How to Configure Script Runner listener to subscribe to EXALATED event

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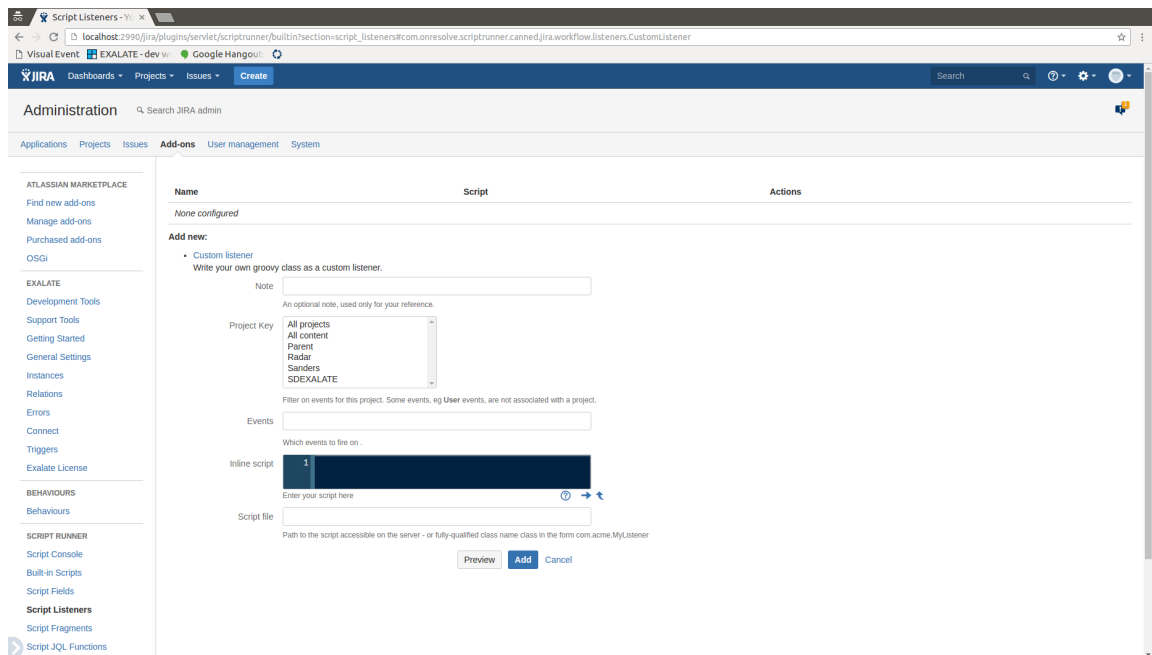
1. Install a Script Runner plugin for Jira:
