Developer's Guide

Advanced Engagement

5/3/2025

Advanced Engagement

Contents

Advanced Engagement enables customization based on business events. You can create new business events by writing new DSL code in the DSL files of your application, as detailed below. In current versions, only `Timeout` and `Search` events are available. To customize this model, you must first define your own events using the DSL loaded in the Browser-Tier Agents. Then, you can use the Advanced Rules Templates to create rules based on these events.

Creating Business Events

Customizing the DSL Files

When you create a new application with the tools script as detailed here, a default set of DSL files is created and used by your application. These files are defined in the `apps\<Your application name>\frontend\src\main\webapp\resources\dsl` directory. You can edit the **domain-model.xml** and add there a list of events, with specific conditions, related to your web pages' content. **Default domain-model.xml**

```
<?xml version="1.0" encoding="utf-8" ?>
<properties debug="false">
    <events>
        <event id="" name="">
            <!-- Add your code here -->
        </event>

        <event id="SearchEvent" name="Search">
                <trigger name="SearchTrigger" element="" action="click" url="" count="1" />
                <val name="searchString"                value="" />
        </event>
        <event id="TimeoutEvent10" name="Timeout-10" condition="">
                <trigger name="TimeoutTrigger" element="" action="timer:10000" type="timeout"
url="" count="1" />
        </event>
        <event id="TimeoutEvent30" name="Timeout-30" condition="">
                <trigger name="TimeoutTrigger" element="" action="timer:30000" type="timeout"
url="" count="1" />
        </event>

    </events>

</properties>
```

By using the `<event></event>` element, you can create as many BUSINESS event that you need. These events can be tied to the HTML components of your page. For instance, in the genesyslab sample, the **domain-model.xml** file declares the **SearchEventClick**, which is related to the `#top-search-input` textbox, the search box on top of all the genesyslab.com pages. **Source code of genesyslab.com**

```
<fieldset  id="top-search">
   <div class="search-bar clearfix">
      <input type="text" id="top-search-input" value="Search"></input>
      <a href="#" id="top-search-submit" class="sprite"></a>
      <input type="hidden" value="1" name="StartRow">
   </div>
//...
```

Event in the domain-model.xml file

```
<?xml version="1.0" encoding="utf-8" ?>
<properties debug="false">
<events>
//...
   <event id="SearchEventClick" name="Search">
      <trigger name="SearchTrigger" element="#top-search-submit" action="click" url=""
count="1" />
      <val name="searchString"                value="$('#top-search-input').val()" />
   </event>
//...
</events>
</properties>
```

You can create several events with the same name but different identifiers. This is useful to associate several HTML components with the same event if these HTML components have the save function. For instance, you can define several events associated to the Search feature and these events will all have the same name. For each event, you can define triggers which describe the condition to match to submit the event:

- Triggers enable to implement timeouts.
- Triggers can be associated with DOM events.
- You can define several triggers for the same event (see <trigger> for further details).

For further details about each element, see the Monitoring DSL API.
User's Guide Next Step: Publish the CEP Rules Template.

Implementing Events in your Webpages

Another solution consists in using the Monitoring JavaScript API, which allows you to submit events and data from the HTML source code. In this case, you can use the _gt.push() method which allows you to decide when events should be submitted and which data they generate, directly from the webpages. See Monitoring JavaScript API Reference for further details.

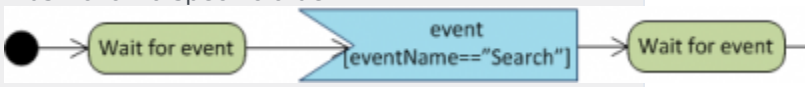
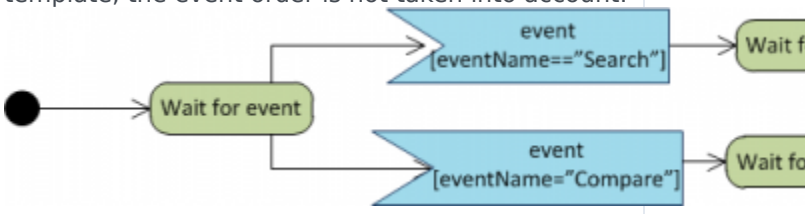
Default Event-Based Template

The templates below define conditions to fulfill in order to generate the actionable event:

- The `Singleton` template allows a single event identified by its name.
- The `Sequence` template defines a sequence of incoming events in a predefined order.
- The `Set` template specifies an unsorted set of incoming events.

All these templates associated with event names are available in the CEP Rules Templates. For further details, see Simple Engagement.

Singleton	
Description	The template receives each single event as a

	formal parameter. If the event's value matches the condition's event name, then the actionable event is sent to the Web Engagement Backend Server.  GWM single.png
Expression Example	**When** event with name $name **Then** generate actionable event
Sequence	
Description	This template analyses the event stream received from the categorization engine and builds the sequence of events by event names. As soon as the event sequence is completed, the actionable event is submitted. Note that the event sequence must follow a specific order.  Click to enlarge.
Expression Example	**When** event with name $name save as $event1 **and** event following `$event1` with name `$name2` save as `$event2` (...) **and** event following $event^{n-1} with name $name^{n} save as $event^{n} **Then** generate actionable event based on $event^{n}
Set	
Description	This template collects the events by event names. As soon as the event set is completed, the actionable event is submitted. If you use this template, the event order is not taken into account. 

	GWM Set.png
Expressions	**When** event with name $name1 save as $event1 **or** event with name $name2 save as $event2 (...) **or** event with name $name^{n} save as $event^{n} **Then** generate actionable event