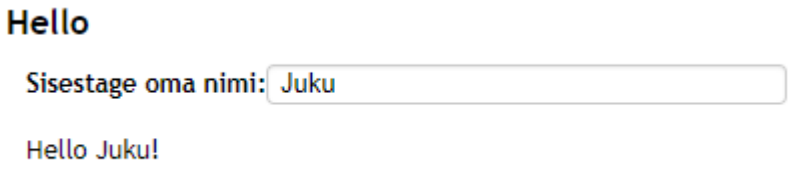


Figure2 Hello

3.5.2 Example 2 – Hello World

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:xforms="http://www.w3.org/2002/xforms"
    xmlns:events="http://www.w3.org/2001/xml-events">
<head>
<xforms:model>
    <xforms:instance xmlns="">
        <data xmlns="" />
    </xforms:instance>
</xforms:model>
</head>
<body>
    <h1>Hello World</h1>
    <p>
        <xforms:trigger>
            <xforms:label xml:lang="et">Nupp</xforms:label>
            <xforms:label xml:lang="en">Button</xforms:label>
            <xforms:hint xml:lang="et">Seda nuppu vajutades ilmub teade.</xforms:hint>
            <xforms:hint xml:lang="en">A message will be displayed when clicking this
button.</xforms:hint>
            <xforms:message level="modal" events:event="DOMActivate">Hello
World!</xforms:message>
        </xforms:trigger>
    </p>
</body>
</html>
```

Form overview:

- In comparison to the previous form, we notice one difference – a new namespace **events** has been added. With *Event* commands we can control the form's activity regarding the trigger time of a command.

- In the `<body>` tag, we create a new button by using the `<xforms:trigger>` control, in which we add three controls.
    - `<xforms:label>` – we give the button text.
    - `<xforms:hint>` – while hovering over the button with the cursor, an explanatory text will appear.
    - `<xforms:message>` – a message that will be displayed after clicking the button. Attribute **level="modal"** gives the message an appearance (the value is the same by default), **events:event="DOMActivate"** triggers the command.

- If needed, the button can be replaced with a link, which can be done by adding **appearance="minimal"** attribute to the `<xforms:trigger>` control.

Figure3 Hello World

3.5.3 Example 3 – Selection list

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:xforms="http://www.w3.org/2002/xforms">
<head>
<xforms:model>
    <xforms:instance xmlns="">
        <data>
            <selectedElement />
        </data>
    </xforms:instance>
</xforms:model>
</head>
<body>
    <h1>Selection</h1>
    <p>
        <xforms:select1 ref="selectedElement">
            <xforms:label xml:lang="et">Valikuloend:</xforms:label>
            <xforms:label xml:lang="en">Selection:</xforms:label>
            <xforms:item>
                <xforms:label xml:lang="et">Esimene</xforms:label>
                <xforms:label xml:lang="en">First</xforms:label>
                <xforms:value>first</xforms:value>
            </xforms:item>
            <xforms:item>
                <xforms:label xml:lang="et">Teine</xforms:label>
                <xforms:label xml:lang="en">Second</xforms:label>
                <xforms:value>second</xforms:value>
            </xforms:item>
            <xforms:item>
                <xforms:label xml:lang="et">Kolmas</xforms:label>
                <xforms:label xml:lang="en">Third</xforms:label>
                <xforms:value>third</xforms:value>
            </xforms:item>
            <xforms:item>
                <xforms:label xml:lang="et">Neljas</xforms:label>
                <xforms:label xml:lang="en">Fourth</xforms:label>
                <xforms:value>fourth</xforms:value>
            </xforms:item>
            <xforms:item>
                <xforms:label xml:lang="et">Viies</xforms:label>
                <xforms:label xml:lang="en">Fifth</xforms:label>
                <xforms:value>fifth</xforms:value>
            </xforms:item>
        </xforms:select1>
    </p>
    <p>
        <xforms:output value="selectedElement">
            <xforms:label xml:lang="et">Valitud: </xforms:label>
            <xforms:label xml:lang="en">Selected: </xforms:label>
        </xforms:output>
    </p>
</body>
</html>
```

Form overview:

- Inside the `<head>` part, we create a data structure, in which we keep the value currently selected in the selection list.

- Inside the `<body>` tag, we create a selection list, `<xforms:select1>` control, to which we add the **ref="selectedElement"** attribute – this saves the selected value of the element. Inside

the selection list, we define the options, using `<xforms:item>` control, in which we define its' displayed name (`<xforms:label>`) and value (`<xforms:value>`).

- It should be noted that inside the `<xforms:output>` output, the response is not displayed after the first loading of the form but only after we have made a choice.

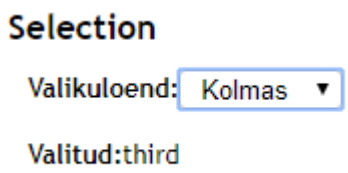


Figure4 Selection list

3.5.4 Example 4 – Input form

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:xforms="http://www.w3.org/2002/xforms"
    xmlns:events="http://www.w3.org/2001/xml-events">
<head>
<xforms:model>
    <xforms:instance id="personal-data" xmlns="">
        <people>
            <person>
                <firstName />
                <lastName />
                <birthDate />
                <contactInfo>
                    <phone />
                    <email />
                </contactInfo>
            </person>
        </people>
    </xforms:instance>
    <xforms:bind nodeset="instance('personal-data')">
        <xforms:bind nodeset="person">
            <xforms:bind nodeset="firstName" type="xforms:string" />
            <xforms:bind nodeset="lastName" type="xforms:string" />
            <xforms:bind nodeset="birthDate" type="xforms:date" />
            <xforms:bind nodeset="contactInfo">
                <xforms:bind nodeset="phone" type="xforms:integer" />
                <xforms:bind nodeset="email" type="xforms:string" />
            </xforms:bind>
        </xforms:bind>
    </xforms:bind>
</xforms:model>
<style>
    .input-w150 input {width:150px !important;}
    .input-50 input {width:50px !important;}
</style>
</head>
<body>
    <h1>Insertion</h1>
    <p>
        <xforms:repeat nodeset="instance('personal-data')/person[position() > 1]"
```

```
id="repeat-list">
            <xforms:output ref="firstName">
                <xforms:label xml:lang="et">Eesnimi</xforms:label>
                <xforms:label xml:lang="en">First name</xforms:label>
            </xforms:output>
            <xforms:output ref="lastName">
                <xforms:label xml:lang="et">Perekonnanimi</xforms:label>
                <xforms:label xml:lang="en">Last name</xforms:label>
            </xforms:output>
            <xforms:output ref="birthDate">
                <xforms:label xml:lang="et">Sünniaeg</xforms:label>
                <xforms:label xml:lang="en">Date of birth</xforms:label>
            </xforms:output>
            <xforms:output ref="contactInfo/phone">
                <xforms:label xml:lang="et">Telefon</xforms:label>
                <xforms:label xml:lang="en">Telephone</xforms:label>
            </xforms:output>
            <xforms:output ref="contactInfo/email">
                <xforms:label xml:lang="et">E-mail</xforms:label>
                <xforms:label xml:lang="en">E-mail</xforms:label>
            </xforms:output>
        </xforms:repeat>
        <xforms:group ref="instance('personal-data')/person" appearance="compact">
            <xforms:input ref="firstName" class="input-w150">
                <xforms:label xml:lang="et">Eesnimi</xforms:label>
                <xforms:label xml:lang="en">First name</xforms:label>
            </xforms:input>
            <xforms:input ref="lastName" class="input-w150">
                <xforms:label xml:lang="et">Perekonnanimi</xforms:label>
                <xforms:label xml:lang="en">Last name</xforms:label>
            </xforms:input>
            <xforms:input ref="birthDate" class="input-w50">
                <xforms:label xml:lang="et">Sünniaeg</xforms:label>
                <xforms:label xml:lang="en">Date of birth</xforms:label>
            </xforms:input>
            <xforms:input ref="contactInfo/phone" class="input-w150">
                <xforms:label xml:lang="et">Telefon</xforms:label>
                <xforms:label xml:lang="en">Telephone</xforms:label>
            </xforms:input>
            <xforms:input ref="contactInfo/email" class="input-w150">
                <xforms:label xml:lang="et">E-mail</xforms:label>
                <xforms:label xml:lang="en">E-mail</xforms:label>
            </xforms:input>
        </xforms:group>
        <xforms:group ref="instance('personal-data')">
            <xforms:trigger>
                <xforms:label xml:lang="et">Lisa</xforms:label>
                <xforms:label xml:lang="en">Insert</xforms:label>
                <xforms:action events:event="DOMActivate">
                    <xforms:insert nodeset="person[last()]" at="last()" position="after"
/>
                    <xforms:setvalue ref="instance('personal-
data')/person[last()]/firstName" value="instance('personal-data')/person/firstName" />
                    <xforms:setvalue ref="instance('personal-
data')/person[last()]/lastName" value="instance('personal-data')/person/lastName" />
                    <xforms:setvalue ref="instance('personal-
data')/person[last()]/birthDate" value="instance('personal-data')/person/birthDate" />
                    <xforms:setvalue ref="instance('personal-
data')/person[last()]/contactInfo/phone" value="instance('personal-
data')/person/contactInfo/phone" />
                    <xforms:setvalue ref="instance('personal-
data')/person[last()]/contactInfo/email" value="instance('personal-
data')/person/contactInfo/email" />
                </xforms:action>
            </xforms:trigger>
            <xforms:trigger ref=".[count(person) > 1]">
```

