# ScriptRunner IT Service Management (ITSM)

Lab 4

October 2018

# **Table of Contents**

SCRIPTRUNNER IT SERVICE MANAGEMENT (ITSM)	3
<u>OVERVIEW</u>	3
WHAT DO THESE LABS SHOWCASE?	
TECHNOLOGY USED	3
PREREQUISITES	3
LAB 4 – ADDING SCRIPTED FIELDS TO COUNT ALERTS	4
BEFORE YOU START	4
STEP 1	6
STEP 2	8
TEST STEP	9

# ScriptRunner IT Service Management (ITSM)

### Overview

This lab brings together ScriptRunner features that allow ITSM practices such as updating all Linked incidents, count alerts using Scripted Fields, Using PostFunctions to set Remedial Action Priority, Optimising the Jira Service Desk behaviour, and constraining the Create Issue dialog with customised rules.

### What do these labs showcase?

- How ScriptRunner enables a one-stop shop for updating all linked incidents in one go, saving time and cost for IT teams by automating these tasks and provide time to focus on more critical issues
- Using ScriptRunner Behaviours, you can now control the user interaction in a Jira Service
  Desk form, enabling IT teams to decipher key information by controlling what input the user
  must provide in a support desk form
- Using the power of Behaviours again, we see how to constrain the Create Issue dialog in Jira, providing more control over how your Jira users interact with the form
- Supercharging custom fields to show the number of alerts in an issue, users will learn how powerful Scripted Fields is with this example
- Tailor your Jira workflow to put in place a post function that sets a remedial action priority, for example, automatically set the remedial action issue priority to 'Highest' if it has 2 or more linked alerts.

# Technology used

The following technology components are used in this solution:

- ScriptRunner for Jira Server 5.4.12
  - Scripted Fields
  - Post Functions
  - Behaviours
  - Script Listener
- Jira Service Desk 3.8.1
- Jira Software 7.5.0
- Jira Core 7.5.0

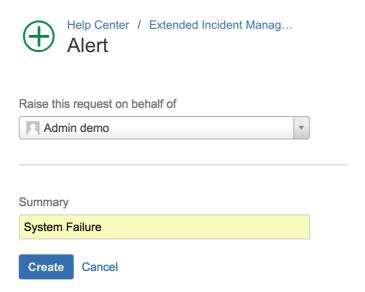
## Prerequisites

- A demo/magic button Jira with Service Desk and Software license
- A demo/magic administration account
- A ScriptRunner for Jira Server license (evaluation or commercial):
- A basic understanding of Jira

# Lab 4 – Adding Scripted Fields to Count Alerts

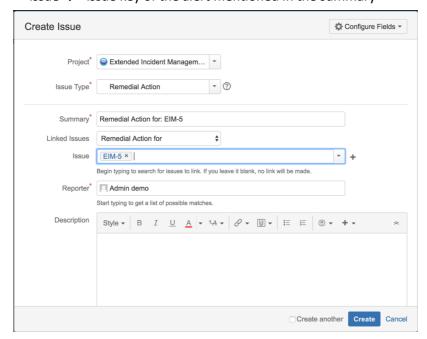
# Before you start

Create an Alert Issue using the customer portal. Give it a summary to reflect a real-life alert, such as 'System failure'. Once this has been created, go and create **another** one using the customer portal, again with a summary such as 'System failure'. Here we are just replicating the sort of Alerts that an external monitoring system may create for us in an ITSM stack.



Using the normal Create button in the banner menu create a new Remedial Action issue

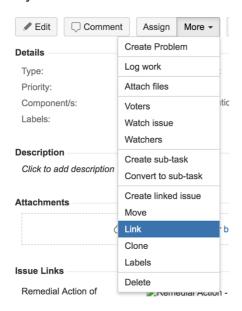
- Summary → 'Remedial Action for: <issue key of one of the alerts>
- Linked Issues → Remedial Action for
- Issue → <issue key of the alert mentioned in the summary>

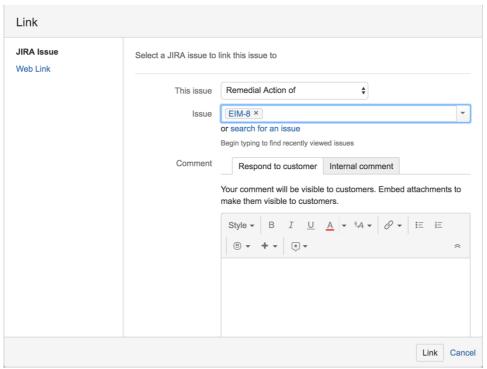


Now go to the other Alert that you created earlier and manually link it to the newly created Remedial Action issue using the same link type. You can do this by finding the issue and clicking on More  $\rightarrow$  Link:

- This issue → Remedial Action of
- Issue → <issue key of the Remedial Action issue>

# System failure

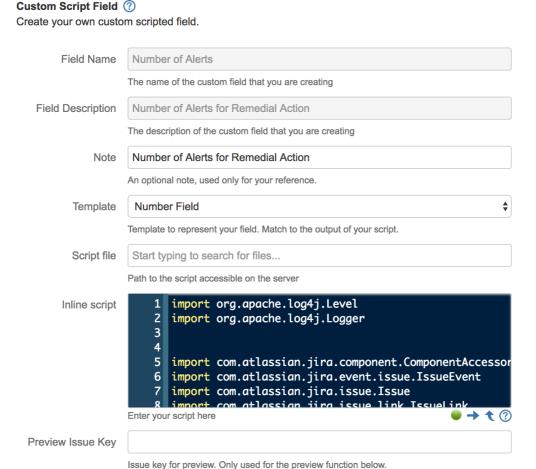




Now you will have one Remedial Action with two Alerts linked to it with the same link type.