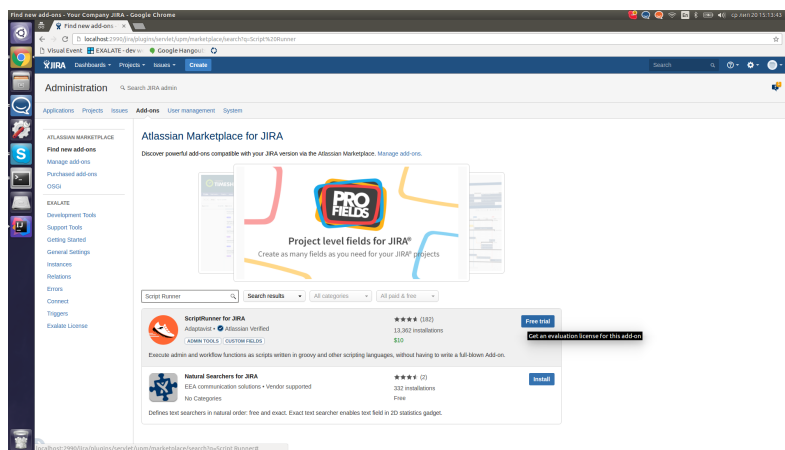
2. After a successful installation navigate to the Script Listeners section of the Script Runner plugin:
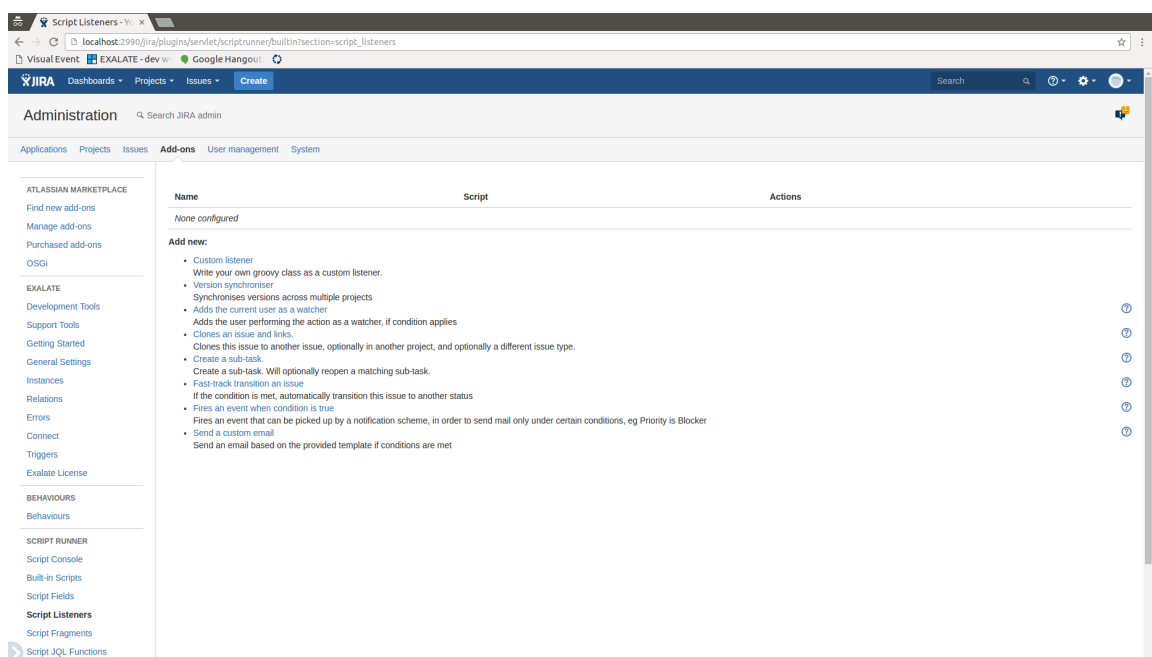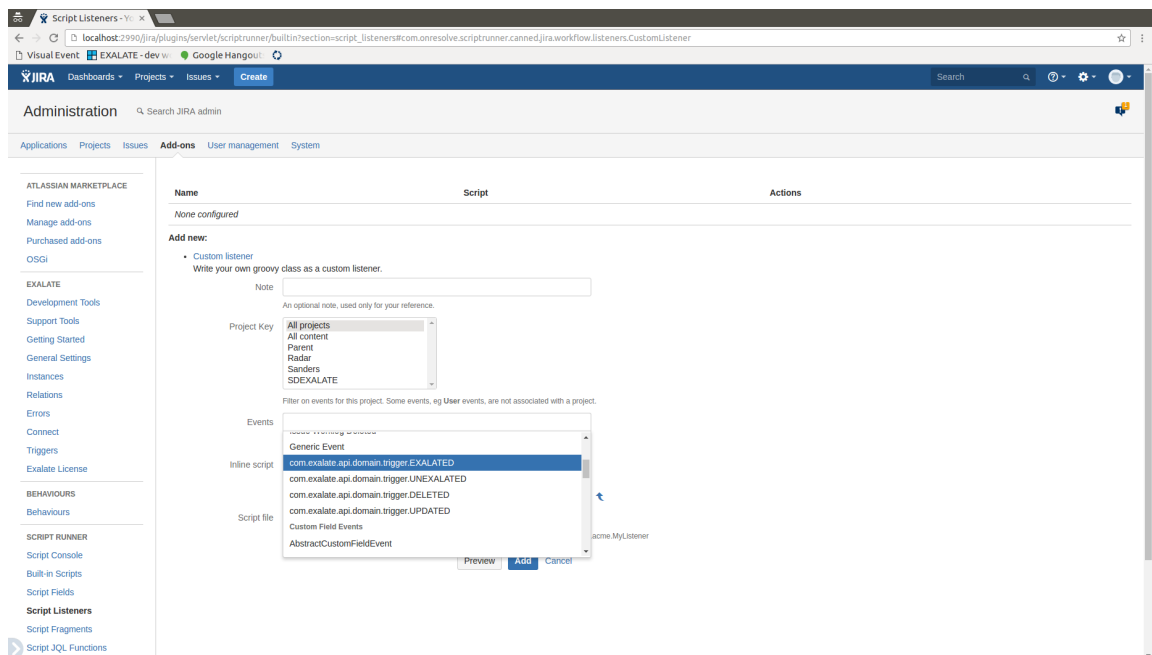