```
                <xforms:label xml:lang="et">Kustuta</xforms:label>
                <xforms:label xml:lang="en">Delete</xforms:label>
                <xforms:action events:event="DOMActivate">
                    <xforms:delete nodeset="person[last()]" at="last()"/>
                </xforms:action>
            </xforms:trigger>
        </xforms:group>
    </p>
</body>
</html>
```

Form overview:

- Inside `<xforms:model>`, we once again create our data structure, but in addition, we will define our validation rules. It should be noted that the validation rules will be applied only during the (`<xforms:submission>`) data submission, which we will not do here. You only need to add **type="xforms:date"** to the date for the calender to appear next to that field.

- In addition, you can see the `<style>` tag inside the`<head>` tag. You can also change the style on the XForms forms by using the `<style>` tag. With this example, we will only change the width of the table's columns. More information about the XForms styling can be found here: here:https://www.w3.org/TR/2003/REC-xforms-20031014/sliceF.html.

- Inside the `<body>`, we can see the control `<xforms:repeat>`, which can be used to go through all the data in the list. Using the **nodeset** attribute, we determine which list we put in the cycle; in addition, a condition has been set inside the square brackets that we skip the first element (**instance('personal-data')/person**) (we will use it for the template data). Inside the cycle we will, however, write the data on the display by using `<xforms:output>` control. This is followed by `<xforms:group>`, which we use to group our input fields into one table. Attribute **appearance="compact"** gives our table a horizontal formation.

- We use the `<xforms:trigger>` control to create the buttons to add and delete rows. Inside it, we have to add control `<xforms:action events:event="DOMActivate">`, which gives our button the command to do something when the button is clicked. `<xforms:insert>` control can be used to add data to the list; it is necessary to select three attributes. With **Nodeset**, we determine the list to which we add data. Attribute **at** is an index which we use to determine a target; in this case, the last object on the list. **Position** determines where the data is added to relative to the target.
  To set the value of the new object, we use the `<xforms:setvalue>` command with whose **ref** attribute we set the location where the value is saved, and we use **value** to set value. As we use the first object of the list as a template, it is therefore a value and the last object of the list is where we save.
  Last, we create a new button and give it an ability with the `<xforms:delete>` control to delete the object at the end of the list.

Figure 5 Input form

4. Necessary XForms elements of the X-Road service

When you generate an XForms description for the X-Road service via the MISP application, the form will get certain elements via which the data is being exchanged between the input, output, and user interface request controls.
To explain the code, we use Persons List query's XForms (*aktorstest-db01.personDetails.v1*).

4.1.1 Necessary XForms elements of the X-Road service – Model

Inside the `<xforms:model> tag`, which is inside `<xhtml:head>`, data structures (`<xforms:instance>`), validation rules (`<xforms:bind>`) and queries (`<xforms:submission>`) are defined.

- `<xforms:instance>` – data structure, which in the case of X-Road service is XML SOAP. Output data structure is provisional, because in case of a successful request submission, the data structure will be replaced with request responses. In addition to the input and output of services, provisional data structures and classificators can also be created.

```
<xforms:instance id="personList.input">
    <SOAP-ENV:Envelope>
        <SOAP-ENV:Header>
            <xrd:protocolVersion>4.0</xrd:protocolVersion>
            <xrd:id/>
            <xrd:userId>EE</xrd:userId>
            <xrd:service iden:objectType="SERVICE">
                <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                <iden:memberClass>COM</iden:memberClass>
                <iden:memberCode>11333578</iden:memberCode>
                <iden:subsystemCode>aktorstest-db01</iden:subsystemCode>
                <iden:serviceCode>personList</iden:serviceCode>
                <iden:serviceVersion>v1</iden:serviceVersion>
            </xrd:service>
            <xrd:client iden:objectType="SUBSYSTEM">
                <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                <iden:memberClass/>
                <iden:memberCode/>
                <iden:subsystemCode/>
            </xrd:client>
        </SOAP-ENV:Header>
        <SOAP-ENV:Body>
            <ns5:personList xmlns:ns5="http://aktorstest.x-road.ee/producer">
                <request>
                    <givenName/>
                    <surname/>
                </request>
            </ns5:personList>
        </SOAP-ENV:Body>
    </SOAP-ENV:Envelope>
</xforms:instance>
<xforms:instance id="personList.output">
    <dummy/>
</xforms:instance>
```

- `<xforms:bind>` – validation rules controlled during request submissions. Here we bind our input and output data (**nodeset**) into certain data types (**type**). It is also possible to use other attributes in addition to data types.
    - **relevant** – condition; if it is met, that part will be displayed in your form
    - **required** – required, data field has to have a value
    - **readonly** – read-only, the data field value can not be changed
    - **constraint** – constraint, for setting minimum/maximum value
    - **calculate** – calculation

```
<xforms:bind nodeset="instance('personList.input')/SOAP-ENV:Body">
    <xforms:bind nodeset="tns:personList">
        <xforms:bind nodeset="request">
            <xforms:bind nodeset="givenName" type="xforms:string"/>
            <xforms:bind nodeset="surname" type="xforms:string"/>
        </xforms:bind>
    </xforms:bind>
</xforms:bind>
<xforms:bind nodeset="instance('personList.output')/SOAP-ENV:Body">
    <xforms:bind nodeset="tns:personListResponse">
        <xforms:bind nodeset="response">
            <xforms:bind nodeset="faultCode" type="xforms:string"/>
            <xforms:bind nodeset="faultString" type="xforms:string"/>
            <xforms:bind nodeset="person">
                <xforms:bind nodeset="givenName" type="xforms:string"/>
                <xforms:bind nodeset="surname" type="xforms:string"/>
                <xforms:bind nodeset="ssn" type="xforms:string"/>
            </xforms:bind>
        </xforms:bind>
    </xforms:bind>
</xforms:bind>
```

- `<xforms:submission>` – requests. Inside the submission, it is possible to determine the request input data, output data, how they are formatted, and error message processing.
    - **action** – address where we submit our request. This is not taken into account in the MISP portal, but instead a certain address is used, which is determined by an administrator.
    - **mediatype** – we determine that our request is application/soap+xml type.
    - **encoding** – encoding
    - **ref** – input data
    - **method** – HTTP method used to transmit the request.
    - **replace** – we determine what will be done with the output data; in this case, we save them inside *instance*.
    - **instance** – *Instances* where the output data is saved.
    - `<xforms:toggle>` determines which (`<xforms:case>`) view we display after a successful request (`xforms-submit-done`).
    - `<xforms:message>` – a message that is displayed after an unsuccessful request**. level="modal"** attribute gives our message an appearance and with the **events:event="xforms-submit-error"** attribute we make sure that it is displayed only if the request is unsuccessful.

```
<xforms:submission id="personList.submission"
                   action="http://192.168.219.190:8080/aktorstest-xroad-
v6/services/aktorstestServicePort"
                   mediatype="application/soap+xml; charset=UTF-8; action="
```

```
                            encoding="UTF-8"
                            ref="instance('personList.input')"
                            method="post"
                            replace="instance"
                            instance="personList.output">
                <xforms:setvalue ref="instance('temp')/relevant"
                             value="false()"
                             events:event="xforms-submit"/>
                <xforms:setvalue ref="instance('personList.input')/SOAP-ENV:Header/*:id"
                             value="digest(string(random()), 'SHA-1', 'hex')"
                             events:event="xforms-submit"/>
                <xforms:toggle case="personList.response" events:event="xforms-submit-
done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                             value="true()"
                             events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                             value="true()"
                             events:event="xforms-submit-error"/>
                <xforms:message level="modal" events:event="xforms-submit-error">
                    <xforms:output xml:lang="et"
                                 value="if (event('error-type') = 'submission-in-progress')
then 'Üks päring juba käib!'        else if (event('error-type') = 'no-data') then 'Pole
andmeid, mida saata!'        else if (event('error-type') = 'validation-error') then
'Valideerimise viga!'        else if (event('error-type') = 'parse-error') then 'Viga
vastuse töötlemisel!'        else if (event('error-type') = 'resource-error') then
'Päringu vastus ei ole XML!'        else if (event('error-type') = 'target-error') then
'Sihtkoha viga!'        else 'Sisemine viga!'"/>
                    <xforms:output xml:lang="en"
                                 value="if (event('error-type') = 'submission-in-progress')
then 'Submission already started!'        else if (event('error-type') = 'no-data') then
'No data to submit!'        else if (event('error-type') = 'validation-error') then
'Validation error!'        else if (event('error-type') = 'parse-error') then 'Error
parsing response!'        else if (event('error-type') = 'resource-error') then 'Response
is not XML!'        else if (event('error-type') = 'target-error') then 'Target error!'
else 'Internal error!'"/>
                </xforms:message>
            </xforms:submission>
```

- The `<xforms:dispatch>` **targetid** attribute (case id as input) determines which view will be displayed upon first loading.

```
<xforms:dispatch targetid="personList.request" name="xforms-select"
events:event="xforms-ready"/>
```

4.1.2 Necessary XForms elements of the X-Road service – User interface

User interface controls of XForms are inside the <xhtml:body> tag.

- The previously mentioned views can be determined with<xforms:case> tags that are inside <xforms:switch>.

```
<xforms:switch>
    <xforms:case id="personList.request">
        ...
    </xforms:case>
    <xforms:case id="personList.response">
        ...
    </xforms:case>
</xforms:switch>
```

- <xforms:group> control can group and optimise elements by using **ref** attribute whose value is XPath.

```
<xforms:group ref="instance('personList.input')/SOAP-ENV:Body">
    <xforms:group ref="tns:personList">
        <xforms:group ref="request">
        ...
        </xforms:group>
    </xforms:group>
</xforms:group>
```

- With <xforms:input> we create a field for submitting data. We use **ref** attribute to determine where the submitted data will be saved.
- We use <xforms:label> control to display explanatory text in front of the input field. We use **xml:lang** attribute to define texts with different languages.

