



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Orchestration Server Developer's Guide

Work With Chat

Work With Chat

The following SCXML strategy receives an inbound chat from the customer. Submit is called to find an available agent and the chat interaction is redirected to the agent. Upon termination of the chat interaction, a transcript of the chat session is e-mailed to the customer.

```
<scxml version="1.0" xmlns="http://www.w3.org/2005/07/scxml"
  xmlns:queue="http://www.genesyslab.com/modules/queue"
  xmlns:ixn="http://www.genesyslab.com/modules/interaction"
  xmlns:ws="http://www.genesyslab.com/modules/ws"
  xmlns:dialog="http://www.genesyslab.com/modules/dialog"
  _persist="false" >
  <datamodel>
    <data ID="reqid" expr="''"/>
    <data ID="ixnid" expr="''"/>
    <data ID="routetarget" expr="''"/>
  </datamodel>
  <initial>
    <transition target="listening">
    </transition>
  </initial>
  <state id="main" initial="listening">
    <state id="listening">
      <onentry>
        <log expr="'Listening ...'" />
      </onentry>
      <!-- This script handles chat interactions -->
      <transition event="interaction.present"
cond="_genesys.ixn.interactions[_event.data.interactionid].category == 'chat'"
target="find_agent">
        <script>
          _data.ixnid = _event.data.interactionid;
        </script>
      </transition>
    </state>
    <state id="find_agent">
      <onentry>
        <log expr="uneval( global )" />
        <!-- Submit URS request to find an agent. -->
        <queue:submit route="false">
          <queue:targets>
            <queue:target name="'3'" type="agent"
statserver="'statserver'" />
            <queue:target name="'4'" type="agent"
statserver="'statserver'" />
            <queue:target name="'5'" type="agent"
statserver="'statserver'" />
          </queue:targets>
        </queue:submit>
      </onentry>
      <transition event="queue.submit.done" target="deliver_to_agent">
        <log expr="uneval( _event )" />
        <assign location="_data.routetarget"
expr="_event.data.resource" />
      </transition>
      <transition event="error.queue.submit" target="error">
        <log expr="uneval( _event )" />
      </transition>
```

```
</state>
<state id="deliver_to_agent">
  <onentry>
    <log expr="'Routing ...'" />
    <script>
      var hint = new Object();
      hint.outqueues = new Object();
      hint.outqueues.Orchestration_queue = "queue for
outbound response";
    </script>
    <!-- For redirect, the 'to' object is gotten from data
returned from queue:submit -->
    <ixn:redirect requestid="_data.reqid"
interactionid="_genesys.ixn.firstixnid"
      from="'scxml'" to="_data.routetarget" hints="hint"/>
    </onentry>
    <transition event="interaction.redirect.done" target="responding">
      <log expr="uneval( _event )" />
    </transition>
    <transition event="error.interaction.redirect" target="error">
      <log expr="uneval( _event )" />
    </transition>
  </state>
  <state id="responding">
    <onentry>
      <log expr="'Wait for interaction to be terminated and send
chat transcript...'" />
      <script>
        var customer_email_address = "undefined";
      </script>
      <if cond="typeof
_genesys.ixn.interactions[_data.ixnid].udata.EmailAddress != 'undefined'">
        <assign location="customer_email_address"
expr="_genesys.ixn.interactions[_data.ixnid].udata.EmailAddress" />
      <else/>
        <!-- No email address provided so continue without
sending transcript. -->
        <event name="no.email.address"/>
      </if>
    </onentry>
    <transition event="interaction.notcontrolled">
      <log expr="'Initial interaction is no longer controlled by
this scxml.'" />
    </transition>
    <transition event="interaction.deleted">
      <!-- Initial interaction has been terminated -->
      <if cond="customer_email_address != 'undefined'">
        <!-- Email the chat transcript. Message text is a
standard response. -->
        <ixn:createmessage type="outbound_new"
          subject="'Chat transcript'"
          relatedixnid="_data.ixnid"
          msgsrc="'gdata:config\\SR.0000Na5C1F3500FY'"
          to="customer_email_address" />
      </if>
    </transition>
    <transition event="msgbased.createmessage.done" target="cleanup"/>
    <transition event="no.email.address" target="cleanup"/>
    <transition event="error.msgbased.createmessage"
target="requesterror">
      <log expr="uneval( _event )" />
    </transition>
  </state>
</scxml>
```

```
        </transition>
    </state>
    <state id="cleanup">
        <onentry>
            <log expr="'Cleanup logic goes here...'" />
        </onentry>
        <transition target="exit"/>
    </state>
    <state id="requesterror">
        <onentry>
            <log expr="'Request error processing goes here...'" />
        </onentry>
        <transition target="error"/>
    </state>
</state>
<final id="error"/>
<final id="exit"/>
</scxml>
```