Schema documentation for service-eca-1.3.xsd
20 october 2013

Table of Contents

Namespace: "" ................................................................. 1
Schemas .................................................................................. 1
Elements .................................................................................. 1
Element secas ............................................................................. 1
Element seca ............................................................................. 1

Namespace: ""

Schemas

Main schema service-eca-1.3.xsd

Namespace

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No namespace</td>
<td></td>
</tr>
</tbody>
</table>

Elements

Element secas

Namespace

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No namespace</td>
<td></td>
</tr>
</tbody>
</table>

Diagram

```
<secas>
  <seca run-on-error="false" service="" when="">[0,unbounded]</seca>
</secas>
```

Instance

```
<secas>
  <seca run-on-error="false" service="" when="">[0,unbounded]</seca>
</secas>
```

Element seca

Namespace

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No namespace</td>
<td></td>
</tr>
</tbody>
</table>

Annotations

Triggered by service calls with various options about when during the service call to trigger the event.
If condition (optional) evaluates to true then the actions are run.
Service ECAs are meant for triggering business processes and for extending the functionality of existing services that you don't want to, or can't, modify.
Service ECAs should NOT generally be used for maintenance of data derived from other entities, Entity ECA rules are a much better tool for that.

When this runs the context will be whatever context the service was run in, plus the individual parameters for convenience in reading the values.
If when is before the service itself is run there will be a context variable called "parameters" with the parameters Map in it that you can modify as needed in the actions. If when is after the service itself the "parameters" variable will contain the parameters and a "results" parameter will contain the results that also may be modified.
Diagram

Used by Element secas

Instance

<secas run-on-error="false" service="" when="">
  <condition>{0,1}</condition>
  <actions>{1,1}</actions>
</secas>

Attributes

<table>
<thead>
<tr>
<th>QName</th>
<th>Type</th>
<th>Fixed</th>
<th>Default</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>run-on-error</td>
<td>boolean</td>
<td></td>
<td>false</td>
<td>optional</td>
</tr>
<tr>
<td>service</td>
<td>xs:string</td>
<td></td>
<td></td>
<td>required</td>
</tr>
</tbody>
</table>

The combined service name, like: "${path}.${verb}${noun}". To explicitly separate the verb and noun, put a hash (#) between them, like: "${path}.${verb}#${noun}".

| when             | restriction of xs:token |         |         | required |