```
<xforms:group ref="instance('personList.input')/SOAP-ENV:Body">
    <xforms:group ref="tns:personList">
        <xforms:group ref="request">
            <xforms:input ref="givenName">
                <xforms:label xml:lang="en">Given name</xforms:label>
                <xforms:label xml:lang="et">Eesnimi</xforms:label>
            </xforms:input>
            ...
        </xforms:group>
    </xforms:group>
</xforms:group>
```

- We use <xforms:output> control to display data.

```
<xforms:output ref="givenName">
    <xforms:label xml:lang="en">Given name</xforms:label>
    <xforms:label xml:lang="et">Eesnimi</xforms:label>
</xforms:output>
```

- `<xforms:repeat>` control is used to display data that is located in the list. We use nodeset attribute to set XPath value. We can set its' control an id value with **Id** attribute.

```
<xforms:repeat nodeset="person"
        id="personList_output_tns_personListResponse_response_person">
    <xforms:output ref="givenName">
        <xforms:label xml:lang="en">Given name</xforms:label>
        <xforms:label xml:lang="et">Eesnimi</xforms:label>
    </xforms:output>
    ...
</xforms:repeat>
```

- We use `<xforms:submit>` control to create a button for data submission. Request (*submission*) that we submit is specified in the **submission** attribute.

```
<xforms:submit submission="personList.submission">
    <xforms:label xml:lang="et">Esita päring</xforms:label>
    <xforms:label xml:lang="en">Submit</xforms:label>
</xforms:submit>
```

- We create a button by using `<xforms:trigger>` control. We can give the button functionality with **events:event="DOMActivate"** attribute that has to be inside an *action* type control. As an example we will use `<xforms:toggle>` control – by clicking on the button, we change the view. By giving the **case** attribute one view (*case*) id, we can display the respective view.

```
<xforms:trigger>
    <xforms:label xml:lang="et">Uuesti</xforms:label>
    <xforms:label xml:lang="en">Again</xforms:label>
    <xforms:toggle events:event="DOMActivate" case="personList.request"/>
</xforms:trigger>
```

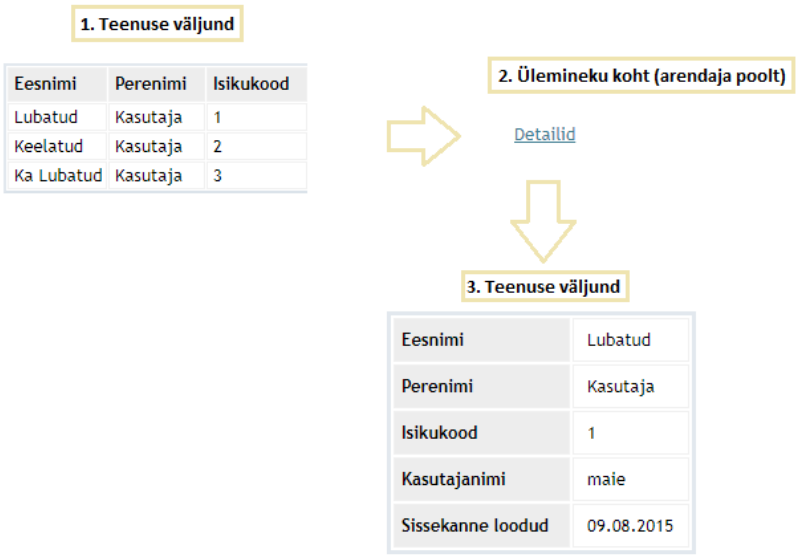
5. Simple services in MISP2

When creating a new complex service, we use two simple services (aktorstest database, WSDL: http://www.aktors.ee/support/xroad/wsdl/aktorstest.wsdl):

- The first shows as an output a person's list that contains social security numbers and names (*aktorstest-db01.personList.v1*)
- The other uses a social security number as an input and shows more specific information about the person (*aktorstest-db01.personDetails.v1*)

Let us assume that the simple service forms have already been generated (using either wsdl2xforms command-line script or the standard form generating functionality of the MISP2 application).

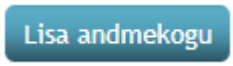
Complex service development idea is based on realising a switch from one service to another: how to use the output (or part of the output) of the first service as the input of the other. The goal of the developer of a new complex service is to find the places in the XForms descriptions where you have to add these commands to realize the switch. This is described by the following scheme:



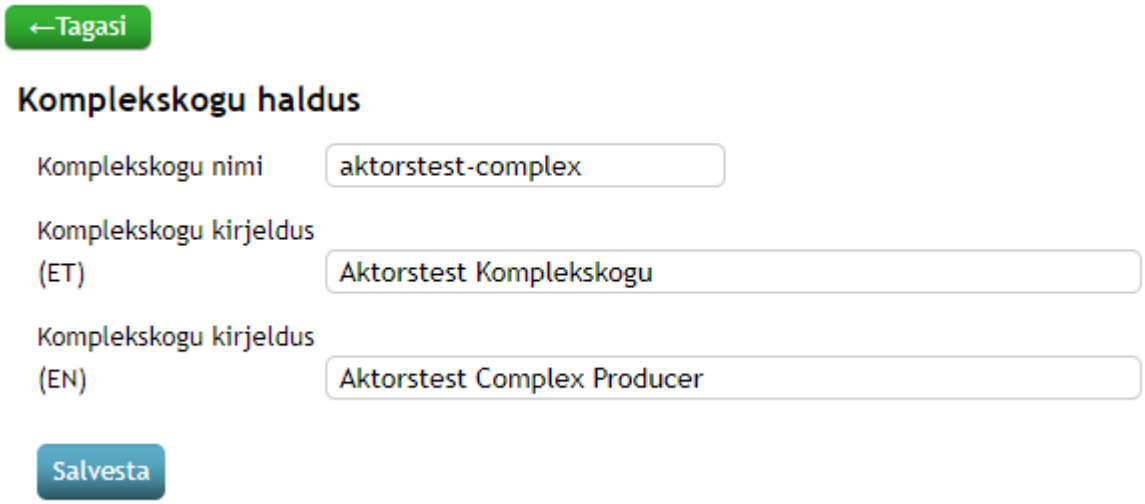
6. Complex services in MISP2

6.1 Adding a complex collection

In order to begin creating a new complex service in the MIPS2 application, it is necessary to add a new complex collection. This can be done by a user in the role of a portal administrator or a service administrator under the Menu option “Services”:

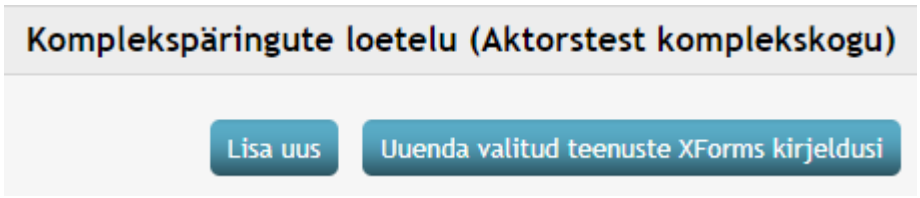


On the complex collection management page, you must enter the complex collection name and a description, and click the “Save” button.

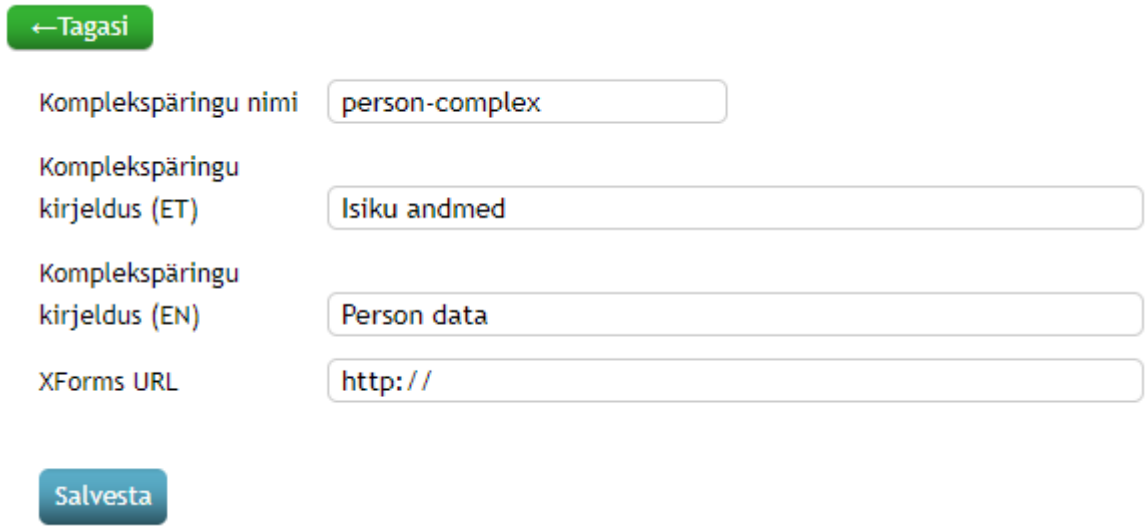


Once the complex collection is saved, it must be visible in the complex collection list:

By clicking on the complex collection name, we can look at the list of complex services in that database. The list is empty in the case of a new complex collection:

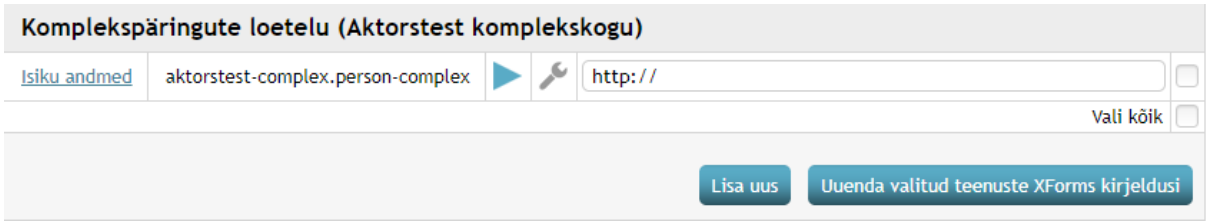


The button “Add new” leads to the page with the metadata of the new complex service where you can enter a systematic name, description, and XForms URL of the new service.



XForms URL is useful when the XForms description of the service is already created and it can be loaded from the entered URL.
Systematic name must be in the format **servicename.vX**, where X is the version number.

Once the new service is saved, it will appear in the service list of the complex collection:



6.2 Ways to develop the service

When clicking on the service name, a screen form will appear with the description of the XForms service. This is one way how to develop the service form. The other way is to use your favorite text editor and afterwards, by using XForms URL, load the given description into the application.

6.3 Service development

Let us copy the XForms description of the first simple service (*aktorstest-db01.personList.v1*) into the form of the new complex service *aktorstest-complex.person-complex*. The resulting document will look like a regular document with an XML/HTML markup, which contains `<xforms:model>` and other XForms-specific elements.

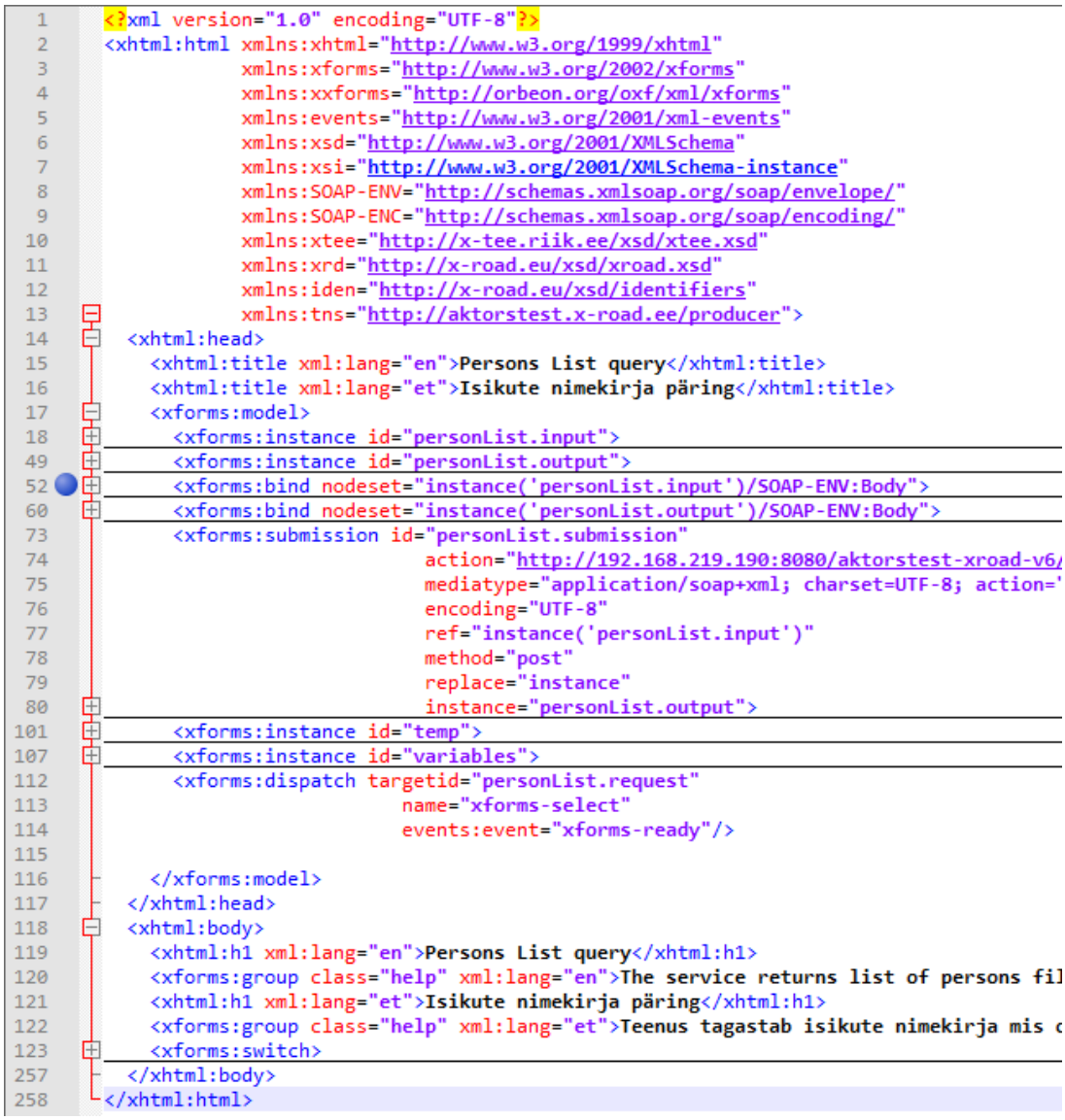


Figure 6 XForms description in a text editor

Brief overview of the file:

The file begins with an XML declaration.

- Then follows a `<xhtml:html>` tag. Next to that tag, all the prefixes of the namespaces used in the file are declared.

- `<xhtml:head>` section begins as usual with a `<xhtml:title>` tag that determines the title of the form. This has been generated based on the service `<xtee:title>` tag that is included in the WSDL file. The title is followed by `<xforms:model>` where three main things are defined: **Data structures**, (`<xforms:instance>`), **validation rules** (`<xforms:bind>`), **requests** (`<xforms:submission>`).

- `<xhtml:body>` section also begins with a title, which in this case is included inside a `<xhtml:h1>` tag. This is also generated based on `<xtee:title>`. Then follows a brief service

description inside a `<xhtml:h1>` tag. `<xforms:switch>` marks the beginning of the request forms.

Now, if we save the service form without changing it, we get a regular simple service.

In order to get a complex service, you have to copy the following elements from the XForms of the service *aktorstest.personDetails.v1*:

- `<xforms:instance>` (excluding **temp** with an **id**, because it is already there)
- `<xforms:bind>`
- `<xforms:submission>`
    - All of the aforementioned must be copied under the XForms' `<xforms:model>` of the service *aktorstest-complex.isikuandmed.v1*. The order is not important.
- `<xforms:case id="personDetails.response">`
- This goes under the XForms' `<xforms:switch>` of the service.

Hence, we get an XForms description that contains content elements necessary for activating the two services.

6.4 Link for switching to the other service

Now, a trigger (or, in other words, a button) has to be created under the `<xforms:repeat nodeset="person">` element, which could be used to activate the next service from the input of the first one. This has to be created under the element:
`<xforms:case id="personList.response">`.

Before that, you can change the `<xforms:repeat nodeset="person">` generated **id** value that will be hereafter used in the trigger description. In this example, it is **person_index** (`<xforms:repeat nodeset="person" id="person_index">`).

Trigger code:

