

REL STANDARD TAG LIBRARY & ODATA QUERY EXTENSION

Mihai Cădăriu

yatb.mitza.net

About Mihai



Principal Consultant, SDL WCM, San Jose, USA

Joined Tridion in 2005, Amsterdam, NL

Previously Java developer

Yet Another Tridion Blog (yatb.mitza.net)

Agenda

- OData Query Extension
 - ▣ Extending OData?
 - ▣ Why a Query Extension?
 - ▣ It's all about the parser...
 - ▣ Demo, demo, demo
 - ▣ Resources
- REL Standard Tag Library
 - ▣ JSTL only with REL
 - ▣ The Tags
 - ▣ It's all about the parsers...
 - ▣ Demo
 - ▣ Resources

OData Query Extension

Extending OData?

- Yes, definitely possible... totally unsupported 😊
- Implementing the CD WebService framework
- Tridion 2011
 - ▣ `com.tridion.webservices.extension.ODataEntryRegistrationService`
- Tridion 2013
 - ▣ `com.tridion.webservices.extension.ODataResourceBundleRegistrationService`
 - ▣ Tridion 2013
- Self-registering... Woo-hoo!!!
 - ▣ Just drop JAR in **/web-inf/lib**

Why a Query Extension?

- ▣ CD Web Service – limited functionality
- ▣ CD API – full functionality
- ▣ OData Query Extension uses CD API
 - Specially crafted URL allows queries for content
 - URL syntax almost OData standard
 - Parser breaks down query into CD API calls
 - Make full usage of CD API (no hacks)
 - Uses storage cache, Cache Channel Service
 - Parser cached expressions

What Does it Do?

- Query Components
 - ▣ Generates response with **Component** entities
 - ▣ <http://server/odata/odata.svc/QueryComponents>
- Query Component Presentations
 - ▣ Generates response with **ComponentPresentation**
 - ▣ <http://server/odata/odata.svc/QueryComponentPresentations>
- Query Pages
 - ▣ Generates response with **Page** entities
 - ▣ <http://server/odata/odata.svc/Pages>

Querying Criterias

- ❑ BinaryType
- ❑ ItemCreationDate
- ❑ ItemInitialPublishDate
- ❑ ItemLastPublishedDate
- ❑ ItemModificationDate
- ❑ ItemReference
- ❑ ItemSchema
- ❑ ItemTitle
- ❑ Multimedia
- ❑ PageTemplate
- ❑ PageURL
- ❑ Publication
- ❑ SchemaTitle
- ❑ PublicationKey
- ❑ PublicationMultimediaPath
- ❑ PublicationMultimediaURL
- ❑ PublicationPath
- ❑ PublicationTitle
- ❑ PublicationURL
- ❑ StructureGroup
- ❑ StructureGroupDirectory
- ❑ StructureGroupTitle
- ❑ Taxonomy
- ❑ CustomMetaKey
- ❑ CustomMetaValue

Querying Criterias (2)

□ Using OData resource identifiers

`/odata.svc/Collection(CriteriaName=CriteriaValue)`

Example:

`/odata.svc/QueryComponents(Publication=3,ItemTitle=My Component Name)`

□ Using \$filter

`/odata.svc/Collection?$filter=CriteriaName field_operator CriteriaValue`

Example:

`$filter=Multimedia eq true and (Publication eq 3 or Publication eq 4)`

`$filter=CustomMetaValue like 'Artic%'`

`$filter=ItemModificationDate gt '2013-05-01 13:24:56.999'`

Complex Queries – Functions

- CustomMetaKey
- CustomMetaValue
- CustomMetaDateRange
- CustomMetaStringRange
- NumericalRange
- StructureGroup
- StructureGroupDirectory
- StructureGroupTitle
- TaxonomyKeyword
- TaxonomyKeywordDescription
- TaxonomyKeywordKey
- TaxonomyKeywordName

Complex Queries – Functions (2)

- **CustomMetaKey(keyName[, field_operator])**
- **CustomMetaKey('doc%', like)**

- **CustomMetaValue(CustomMetaKey(keyName), value[, field_operator])**
- **CustomMetaValue(CustomMetaKey('color'), 'red')**

- **Field operators:**
 - eq, neq, gt, ge, lt, le, like

Sorting Results

□ Using **\$orderby**

`/odata.svc/Collection?$orderby=Column [asc | desc] [CustomMetaKey
[string | float | date] [asc | desc]]`

Examples:

