How to Configure Script Runner listener to subscribe to EXALATED event

Last Modified on 04/03/2024 6:25 am EDT

With Script Runner you can set up triggers to perform actions when issues have been Exalated. Let's assume we would like to trigger a sync back, once the issue has been exalated and modify our issue in some way. Let's say we would like a Mood custom field to be set to 'Happy' and a description of an issue to be 'Created from the Script Runner'. In order to achieve this one can easily configure a custom Script Listener to subscribe to the EXALATED Jira event.

To configure a Script Runner to subscribe to an EXALATED Jira event:

1. Install a Script Runner plugin for Jira:

2. After a successful installation navigate to the Script Listeners section of the Script Runner plugin:

3. Pick **Custom listener** from the list. This takes you to the page where you can actually create your custom listener:

4. Pick up the desired project from a dropdown list and choose a 'com.exalate.api.domain.trigger.EXALATED' event type from the list:

5. Next step would be to write your custom listener script. It is written using Groovy language. Here's an example of such a script. The actual implementation can change depending on the JIRA version.

```
import com.atlassian.jira.event.issue.AbstractIssueEventListener
import com.atlassian.jira.event.issue.IssueEvent
import com.atlassian.jira.event.type.EventType
import com.atlassian.jira.event.type.EventTypeManager
import com.atlassian.jira.issue.Issue

import org.slf4j.Logger
import org.slf4j.LoggerFactory

class DefaultAccountListener extends AbstractIssueEventListener {

    private final Logger LOG = LoggerFactory.getLogger(DefaultAccountListener)

    private static final String EXALATED_EVENT_TYPE = "com.exalate.api.domain.trigger.EXALATED"
    private static final String EXALATE_UPDATED_EVENT_TYPE = "com.exalate.api.domain.trigger.UPDATED"

    @Override
    void customEvent(IssueEvent event) {
        Issue issue = event.issue
        EventType eventType = eventTypeManager.getEventType(event.getEventTypeId())
        LOG.debug("Custom event caught for issue ${issue.key}: type [${eventType.type}], name [${eventType.name}], nameKey [${eventType.nameKey}]")

        if (eventType.name.equalsIgnoreCase(EXALATED_EVENT_TYPE)) {
            LOG.debug("EXALATED event caught for issue ${issue.key}")
        }
    }
}
```

6. Save the listener, then navigate to an issue you'd like to Exalate and perform an Exalate operation:

7. Now let's wait until the synchronization is performed (there is a lozenge on the right side which indicates the state of the synchronization):

8. Once the issue has been Exalated, refresh the page and check whether the changes were applied:

We can see that the Mood has been changed to 'Happy' and that the description has changed as well.