```
<xforms:trigger appearance="minimal">
  <xforms:label xml:lang="et">Detailid</xforms:label>
  <xforms:label xml:lang="en">Details</xforms:label>
  <xforms:action events:event="DOMActivate">
    <xforms:setvalue ref="instance('personDetails.input')/SOAP-
ENV:Body/tns:personDetails/request/ssn" value="context()/ssn"/>
    <xforms:send submission="personDetails.submission"/>
  </xforms:action>
</xforms:trigger>
```

Trigger subparts:

- **appearance="minimal" –** determines that the trigger appears as a link, not a button

- **<xforms:label>** – determines the text shown in the user interface. Two texts have been defined: for Estonian (**xml:lang="et"**) and English (**xml:lang="en"**) language.

- **<xforms:setvalue>** – assigns personDetails service to the SOAP-message input (**personDetails.input**), a value to the **ssn** field from this row (meaning the filed that the user selects), from the "ssn" field (or, in other words, the value of the **value** attribute) with a button clicking event (event **DOMActivate**).

- **<xforms:send>** – sends a SOAP-request, using the xforms:submission

6.5 Changing the button logic

It would be good to go back from the response of the second service to the previous list. For this, we change
the "Back" button under `<xforms:case id="personDetails.response">` in a way that it would direct back to the previous `<xforms:case>`.

Then we change the trigger under the `<xforms:group class="actions">` element as follows:

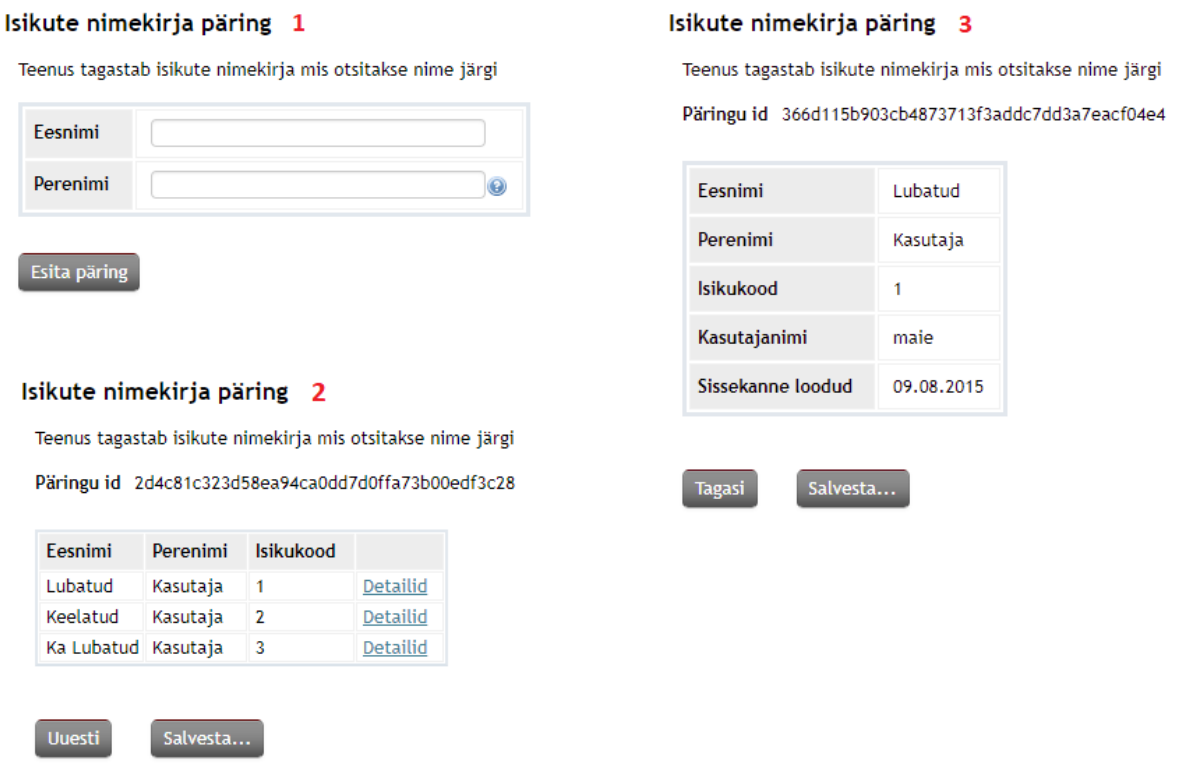
- For the `<xforms:toggle>` **case** attribute we set the value **personList.response**

- We change `<xforms:label>` texts respectively.

The result is the following code:

