Schema documentation for service-eca-1.1.xsd

Publication date 7 august 2012

Table of Contents

Namespace: "" 1
Schemas 1
Main schema service-eca-1.1.xsd 1
Elements 1
Element secas 1
Element seca 1

Namespace: ""

Schemas

Main schema `service-eca-1.1.xsd`

Namespace	No namespace
Properties	attribute form default: `unqualified` element form default: `qualified`

Elements

Element secas

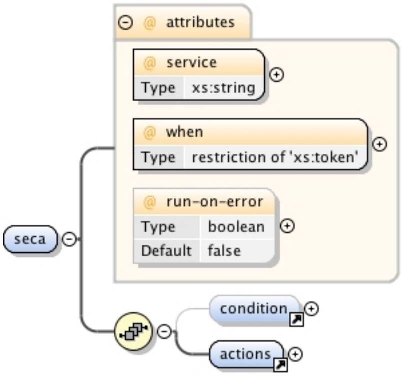
Namespace	No namespace
Diagram	secas 0..∞ seca
Properties	content: `complex`
Model	seca*
Children	seca

`Instance`	`<secas>`
	`  <seca run-on-error="false" service="" when="">{0,unbounded}</seca>`
	`</secas>`

Element seca

Namespace	No namespace

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.`

<table>
<tr><td>Diagram</td><td>

attributes
service
Type xs:string
when
Type restriction of 'xs:token'
run-on-error
Type boolean
Default false
seca
condition
actions

</td></tr>
<tr><td>Properties</td><td>content: complex</td></tr>
<tr><td>Used by</td><td>Element secas</td></tr>
<tr><td>Model</td><td>condition{0,1} , actions</td></tr>
<tr><td>Children</td><td>actions, condition</td></tr>
<tr><td>Instance</td><td><seca run-on-error="false" service="" when="">
  <condition>{0,1}</condition>
  <actions>{1,1}</actions>
</seca></td></tr>
<tr><td>Attributes</td><td>
<table>
<tr><th>QName</th><th>Type</th><th>Fixed</th><th>Default</th><th>Use</th></tr>
<tr><td><b>run-on-error</b></td><td>boolean</td><td></td><td>false</td><td>optional</td></tr>
<tr><td><b>service</b></td><td>xs:string</td><td></td><td></td><td>required</td></tr>
<tr><td></td><td colspan="4">The combined service name, like: "${path}.${verb}${noun}". To explicitly separate the verb and noun put a hash (#) between them, like: "${path}.${verb}#${noun}".</td></tr>
<tr><td><b>when</b></td><td>restriction of xs:token</td><td></td><td></td><td>required</td></tr>
</table>
</td></tr>
</table>