3. Pick **Custom listener** from the list. This takes you to the page where you can actually

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.
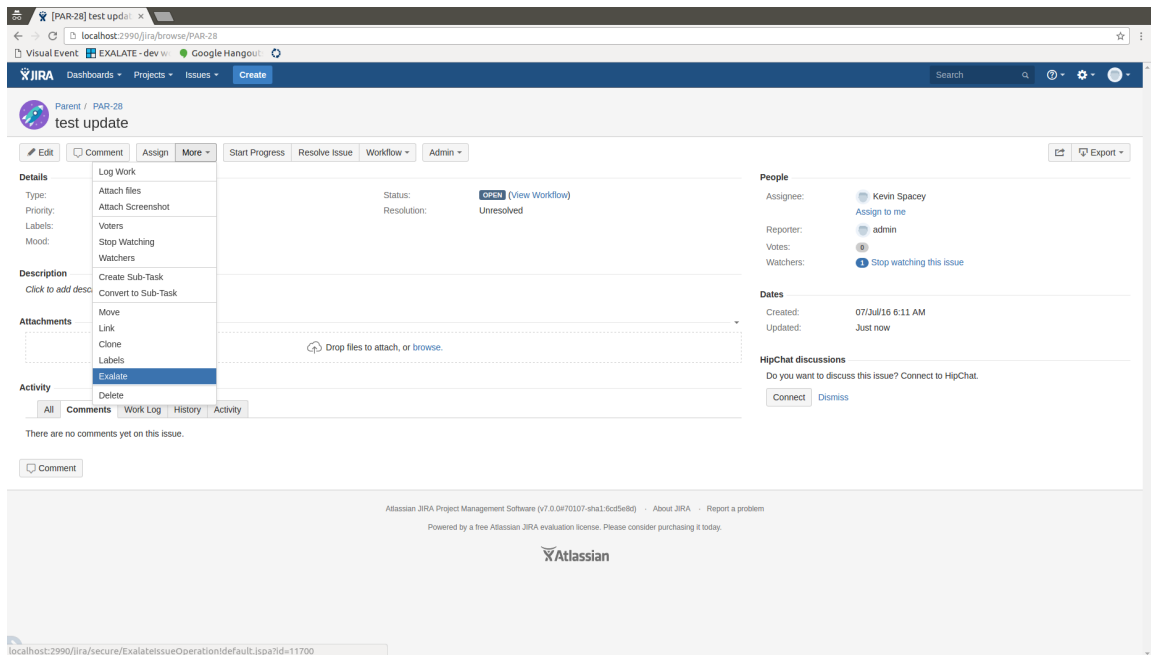
```
1 import com.atlassian.jira.event.issue.AbstractIssueEventListener
2 import com.atlassian.jira.event.issue.IssueEvent
3 import com.atlassian.jira.event.type.EventType
4 import com.atlassian.jira.event.type.EventTypeManager
5 import com.atlassian.jira.issue.Issue
6
7 import org.slf4j.Logger
8 import org.slf4j.LoggerFactory
9
10 class DefaultAccountListener extends AbstractIssueEventListener {
11
12    private final Logger LOG = LoggerFactory.getLogger(DefaultAccountListener)
13
14    private static final String EXALATED_EVENT_TYPE = "com.exalate.api.domain.trigger.EXALATED"
15    private static final String EXALATE_UPDATED_EVENT_TYPE = "com.exalate.api.domain.trigger.UPDATED"
16
17    @Override
18    void customEvent(IssueEvent event) {