```
<xforms:group class="actions">
  <xforms:trigger>
    <xforms:label xml:lang="et">Tagasi</xforms:label>
    <xforms:label xml:lang="en">Back</xforms:label>
    <xforms:toggle events:event="DOMActivate" case="personList.response"/>
  </xforms:trigger>
</xforms:group>
```

6.6 Result



The result is a complex service which can be used to look at the persons list with names found (form 2) by entering a person's name (form 1) and moving on from there via a link named "Details" to form 3, where the details of the chosen person are shown.

Annex I complex service XForms

```
<?xml version="1.0" encoding="UTF-8"?>
<xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml"
            xmlns:xforms="http://www.w3.org/2002/xforms"
            xmlns:xxforms="http://orbeon.org/oxf/xml/xforms"
            xmlns:events="http://www.w3.org/2001/xml-events"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:xtee="http://x-tee.riik.ee/xsd/xtee.xsd"
            xmlns:xrd="http://x-road.eu/xsd/xroad.xsd"
            xmlns:iden="http://x-road.eu/xsd/identifiers"
            xmlns:tns="http://aktorstest.x-road.ee/producer">
    <xhtml:head>
        <xhtml:title xml:lang="en">Persons List query</xhtml:title>
        <xhtml:title xml:lang="et">Isikute nimekirja päring</xhtml:title>
        <xforms:model>
            <xforms:instance id="personList.input">
                <SOAP-ENV:Envelope>
                    <SOAP-ENV:Header>
                        <xrd:protocolVersion>4.0</xrd:protocolVersion>
                        <xrd:id/>
                        <xrd:userId>EE</xrd:userId>
                        <xrd:service iden:objectType="SERVICE">
                            <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                            <iden:memberClass>COM</iden:memberClass>
                            <iden:memberCode>11333578</iden:memberCode>
                            <iden:subsystemCode>aktorstest-db01</iden:subsystemCode>
                            <iden:serviceCode>personList</iden:serviceCode>
                            <iden:serviceVersion>v1</iden:serviceVersion>
                        </xrd:service>
                        <xrd:client iden:objectType="SUBSYSTEM">
                            <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                            <iden:memberClass/>
                            <iden:memberCode/>
                            <iden:subsystemCode/>
                        </xrd:client>
                    </SOAP-ENV:Header>
                    <SOAP-ENV:Body>
                        <ns5:personList xmlns:ns5="http://aktorstest.x-road.ee/producer">
                            <request>
                                <givenName/>
                                <surname/>
                            </request>
                        </ns5:personList>
                    </SOAP-ENV:Body>
                </SOAP-ENV:Envelope>
            </xforms:instance>
            <xforms:instance id="personList.output">
                <dummy/>
            </xforms:instance>
            <xforms:bind nodeset="instance('personList.input')/SOAP-ENV:Body">
                <xforms:bind nodeset="tns:personList">
                    <xforms:bind nodeset="request">
                        <xforms:bind nodeset="givenName" type="xforms:string"/>
                        <xforms:bind nodeset="surname" type="xforms:string"/>
                    </xforms:bind>
                </xforms:bind>
            </xforms:bind>
            <xforms:bind nodeset="instance('personList.output')/SOAP-ENV:Body">
                <xforms:bind nodeset="tns:personListResponse">
                    <xforms:bind nodeset="response">
                        <xforms:bind nodeset="faultCode" type="xforms:string"/>
                        <xforms:bind nodeset="faultString" type="xforms:string"/>
                        <xforms:bind nodeset="person">
                            <xforms:bind nodeset="givenName" type="xforms:string"/>
                            <xforms:bind nodeset="surname" type="xforms:string"/>
                            <xforms:bind nodeset="ssn" type="xforms:string"/>
                        </xforms:bind>
                    </xforms:bind>
                </xforms:bind>
            </xforms:bind>
            <xforms:submission id="personList.submission"
                               action="http://192.168.219.190:8080/aktorstest-xroad-v6/services/aktorstestServicePort"
                               mediatype="application/soap+xml; charset=UTF-8; action="
                               encoding="UTF-8"
                               ref="instance('personList.input')"
                               method="post"
                               replace="instance"
```

```
                        instance="personList.output">
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="false()"
                                 events:event="xforms-submit"/>
                <xforms:setvalue ref="instance('personList.input')/SOAP-ENV:Header/*:id"
                                 value="digest(string(random()), 'SHA-1', 'hex')"
                                 events:event="xforms-submit"/>
                <xforms:toggle case="personList.response" events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="true()"
                                 events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="true()"
                                 events:event="xforms-submit-error"/>
                <xforms:message level="modal" events:event="xforms-submit-error">
                    <xforms:output xml:lang="et"
                                   value="if (event('error-type') = 'submission-in-progress') then 'Üks päring juba käib!'
else if (event('error-type') = 'no-data') then 'Pole andmeid, mida saata!'        else if (event('error-type') =
'validation-error') then 'Valideerimise viga!'        else if (event('error-type') = 'parse-error') then 'Viga
vastuse töötlemisel!'        else if (event('error-type') = 'resource-error') then 'Päringu vastus ei ole XML!'
else if (event('error-type') = 'target-error') then 'Sihtkoha viga!'        else 'Sisemine viga!'"/>
                    <xforms:output xml:lang="en"
                                   value="if (event('error-type') = 'submission-in-progress') then 'Submission already
started!'        else if (event('error-type') = 'no-data') then 'No data to submit!'        else if (event('error-
type') = 'validation-error') then 'Validation error!'        else if (event('error-type') = 'parse-error') then
'Error parsing response!'        else if (event('error-type') = 'resource-error') then 'Response is not XML!'
else if (event('error-type') = 'target-error') then 'Target error!'        else 'Internal error!'"/>
                </xforms:message>
            </xforms:submission>
            <xforms:instance id="temp">
                <temp>
                    <relevant xsi:type="boolean">true</relevant>
                    <ssn/>
                </temp>
            </xforms:instance>
            <xforms:instance id="variables">
                <variables>
                    <ssn/>
                </variables>
            </xforms:instance>
            <xforms:dispatch targetid="personList.request"
                             name="xforms-select"
                             events:event="xforms-ready"/>


            <!-- personDetails service starts -->
            <xforms:instance id="personDetails.input">
                <SOAP-ENV:Envelope>
                    <SOAP-ENV:Header>
                        <xrd:protocolVersion>4.0</xrd:protocolVersion>
                        <xrd:id/>
                        <xrd:userId>EE</xrd:userId>
                        <xrd:service iden:objectType="SERVICE">
                            <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                            <iden:memberClass>COM</iden:memberClass>
                            <iden:memberCode>11333578</iden:memberCode>
                            <iden:subsystemCode>aktorstest-db01</iden:subsystemCode>
                            <iden:serviceCode>personDetails</iden:serviceCode>
                            <iden:serviceVersion>v1</iden:serviceVersion>
                        </xrd:service>
                        <xrd:client iden:objectType="SUBSYSTEM">
                            <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                            <iden:memberClass/>
                            <iden:memberCode/>
                            <iden:subsystemCode/>
                        </xrd:client>
                    </SOAP-ENV:Header>
                    <SOAP-ENV:Body>
                        <ns5:personDetails xmlns:ns5="http://aktorstest.x-road.ee/producer">
                            <request>
                                <ssn/>
                            </request>
                        </ns5:personDetails>
                    </SOAP-ENV:Body>
                </SOAP-ENV:Envelope>
            </xforms:instance>
            <xforms:instance id="personDetails.output">
                <dummy/>
            </xforms:instance>
            <xforms:bind nodeset="instance('personDetails.input')/SOAP-ENV:Body">
                <xforms:bind nodeset="tns:personDetails">
                    <xforms:bind nodeset="request">
                        <xforms:bind nodeset="ssn" type="xforms:string"/>
                    </xforms:bind>
                </xforms:bind>
```