`$orderby=ItemLastPublish desc`

`$orderby=ItemTitle ItemLastPublish desc`

`$orderby=GoLiveDate date desc`

Pagination

- Using **\$top** and **\$skip**
 - specify **\$top**, but not **\$skip**:
 - a LimitFilter returns only the top specified results
 - specify **\$skip**, but not **\$top** :
 - a PagingFilter returns all results after skipping a few
 - specify both **\$top** and **\$skip**:
 - a PagingFilter returns only the top results after skipping a few

Parser

- Custom LL1 recursive parser
- Breaks down query string (**\$filter**)
- Caches parsed expressions

Resources

- On yatb.mitza.net
- <http://yatb.mitza.net/2013/06/extending-odata-query-functionality.html>
- <http://yatb.mitza.net/2013/06/url-syntax-for-odata-query-extension.html>
- <http://yatb.mitza.net/2013/06/function-support-for-queries-in-odata.html>

- On Google Code
- <https://code.google.com/p/yet-another-tridion-blog/source/browse/trunk/OData+Query+Extension/>
- <https://code.google.com/p/yet-another-tridion-blog/source/browse/branches/Tridion+2011+SP1+HR1/OData+Query+Extension/> - Tridion 2011 branch



REL Standard Tag Library

Just like JSTL... only with REL

- Java Standard Tag Library
- RSTL is the Tridion equivalent
- Tags for performing several common tasks
 - ▣ Iteration, loops
 - ▣ Conditionals (if, switch)
 - ▣ Formatting (XML/XSLT transforms, date time)

The Tags

□ `<c:choose>` / `<c:when>` / `<c:otherwise>`

`<c:choose>`

`<c:when test="expr1">`

Do something

`</c:when>`

`<c:when test="expr2">`

Do something else

`</c:when>`

`<c:otherwise>`

Do something otherwise

`</c:otherwise>`

`</c:choose>`

ForEach

□ `<c:forEach>`

```
<c:forEach var="expr" items="expr">
```

Do something

```
</c:forEach>
```

```
<c:forEach var="cpMeta"
```

```
  items="pageMeta.componentPresentationMetas">
```

```
  <c:out value="cpMeta.componentId"/>
```

```
</c:forEach>
```

If

□ `<c:if>`

```
<c:if test="expr">
```

```
    Do something
```

```
</c:if>
```

```
<c:if test="userName == 'Mihai'">
```

```
    Hello, Mihai!
```

```
</c:if>
```

Import resource

□ `<c:import>`

```
<c:import var="expr" url="expr" componentTemplate="expr">  
  <c:param name="expr" value="expr"/>  
</c:import>
```

```
<c:import var="xml" url="tcm:1-23-64">  
  <c:param name="color" value="{myColor}"/>  
</c:import>
```

Output a Value

□ `<c:out>`

`<c:out value="expr"/>`

`<c:out value="cp.componentId"/>`

Set/Remove a Variable

□ `<c:set>` / `<c:remove>`

```
<c:set var="expr" value="expr"/>
```

```
<c:remove var="expr"/>
```

```
<c:set var="expr">
```

Do something

```
</c:set>
```

```
<c:remove var="cpList"/>
```

Format Date

□ `<fmt:formatDate>` / `<fmt:parseDate>`

```
<fmt:formatDate var="expr" value="expr" pattern="expr"/>
```

```
<fmt:parseDate var="expr" value="expr" pattern="expr"/>
```

```
<fmt:formatDate value="myDate" pattern="dd.MM.yyyy HH:mm"/>
```

```
<fmt:parseDate var="myDate" value="2013-07-23"  
    pattern="yyyy-MM-dd"/>
```


XML/XSLT Transform

□ `<x:transform>`

```
<x:transform var="expr" doc="expr" xslt="expr">
```

```
  <x:param var="expr" select="expr" />
```

```
</x:transform>
```

```
<x:transform doc="nav-xml" xslt="top-xslt">
```

```
  <x:param var="color" select="{bg-color}"/>
```

```
</x:transform>
```

Demo



Resources

- On yatb.mitza.net
- <http://yatb.mitza.net/2013/07/rel-standard-tag-library-expression.html>
- <http://yatb.mitza.net/2013/07/rel-standard-tag-library.html>
- <http://yatb.mitza.net/2013/07/rel-standard-tag-library-installation.html>

- On Google Code
- <https://code.google.com/p/yet-another-tridion-blog/source/browse/trunk/REL+Standard+Tag+Library/>

Questions & Answers



That's all, folks!