### Step 1

- Go to Add-ons → Script Fields using the cog in the top right corner
- Then click the Add New Item button and choose Custom Script Field
- Configure the screen to look like the below:



• Here is the code snippet:

```
import org.apache.log4j.Level
import org.apache.log4j.Logger

import com.atlassian.jira.component.ComponentAccessor
import com.atlassian.jira.event.issue.IssueEvent
import com.atlassian.jira.issue.Issue
import com.atlassian.jira.issue.link.IssueLink

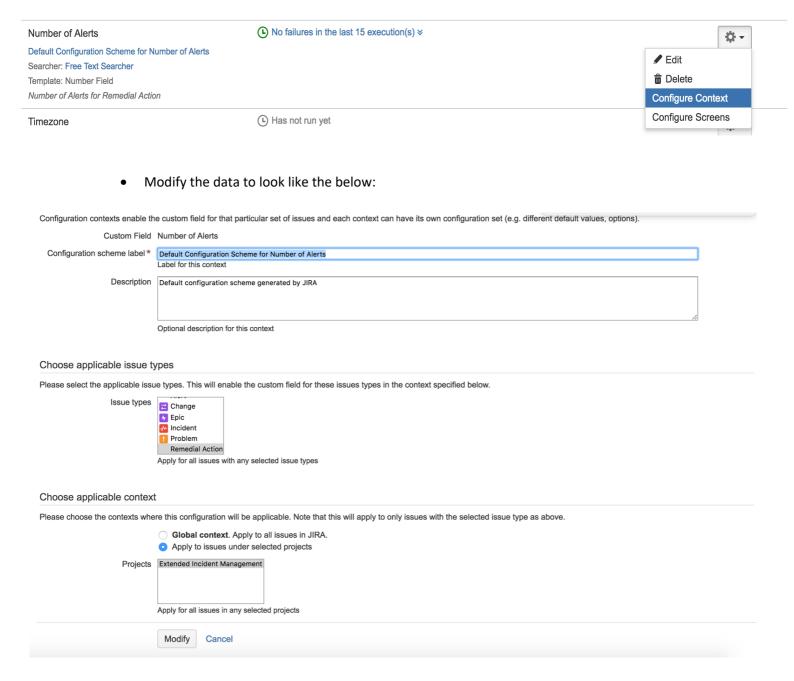
import com.atlassian.jira.ComponentManager
import com.atlassian.jira.issue.CustomFieldManager
import com.atlassian.jira.issue.fields.CustomField
String scriptCode = "EIM Alert Count:"
String robotUsername = "admin"
```

```
// List of link types we are counting for Incidents
List<String> linkNamesForIncident = [ "Remedial Action Link" ]
        // Trigger an event of the given type {\tt ID}
// Set logging appropriately
        // Set logging appropriately
        Logger logger = log
        logger.setLevel( Level.ALL )
        // Get the current issue object
        Issue triggerIssue = issue
        logger.debug( "$scriptCode Script triggered on issue ${
triggerIssue.getKey() }" )
        // Get all the trigger issue links
        List<IssueLink> triggerIssueInwardLinks =
ComponentAccessor.getIssueLinkManager().getOutwardLinks( triggerIssue.getId() )
        // Get all the link types that we are looking for
List<IssueLink> filteredLinks = triggerIssueInwardLinks.findAll {
it.getIssueLinkType().getName() in linkNamesForIncident &&
it.destinationObject.issueType.name == "Alert"}
        // Get the number of filtered links
        double AlertCount = (double) filteredLinks.size()
        log.info(AlertCount)
        // Return that value
        return AlertCount
```

Click Add

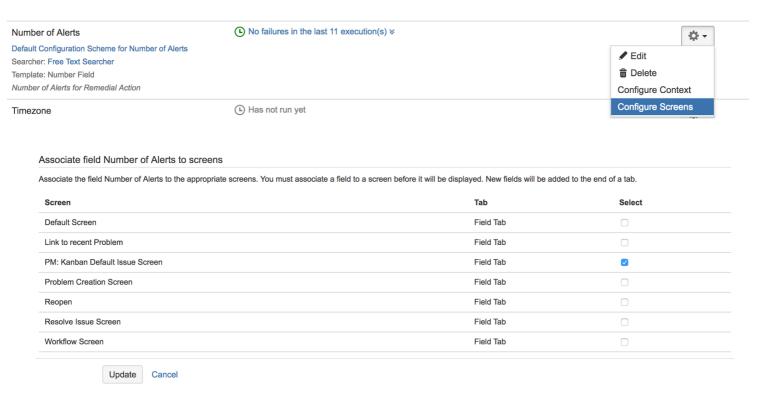
# Step 2

• Click on the cog to the right of the new field you just created



- Click Modify
- Navigate back to the Script Fields sections. Add-ons → Script Fields using the cog in the top right corner

• Now go to the Configure Screen option in the same cog and add the field to the PM: Kanban Default Issue Screen



Click Update

# Test Step

- Click the Issue drop down in the banner menu and find the Remedial Action issue created in this lab
- Check the Number of Alerts field in showing the value 2

Extended Incident Management / EIM-5