```
            </xforms:bind>
            <xforms:bind nodeset="instance('personDetails.output')/SOAP-ENV:Body">
                <xforms:bind nodeset="tns:personDetailsResponse">
                    <xforms:bind nodeset="response">
                        <xforms:bind nodeset="faultCode" type="xforms:string"/>
                        <xforms:bind nodeset="faultString" type="xforms:string"/>
                        <xforms:bind nodeset="personDetailInfo">
                            <xforms:bind nodeset="givenName" type="xforms:string"/>
                            <xforms:bind nodeset="surname" type="xforms:string"/>
                            <xforms:bind nodeset="ssn" type="xforms:string"/>
                            <xforms:bind nodeset="username" type="xforms:string"/>
                            <xforms:bind nodeset="created" type="xforms:date"/>
                        </xforms:bind>
                    </xforms:bind>
                </xforms:bind>
            </xforms:bind>
            <xforms:submission id="personDetails.submission"
                               action="http://192.168.219.190:8080/aktorstest-xroad-v6/services/aktorstestServicePort"
                               mediatype="application/soap+xml; charset=UTF-8; action="
                               encoding="UTF-8"
                               ref="instance('personDetails.input')"
                               method="post"
                               replace="instance"
                               instance="personDetails.output">
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="false()"
                                 events:event="xforms-submit"/>
                <xforms:setvalue ref="instance('personDetails.input')/SOAP-ENV:Header/*:id"
                                 value="digest(string(random()), 'SHA-1', 'hex')"
                                 events:event="xforms-submit"/>
                <xforms:toggle case="personDetails.response" events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="true()"
                                 events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="true()"
                                 events:event="xforms-submit-error"/>
                <xforms:message level="modal" events:event="xforms-submit-error">
                    <xforms:output xml:lang="et"
                                   value="if (event('error-type') = 'submission-in-progress') then 'Üks päring juba käib!'
else if (event('error-type') = 'no-data') then 'Pole andmeid, mida saata!'        else if (event('error-type') =
'validation-error') then 'Valideerimise viga!'        else if (event('error-type') = 'parse-error') then 'Viga
vastuse töötlemisel!'        else if (event('error-type') = 'resource-error') then 'Päringu vastus ei ole XML!'
else if (event('error-type') = 'target-error') then 'Sihtkoha viga!'        else 'Sisemine viga!'"/>
                    <xforms:output xml:lang="en"
                                   value="if (event('error-type') = 'submission-in-progress') then 'Submission already
started!'        else if (event('error-type') = 'no-data') then 'No data to submit!'        else if (event('error-
type') = 'validation-error') then 'Validation error!'        else if (event('error-type') = 'parse-error') then
'Error parsing response!'        else if (event('error-type') = 'resource-error') then 'Response is not XML!'
else if (event('error-type') = 'target-error') then 'Target error!'        else 'Internal error!'"/>
                </xforms:message>
            </xforms:submission>
        </xforms:model>
    </xhtml:head>
    <xhtml:body>
        <xhtml:h1 xml:lang="en">Persons List query</xhtml:h1>
        <xforms:group class="help" xml:lang="en">The service returns list of persons filtered by name</xforms:group>
        <xhtml:h1 xml:lang="et">Isikute nimekirja päring</xhtml:h1>
        <xforms:group class="help" xml:lang="et">Teenus tagastab isikute nimekirja mis otsitakse nime
järgi</xforms:group>
        <xforms:switch>
            <xforms:case id="personList.request">
                <xforms:group ref="instance('personList.input')/SOAP-ENV:Body">
                    <xforms:group ref="tns:personList">
                        <xforms:group ref="request">
                            <xforms:input ref="givenName">
                                <xforms:label xml:lang="en">Given name</xforms:label>
                                <xforms:label xml:lang="et">Eesnimi</xforms:label>
                            </xforms:input>
                            <xforms:input ref="surname">
                                <xforms:label xml:lang="en">Surname</xforms:label>
                                <xforms:label xml:lang="et">Perenimi</xforms:label>
                                <xforms:help>Perekonnanimi v teine nimi</xforms:help>
                            </xforms:input>
                        </xforms:group>
                    </xforms:group>
                </xforms:group>
                <xforms:group class="actions">
                    <xforms:submit submission="personList.submission">
                        <xforms:label xml:lang="et">Esita päring</xforms:label>
                        <xforms:label xml:lang="en">Submit</xforms:label>
                    </xforms:submit>
                </xforms:group>
            </xforms:case>
            <xforms:case id="personList.response">
```

```
                <xforms:group ref="instance('personList.output')/SOAP-ENV:Header" class="serviceid">
                    <xforms:output ref="xrd:id">
                        <xforms:label xml:lang="et">Päringu id</xforms:label>
                        <xforms:label xml:lang="en">Query id</xforms:label>
                    </xforms:output>
                </xforms:group>
                <xforms:group ref="instance('personList.output')/SOAP-ENV:Body">
                    <xforms:group ref="tns:personListResponse">
                        <xforms:group ref="response">
                            <xforms:output ref="faultCode"/>
                            <xforms:output ref="faultString"/>
                            <xforms:repeat nodeset="person"
                                    id="person_index">
                                <xforms:output ref="givenName">
                                    <xforms:label xml:lang="en">Given name</xforms:label>
                                    <xforms:label xml:lang="et">Eesnimi</xforms:label>
                                </xforms:output>
                                <xforms:output ref="surname">
                                    <xforms:label xml:lang="en">Surname</xforms:label>
                                    <xforms:label xml:lang="et">Perenimi</xforms:label>
                                </xforms:output>
                                <xforms:output ref="ssn">
                                    <xforms:label xml:lang="en">SSN</xforms:label>
                                    <xforms:label xml:lang="et">Isikukood</xforms:label>
                                </xforms:output>
                                <xforms:trigger appearance="minimal">
                                    <xforms:label xml:lang="et">Detailid</xforms:label>
                                    <xforms:label xml:lang="en">Details</xforms:label>
                                    <xforms:action events:event="DOMActivate">
                                        <xforms:setvalue ref="instance('personDetails.input')/SOAP-
ENV:Body/tns:personDetails/request/ssn" value="context()/ssn"/>
                                        <xforms:send submission="personDetails.submission"/>
                                    </xforms:action>
                                </xforms:trigger>
                            </xforms:repeat>
                        </xforms:group>
                    </xforms:group>
                </xforms:group>
                <xforms:group ref="instance('personList.output')/SOAP-
ENV:Body/tns:personListResponse[not(response/*)]"
                        class="info">
                    <xhtml:span xml:lang="et">Andmeid ei tulnud.</xhtml:span>
                    <xhtml:span xml:lang="en">Service returned no data.</xhtml:span>
                </xforms:group>
                <xforms:group ref="instance('personList.output')/SOAP-ENV:Body/SOAP-ENV:Fault"
                        class="fault">
                    <xforms:output ref="faultstring"/>
                </xforms:group>
                <xforms:group class="actions">
                    <xforms:trigger>
                        <xforms:label xml:lang="et">Uuesti</xforms:label>
                        <xforms:label xml:lang="en">Again</xforms:label>
                        <xforms:toggle events:event="DOMActivate" case="personList.request"/>
                    </xforms:trigger>
                </xforms:group>
            </xforms:case>
            <xforms:case id="personDetails.response">
                <xforms:group ref="instance('personDetails.output')/SOAP-ENV:Header"
                        class="serviceid">
                    <xforms:output ref="xrd:id">
                        <xforms:label xml:lang="et">Päringu id</xforms:label>
                        <xforms:label xml:lang="en">Query id</xforms:label>
                    </xforms:output>
                </xforms:group>
                <xforms:group ref="instance('personDetails.output')/SOAP-ENV:Body">
                    <xforms:group ref="tns:personDetailsResponse">
                        <xforms:group ref="response">
                            <xforms:output ref="faultCode"/>
                            <xforms:output ref="faultString"/>
                            <xforms:group ref="personDetailInfo">
                                <xforms:output ref="givenName">
                                    <xforms:label xml:lang="en">Given name</xforms:label>
                                    <xforms:label xml:lang="et">Eesnimi</xforms:label>
                                </xforms:output>
                                <xforms:output ref="surname">
                                    <xforms:label xml:lang="en">Surname</xforms:label>
                                    <xforms:label xml:lang="et">Perenimi</xforms:label>
                                </xforms:output>
                                <xforms:output ref="ssn">
                                    <xforms:label xml:lang="en">Personal ID code</xforms:label>
                                    <xforms:label xml:lang="et">Isikukood</xforms:label>
                                </xforms:output>
                                <xforms:output ref="username">
                                    <xforms:label xml:lang="en">Username</xforms:label>
                                    <xforms:label xml:lang="et">Kasutajanimi</xforms:label>
```

```
                            </xforms:output>
                            <xforms:output ref="created">
                                <xforms:label xml:lang="en">Entry created</xforms:label>
                                <xforms:label xml:lang="et">Sissekanne loodud</xforms:label>
                            </xforms:output>
                        </xforms:group>
                    </xforms:group>
                </xforms:group>
            </xforms:group>
            <xforms:group ref="instance('personDetails.output')/SOAP-
ENV:Body/tns:personDetailsResponse[not(response/*)]"
                          class="info">
                <xhtml:span xml:lang="et">Andmeid ei tulnud.</xhtml:span>
                <xhtml:span xml:lang="en">Service returned no data.</xhtml:span>
            </xforms:group>
            <xforms:group ref="instance('personDetails.output')/SOAP-ENV:Body/SOAP-ENV:Fault"
                          class="fault">
                <xforms:output ref="faultstring"/>
            </xforms:group>
            <xforms:group class="actions">
                <xforms:trigger>
                    <xforms:label xml:lang="et">Tagasi</xforms:label>
                    <xforms:label xml:lang="en">Back</xforms:label>
                    <xforms:toggle events:event="DOMActivate" case="personList.response"/>
                </xforms:trigger>
            </xforms:group>
        </xforms:case>
    </xforms:switch>
</xhtml:body>
</xhtml:html>
```

Annex II Persons list query Xforms