19       Issue issue = event.issue
20       EventType eventType = eventTypeManager.getEventType(event.getEventTypeId())
21       LOG.debug("Custom event caught for issue ${issue.key}: type [${eventType.type}], name [${eventType.name}], nameKey [${eventType.nameKey}]")
22
23       if (eventType.name.equalsIgnoreCase(EXALATED_EVENT_TYPE)) {
24          LOG.debug("EXALATED event caught for issue ${issue.key}")
25       }
26    }
27 }
```
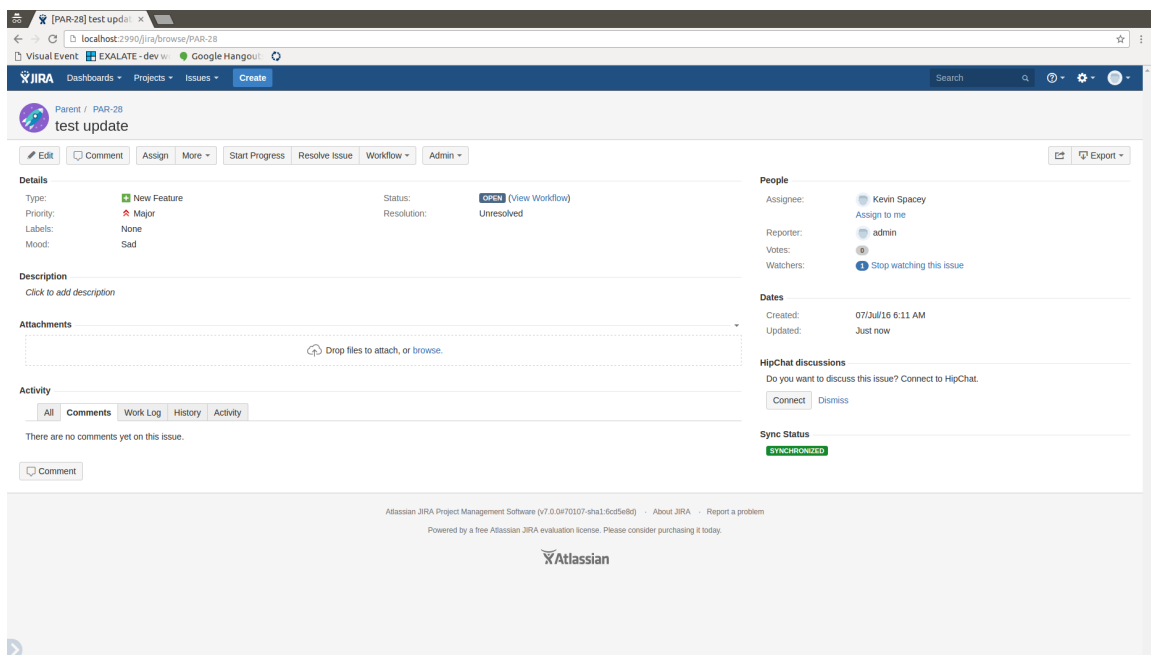
6. Save the listener, then navigate to an issue you'd like to Exalate and perform an Exalate operation:

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.

Product

About Us ⮕

Release History ⮕

Glossary ⮕

API Reference ⮕

Security ⮕

Pricing and Licensing ⮕

Resources

Subscribe for a weekly Exalate hack ⮕

Academy ⮕

Blog ⮕

YouTube Channel ⮕

Ebooks ⮕

Still need help?

Join our Community ⮕

Visit our Service Desk ⮕

Find a Partner ⮕