Viewing Data with XForms

Contents

Data

I have a dataset of performances by a University choir stretching over 65 years or more. It is just a long list of concerts:

```
<concerts>
  <concert>...</concert>
  <concert>...</concert>
  <concert>...</concert>
   ...
</concerts>
```

A typical concert entry looks like this:

```
<concert>
   <year>1970</year>
   <month>5</month>
   <program>Duruflé – Requiem</program>
   <program>Mozart – Krönungsmesse</program>
   <where>Augustinuskerk te Amsterdam</where>
   <where>De Doelen te Rotterdam</where>
   <with>VU-orkest</with>
   <event>LP-opname: Mozart</event>
</concert>
```

I would like to be able to browse this data, but also easily answer questions like "How often have they performed something by Stravinsky?", "How often have they performed in the Concertgebouw", "What did they perform in 1960?", and so on.

Browsing

First to browse. We load the data:

```
<instance src="concerts.xml"/>
```

and then display it, which we'll do as a sort of table, one row per concert:

```
<group>
   <label>Concerts</label>
   <repeat ref="concert">
      ...
   </repeat>
</group>
```

Per concert

For each concert, each group of entries (*date*, *program*, *where*, *with*, and *event*) will be displayed as a column by making the CSS `display` property of each group `inline-block`, so that the groups are displayed next to each other:

```
<group class="concert">
   <output class="when" value="concat(year, '-', month)"/>
   <group class="program">
      <repeat ref="program"><output class="line" ref="."/></repeat>
   </group>
   <group class="where">
      <repeat ref="where"><output class="line" ref="."/></repeat>
   </group>
   <group class="with">
      <repeat ref="with"><output class="line" ref="."/></repeat>
   </group>
   <group class="event">
      <repeat ref="event"><output class="line" ref="."/></repeat>
   </group>
</group>
```

Note that this doesn't require the sub-elements of the concerts to be in this order, or even adjacent; it just selects all sub-elements called `program` (for example) within a concert element, and displays them together.

Headings

Adding a row of titles above this using the same CSS class for the header titles ensures that they line up:

```
<group class="header">
   <output class="when" value="'when'"/>
   <output class="program" value="'what'"/>
   <output class="where" value="'where'"/>
   <output class="with" value="'with'"/>
   <output class="event" value="'why'"/>
</group>
```

Result

With some suitable CSS, we get this:

Concerts

when	**what**	**where**	**with**	**why**
1955-10	Hassler − Cantate Domino	Westerkerk te Amsterdam		75e Dies Natalis VU
1956-onb	Brahms − Alt-Rhapsodie	Onbekend		
1956-10	Onbekend	Amsterdam Rotterdam		
1957-2	Onbekend	Onbekend		
1957-5	Buxtehude − Cantate Distler − Christ, der Du bist der helle Tag	Raphaëlpleinkerk te Amsterdam		
1957-10	Onbekend	Concertgebouw, kleine zaal te Amsterdam		t.g.v. 77e Dies Natalis Studentencorps VU
1957-11	Buxtehude − Alles, was ihr tut mit Worten oder Werken J. S. Bach − Gott soll allein mein Herze haben Mozart − Ave Verum Corpus Distler − Christ, der Du bist der helle Tag Distler − Lobe den Herren Mendelssohn-Bartholdy −	Marcanti te Amsterdam	Bachorkest	

Source

Fancy heading

We may as well fancy up the heading a little bit, and replace:

```
<label class="header">Concerts</label>
```

with

```
<label class="header">Concerts,
                      <output value="min(concert/year)"/> - <output value="m
</label>
```

Answering Questions

We'll just do a search-machine-like search on the data.

We create an instance for the search string:

```
<instance id="search">
   <data xmlns=""><q/></data>
</instance>
```

and an input control for it:

```
<input incremental="true" ref="instance('search')/q">
   <label>Search</label>
</input>
```

That's XForms 1.1. The newer XForms 2 allows you to say:

```
<input incremental="true" ref="instance('search')/q" label="Search"/>
```

Filtering

Whenever you have a sequence of items, such as with `ref="concert"` above, you can select a subset of them using a filter: `ref="concert[`*`condition`*`]"`, selecting only those concerts that match the condition.

If we want only the concerts from 1975, we can write:

```
concert[year=1975]
```

If we want only the concerts that contain a piece composed by Bach, we write:

```
concert[contains(piece, 'Bach')]
```

If we want the concerts where *any* field contains "Amsterdam", we write

```
concert[contains(*, 'Amsterdam')]
```

In fact we can even say:

```
concert[contains(., 'Amsterdam')]
```

which means "any concert that contains the string "Amsterdam" anywhere (the "." means "self").

Matching the search string

Finally if we want the concerts that contain the search string, we write

```
concert[contains(., instance('search')/q)]
```

So we replace:

```
<repeat ref="concert">
```

with that:

```
<repeat ref="concert[contains(., instance('search')/q)]]">
```

So this says "repeat over the concerts that contain the search string".

Result

Concerts, 1955 - 2022
Search

when	what	where	with	why
1955-10	Hassler − Cantate Domino	Westerkerk te Amsterdam		75e Dies Natalis VU
1956-onb	Brahms − Alt-Rhapsodie	Onbekend		
1956-10	Onbekend	Amsterdam Rotterdam		
1957-2	Onbekend	Onbekend		
1957-5	Buxtehude − Cantate Distler − Christ, der Du bist der helle Tag	Raphaëlpleinkerk te Amsterdam		
1957-10	Onbekend	Concertgebouw, kleine zaal te Amsterdam		t.g.v. 77e Dies Natalis Studentencorps VU
1957-11	Buxtehude − Alles, was ihr tut mit Worten oder Werken J. S. Bach − Gott soll allein mein Herze haben Mozart − Ave Verum Corpus Distler − Christ, der Du bist der helle Tag	Marcanti te Amsterdam	Bachorkest	

Source

Making the search case-insensitive

The function `lower-case` returns the lower-case version of its parameter, so that if `q` is `Mozart`, then `lower-case(q)` is `mozart`. (The `lower-case` function is from XForms 2. Previous versions use `translate(q, 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz')` to achieve similar effect).

So in the repeat over the concerts, we replace

```
contains(., instance('search')/q)
```

with

```
contains(lower-case(.), lower-case(instance('search')/q))
```

which checks if a lower-case version of the element content contains the lower-case version of the search string.

Result

Here it is:

Concerts, 1955 - 2022
Search (case-insensitive)

when why	what	where	with
1955-10 75e Dies Natalis VU	Hassler − Cantate Domino	Westerkerk te Amsterdam	
1956-onb	Brahms − Alt-Rhapsodie	Onbekend	
1956-10	Onbekend	Amsterdam Rotterdam	
1957-2	Onbekend	Onbekend	
1957-5	Buxtehude − Cantate Distler − Christ, der Du bist der helle Tag	Raphaëlpleinkerk te Amsterdam	
1957-10 t.g.v. 77e Dies Natalis Studentencorps VU	Onbekend	Concertgebouw, kleine zaal te Amsterdam	
1957-11	Buxtehude − Alles, was ihr tut mit Worten oder Werken J. S. Bach − Gott soll allein mein Herze haben Mozart − Ave Verum Corpus	Marcanti te Amsterdam	Bachorkest

There you have it. A useful application; about 35 lines of XForms. Source