```
<?xml version="1.0" encoding="UTF-8"?>
<xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml"
            xmlns:xforms="http://www.w3.org/2002/xforms"
            xmlns:xxforms="http://orbeon.org/oxf/xml/xforms"
            xmlns:events="http://www.w3.org/2001/xml-events"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:xtee="http://x-tee.riik.ee/xsd/xtee.xsd"
            xmlns:xrd="http://x-road.eu/xsd/xroad.xsd"
            xmlns:iden="http://x-road.eu/xsd/identifiers"
            xmlns:tns="http://aktorstest.x-road.ee/producer">
    <xhtml:head>
        <xhtml:title xml:lang="en">Persons List query</xhtml:title>
        <xhtml:title xml:lang="et">Isikute nimekirja päring</xhtml:title>
        <xforms:model>
            <xforms:instance id="personList.input">
                <SOAP-ENV:Envelope>
                    <SOAP-ENV:Header>
                        <xrd:protocolVersion>4.0</xrd:protocolVersion>
                        <xrd:id/>
                        <xrd:userId>EE</xrd:userId>
                        <xrd:service iden:objectType="SERVICE">
                            <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                            <iden:memberClass>COM</iden:memberClass>
                            <iden:memberCode>11333578</iden:memberCode>
                            <iden:subsystemCode>aktorstest-db01</iden:subsystemCode>
                            <iden:serviceCode>personList</iden:serviceCode>
                            <iden:serviceVersion>v1</iden:serviceVersion>
                        </xrd:service>
                        <xrd:client iden:objectType="SUBSYSTEM">
                            <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                            <iden:memberClass/>
                            <iden:memberCode/>
                            <iden:subsystemCode/>
                        </xrd:client>
                    </SOAP-ENV:Header>
                    <SOAP-ENV:Body>
                        <ns5:personList xmlns:ns5="http://aktorstest.x-road.ee/producer">
                            <request>
                                <givenName/>
                                <surname/>
                            </request>
                        </ns5:personList>
                    </SOAP-ENV:Body>
                </SOAP-ENV:Envelope>
            </xforms:instance>
            <xforms:instance id="personList.output">
                <dummy/>
            </xforms:instance>
            <xforms:bind nodeset="instance('personList.input')/SOAP-ENV:Body">
                <xforms:bind nodeset="tns:personList">
                    <xforms:bind nodeset="request">
                        <xforms:bind nodeset="givenName" type="xforms:string"/>
                        <xforms:bind nodeset="surname" type="xforms:string"/>
                    </xforms:bind>
                </xforms:bind>
            </xforms:bind>
            <xforms:bind nodeset="instance('personList.output')/SOAP-ENV:Body">
                <xforms:bind nodeset="tns:personListResponse">
                    <xforms:bind nodeset="response">
                        <xforms:bind nodeset="faultCode" type="xforms:string"/>
                        <xforms:bind nodeset="faultString" type="xforms:string"/>
                        <xforms:bind nodeset="person">
                            <xforms:bind nodeset="givenName" type="xforms:string"/>
                            <xforms:bind nodeset="surname" type="xforms:string"/>
                            <xforms:bind nodeset="ssn" type="xforms:string"/>
                        </xforms:bind>
                    </xforms:bind>
                </xforms:bind>
            </xforms:bind>
            <xforms:submission id="personList.submission"
                               action="http://192.168.219.190:8080/aktorstest-xroad-v6/services/aktorstestServicePort"
                               mediatype="application/soap+xml; charset=UTF-8; action="
                               encoding="UTF-8"
                               ref="instance('personList.input')"
                               method="post"
                               replace="instance"
```

```
                        instance="personList.output">
                <xforms:setvalue ref="instance('temp')/relevant"
                             value="false()"
                             events:event="xforms-submit"/>
                <xforms:setvalue ref="instance('personList.input')/SOAP-ENV:Header/*:id"
                             value="digest(string(random()), 'SHA-1', 'hex')"
                             events:event="xforms-submit"/>
                <xforms:toggle case="personList.response" events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                             value="true()"
                             events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                             value="true()"
                             events:event="xforms-submit-error"/>
                <xforms:message level="modal" events:event="xforms-submit-error">
                    <xforms:output xml:lang="et"
                              value="if (event('error-type') = 'submission-in-progress') then 'Üks päring juba käib!'
else if (event('error-type') = 'no-data') then 'Pole andmeid, mida saata!'        else if (event('error-type') =
'validation-error') then 'Valideerimise viga!'        else if (event('error-type') = 'parse-error') then 'Viga
vastuse töötlemisel!'        else if (event('error-type') = 'resource-error') then 'Päringu vastus ei ole XML!'
else if (event('error-type') = 'target-error') then 'Sihtkoha viga!'        else 'Sisemine viga!'"/>
                    <xforms:output xml:lang="en"
                              value="if (event('error-type') = 'submission-in-progress') then 'Submission already
started!'        else if (event('error-type') = 'no-data') then 'No data to submit!'        else if (event('error-
type') = 'validation-error') then 'Validation error!'        else if (event('error-type') = 'parse-error') then
'Error parsing response!'        else if (event('error-type') = 'resource-error') then 'Response is not XML!'
else if (event('error-type') = 'target-error') then 'Target error!'        else 'Internal error!'"/>
                </xforms:message>
            </xforms:submission>
            <xforms:instance id="temp">
                <temp>
                    <relevant xsi:type="boolean">true</relevant>
                </temp>
            </xforms:instance>
            <xforms:dispatch targetid="personList.request"
                         name="xforms-select"
                         events:event="xforms-ready"/>
        </xforms:model>
    </xhtml:head>
    <xhtml:body>
        <xhtml:h1 xml:lang="en">Persons List query</xhtml:h1>
        <xforms:group class="help" xml:lang="en">The service returns list of persons filtered by name</xforms:group>
        <xhtml:h1 xml:lang="et">Isikute nimekirja päring</xhtml:h1>
        <xforms:group class="help" xml:lang="et">Teenus tagastab isikute nimekirja mis otsitakse nime
järgi</xforms:group>
        <xforms:switch>
            <xforms:case id="personList.request">
                <xforms:group ref="instance('personList.input')/SOAP-ENV:Body">
                    <xforms:group ref="tns:personList">
                        <xforms:group ref="request">
                            <xforms:input ref="givenName">
                                <xforms:label xml:lang="en">Given name</xforms:label>
                                <xforms:label xml:lang="et">Eesnimi</xforms:label>
                            </xforms:input>
                            <xforms:input ref="surname">
                                <xforms:label xml:lang="en">Surname</xforms:label>
                                <xforms:label xml:lang="et">Perenimi</xforms:label>
                            </xforms:input>
                        </xforms:group>
                    </xforms:group>
                </xforms:group>
                <xforms:group class="actions">
                    <xforms:submit submission="personList.submission">
                        <xforms:label xml:lang="et">Esita päring</xforms:label>
                        <xforms:label xml:lang="en">Submit</xforms:label>
                    </xforms:submit>
                </xforms:group>
            </xforms:case>
            <xforms:case id="personList.response">
                <xforms:group ref="instance('personList.output')/SOAP-ENV:Header" class="serviceid">
                    <xforms:output ref="xrd:id">
                        <xforms:label xml:lang="et">Päringu id</xforms:label>
                        <xforms:label xml:lang="en">Query id</xforms:label>
                    </xforms:output>
                </xforms:group>
                <xforms:group ref="instance('personList.output')/SOAP-ENV:Body">
                    <xforms:group ref="tns:personListResponse">
                        <xforms:group ref="response">
                            <xforms:output ref="faultCode"/>
                            <xforms:output ref="faultString"/>
                            <xforms:repeat nodeset="person"
                                    id="personList_output_tns_personListResponse_response_person">
                                <xforms:output ref="givenName">
                                    <xforms:label xml:lang="en">Given name</xforms:label>
                                    <xforms:label xml:lang="et">Eesnimi</xforms:label>
```

```
                            </xforms:output>
                            <xforms:output ref="surname">
                                <xforms:label xml:lang="en">Surname</xforms:label>
                                <xforms:label xml:lang="et">Perenimi</xforms:label>
                            </xforms:output>
                            <xforms:output ref="ssn">
                                <xforms:label xml:lang="en">SSN</xforms:label>
                                <xforms:label xml:lang="et">Isikukood</xforms:label>
                            </xforms:output>
                        </xforms:repeat>
                    </xforms:group>
                </xforms:group>
                <xforms:group ref="instance('personList.output')/SOAP-ENV:Body/tns:personListResponse[if (keha) then
not(keha/*) else if (response) then not(response/*) else not(*)]"
                              class="info">
                    <xhtml:span xml:lang="et">Andmeid ei tulnud.</xhtml:span>
                    <xhtml:span xml:lang="en">Service returned no data.</xhtml:span>
                </xforms:group>
                <xforms:group ref="instance('personList.output')/SOAP-ENV:Body/SOAP-ENV:Fault"
                              class="fault">
                    <xforms:output ref="faultstring"/>
                </xforms:group>
                <xforms:group class="actions">
                    <xforms:trigger>
                        <xforms:label xml:lang="et">Uuesti</xforms:label>
                        <xforms:label xml:lang="en">Again</xforms:label>
                        <xforms:toggle events:event="DOMActivate" case="personList.request"/>
                    </xforms:trigger>
                </xforms:group>
            </xforms:case>
        </xforms:switch>
    </xhtml:body>
</xhtml:html>
```

Annex III Person details query XForms

