puzzel. Solving Customer Interactions

Run script after chat has ended

Below is an example showing code (only showing the code in the form-tag) needed to initiate a script after ending a chat. The example starts an external survey on the web page. The script can be run after having ended the chat-dialogue (first time clicking on the x) or after having closed the chat window (second time clicking on the x), often determined by weather the e-mail log option is made available or not.

```
<div id="survey1" style="display: none;">
<span>Question after chat-dialogue is ended</span>
<input id="question1" name="question1" type="text" value="How was your experience?" />
</div>
<div id="survey2" style="display: none;">
<span>Question after chat-window has closed</span>
<input id="question2" name="question2" type="text" value="How was your experience?" />
<p>
You must choose one of the two event handlers.
</p>
</div>
<div id="someID" class="intelecomchat">
</div>

<script>
$(document).ready(function () {
$('#someID').intelecomChat({
customerKey: '12345',
queueKey: 'Q_CHAT',
showStarter: false,
// trigged after a chat-dialogue is ended (first time x)
onChatEnd: function () {
$('#survey1').show();
// $('#someID').hide();
},
// trigged when the chat window is closed (second time x)
onChatWindowClose: function () {
$('#survey2').show();
$('.intelecomchatstarter').hide();
}
});
});
</script>
```

In addition to running the function, you can add dependencies like if the chatter has been connected to an agent, as an opposite to having ended the chat while waiting in queue. Also, it is possible to add data related to the chat session into the function (and URL), like the chatters id and variables. In this way, you can run the functions to a more precise group of users containing more details from the chat session. The example below adds data from the chat to the function, and the script is only run if the chatter has been connected to an agent:

```
onChatWindowClose: function (event, command, data) {
if(data.WasConnected){
// do something
}
}
```

“data“ contains the following structure:

```
{
        "ActiveUsers": chat.vm.activeUsers(),
        "ChatMessages": chatMs,
        "SurveyResult": chat.vm.surveyResult(),
        "SurveyComment": chat.vm.surveyComment(),
        "BeenInConference": chat.vm.beenInConference(),
        "BrowserSupported": chat.vm.browserSupported(),
        "ChattersEmailAddress": chat.vm.toEmailAddress(),
        "ChattersName": chat.model.NickName,
        "ChatIssue": chat.model.ChatIssue,
        "ChatId": chat.model.ChatId,
        "IqSessionId": chat.model.IqSessionId,
        "RequestId": chat.model.RequestId,
        "IsMobile": chat.model.IsMobile,
        "Variables": chat.model.Variables,
        "LanguageCode": chat.model.LanguageCode,
        "WasConnected": chat.wasConnected,
        "LastConnectedUserId": chat.lastConnectedUserId,
        "QueueKey": chat.model.EnteredQueueKey || chat.model.QueueKey

     }
```

“chatMessages” contains the following structure:

```
{
                  "Message": item.Message(),
                  "Sent": item.Sent(),
                  "NickName": item.NickName(),
                  "UserId": item.UserId()
               }
```