```
<?xml version="1.0" encoding="UTF-8"?>
<xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml"
            xmlns:xforms="http://www.w3.org/2002/xforms"
            xmlns:xxforms="http://orbeon.org/oxf/xml/xforms"
            xmlns:events="http://www.w3.org/2001/xml-events"
            xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
            xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
            xmlns:xtee="http://x-tee.riik.ee/xsd/xtee.xsd"
            xmlns:xrd="http://x-road.eu/xsd/xroad.xsd"
            xmlns:iden="http://x-road.eu/xsd/identifiers"
            xmlns:tns="http://aktorstest.x-road.ee/producer">
    <xhtml:head>
        <xhtml:title xml:lang="en">Person details query</xhtml:title>
        <xhtml:title xml:lang="et">Isiku detailandmete päring</xhtml:title>
        <xforms:model>
            <xforms:instance id="personDetails.input">
                <SOAP-ENV:Envelope>
                    <SOAP-ENV:Header>
                        <xrd:protocolVersion>4.0</xrd:protocolVersion>
                        <xrd:id/>
                        <xrd:userId>EE</xrd:userId>
                        <xrd:service iden:objectType="SERVICE">
                            <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                            <iden:memberClass>COM</iden:memberClass>
                            <iden:memberCode>11333578</iden:memberCode>
                            <iden:subsystemCode>aktorstest-db01</iden:subsystemCode>
                            <iden:serviceCode>personDetails</iden:serviceCode>
                            <iden:serviceVersion>v1</iden:serviceVersion>
                        </xrd:service>
                        <xrd:client iden:objectType="SUBSYSTEM">
                            <iden:xRoadInstance>ee-dev</iden:xRoadInstance>
                            <iden:memberClass/>
                            <iden:memberCode/>
                            <iden:subsystemCode/>
                        </xrd:client>
                    </SOAP-ENV:Header>
                    <SOAP-ENV:Body>
                        <ns5:personDetails xmlns:ns5="http://aktorstest.x-road.ee/producer">
                            <request>
                                <ssn/>
                            </request>
                        </ns5:personDetails>
                    </SOAP-ENV:Body>
                </SOAP-ENV:Envelope>
            </xforms:instance>
            <xforms:instance id="personDetails.output">
                <dummy/>
            </xforms:instance>
            <xforms:bind nodeset="instance('personDetails.input')/SOAP-ENV:Body">
                <xforms:bind nodeset="tns:personDetails">
                    <xforms:bind nodeset="request">
                        <xforms:bind nodeset="ssn" type="xforms:string"/>
                    </xforms:bind>
                </xforms:bind>
            </xforms:bind>
            <xforms:bind nodeset="instance('personDetails.output')/SOAP-ENV:Body">
                <xforms:bind nodeset="tns:personDetailsResponse">
                    <xforms:bind nodeset="response">
                        <xforms:bind nodeset="faultCode" type="xforms:string"/>
                        <xforms:bind nodeset="faultString" type="xforms:string"/>
                        <xforms:bind nodeset="personDetailInfo">
                            <xforms:bind nodeset="givenName" type="xforms:string"/>
                            <xforms:bind nodeset="surname" type="xforms:string"/>
                            <xforms:bind nodeset="ssn" type="xforms:string"/>
                            <xforms:bind nodeset="username" type="xforms:string"/>
                            <xforms:bind nodeset="created" type="xforms:date"/>
                        </xforms:bind>
                    </xforms:bind>
                </xforms:bind>
            </xforms:bind>
            <xforms:submission id="personDetails.submission"
                               action="http://192.168.219.190:8080/aktorstest-xroad-v6/services/aktorstestServicePort"
                               mediatype="application/soap+xml; charset=UTF-8; action="
                               encoding="UTF-8"
                               ref="instance('personDetails.input')"
                               method="post"
                               replace="instance"
```

```
                        instance="personDetails.output">
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="false()"
                                 events:event="xforms-submit"/>
                <xforms:setvalue ref="instance('personDetails.input')/SOAP-ENV:Header/*:id"
                                 value="digest(string(random()), 'SHA-1', 'hex')"
                                 events:event="xforms-submit"/>
                <xforms:toggle case="personDetails.response" events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="true()"
                                 events:event="xforms-submit-done"/>
                <xforms:setvalue ref="instance('temp')/relevant"
                                 value="true()"
                                 events:event="xforms-submit-error"/>
                <xforms:message level="modal" events:event="xforms-submit-error">
                    <xforms:output xml:lang="et"
                                   value="if (event('error-type') = 'submission-in-progress') then 'Üks päring juba käib!'
else if (event('error-type') = 'no-data') then 'Pole andmeid, mida saata!'        else if (event('error-type') =
'validation-error') then 'Valideerimise viga!'        else if (event('error-type') = 'parse-error') then 'Viga
vastuse töötlemisel!'        else if (event('error-type') = 'resource-error') then 'Päringu vastus ei ole XML!'
else if (event('error-type') = 'target-error') then 'Sihtkoha viga!'        else 'Sisemine viga!'"/>
                    <xforms:output xml:lang="en"
                                   value="if (event('error-type') = 'submission-in-progress') then 'Submission already
started!'        else if (event('error-type') = 'no-data') then 'No data to submit!'        else if (event('error-
type') = 'validation-error') then 'Validation error!'        else if (event('error-type') = 'parse-error') then
'Error parsing response!'        else if (event('error-type') = 'resource-error') then 'Response is not XML!'
else if (event('error-type') = 'target-error') then 'Target error!'        else 'Internal error!'"/>
                </xforms:message>
            </xforms:submission>
            <xforms:instance id="temp">
                <temp>
                    <relevant xsi:type="boolean">true</relevant>
                </temp>
            </xforms:instance>
            <xforms:dispatch targetid="personDetails.request"
                             name="xforms-select"
                             events:event="xforms-ready"/>
        </xforms:model>
    </xhtml:head>
    <xhtml:body>
        <xhtml:h1 xml:lang="en">Person details query</xhtml:h1>
        <xforms:group class="help" xml:lang="en">The service returns detailed information for zero or one users who's
SSN (Social Security Number) matches exactly to the requested SSN.</xforms:group>
        <xhtml:h1 xml:lang="et">Isiku detailandmete päring</xhtml:h1>
        <xforms:group class="help" xml:lang="et">Teenus tagastab maksimaalselt ühe kasutaaja detailandmed, kelle
isikukood langeb kokku päringus antud isikukoodiga.</xforms:group>
        <xforms:switch>
            <xforms:case id="personDetails.request">
                <xforms:group ref="instance('personDetails.input')/SOAP-ENV:Body">
                    <xforms:group ref="tns:personDetails">
                        <xforms:group ref="request">
                            <xforms:input ref="ssn">
                                <xforms:label xml:lang="en">SSN</xforms:label>
                                <xforms:label xml:lang="et">Isikukood</xforms:label>
                            </xforms:input>
                        </xforms:group>
                    </xforms:group>
                </xforms:group>
                <xforms:group class="actions">
                    <xforms:submit submission="personDetails.submission">
                        <xforms:label xml:lang="et">Esita päring</xforms:label>
                        <xforms:label xml:lang="en">Submit</xforms:label>
                    </xforms:submit>
                </xforms:group>
            </xforms:case>
            <xforms:case id="personDetails.response">
                <xforms:group ref="instance('personDetails.output')/SOAP-ENV:Header"
                              class="serviceid">
                    <xforms:output ref="xrd:id">
                        <xforms:label xml:lang="et">Päringu id</xforms:label>
                        <xforms:label xml:lang="en">Query id</xforms:label>
                    </xforms:output>
                </xforms:group>
                <xforms:group ref="instance('personDetails.output')/SOAP-ENV:Body">
                    <xforms:group ref="tns:personDetailsResponse">
                        <xforms:group ref="response">
                            <xforms:output ref="faultCode"/>
                            <xforms:output ref="faultString"/>
                            <xforms:group ref="personDetailInfo">
                                <xforms:output ref="givenName">
                                    <xforms:label xml:lang="en">Given name</xforms:label>
                                    <xforms:label xml:lang="et">Eesnimi</xforms:label>
                                </xforms:output>
                                <xforms:output ref="surname">
                                    <xforms:label xml:lang="en">Surname</xforms:label>
```

```
                                <xforms:label xml:lang="et">Perenimi</xforms:label>
                            </xforms:output>
                            <xforms:output ref="ssn">
                                <xforms:label xml:lang="en">Personal ID code</xforms:label>
                                <xforms:label xml:lang="et">Isikukood</xforms:label>
                            </xforms:output>
                            <xforms:output ref="username">
                                <xforms:label xml:lang="en">Username</xforms:label>
                                <xforms:label xml:lang="et">Kasutajanimi</xforms:label>
                            </xforms:output>
                            <xforms:output ref="created">
                                <xforms:label xml:lang="en">Entry created</xforms:label>
                                <xforms:label xml:lang="et">Sissekanne loodud</xforms:label>
                            </xforms:output>
                        </xforms:group>
                    </xforms:group>
                </xforms:group>
            </xforms:group>
            <xforms:group ref="instance('personDetails.output')/SOAP-ENV:Body/tns:personDetailsResponse[if (keha)
then not(keha/*) else if (response) then not(response/*) else not(*)]"
                          class="info">
                <xhtml:span xml:lang="et">Andmeid ei tulnud.</xhtml:span>
                <xhtml:span xml:lang="en">Service returned no data.</xhtml:span>
            </xforms:group>
            <xforms:group ref="instance('personDetails.output')/SOAP-ENV:Body/SOAP-ENV:Fault"
                          class="fault">
                <xforms:output ref="faultstring"/>
            </xforms:group>
            <xforms:group class="actions">
                <xforms:trigger>
                    <xforms:label xml:lang="et">Uuesti</xforms:label>
                    <xforms:label xml:lang="en">Again</xforms:label>
                    <xforms:toggle events:event="DOMActivate" case="personDetails.request"/>
                </xforms:trigger>
            </xforms:group>
        </xforms:case>
    </xforms:switch>
</xhtml:body>
</xhtml:html>
```