

Improving JIRA project flow with ScriptRunner

Introduction

This page contains information on what needs to be done to make changes to JIRA project flow and how to test them before applying them in production JIRA.?

Automating creation of issues

Request access to test JIRA instance

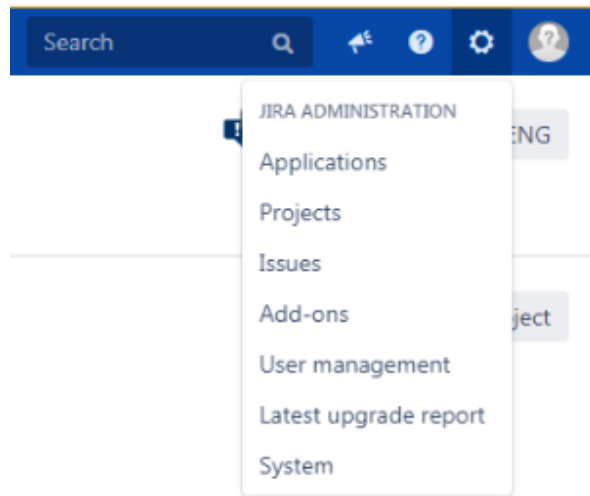
If you don't have it, request access to? [JIRA test instance](#) from SRE team on their [Self Service page](#). Most likely you will have access to it already.

Request global administration rights

To be able to manipulate the project flow freely, you'll first need to obtain JIRA global admin rights - JIRA project admin rights are not enough to make changes to the workflow.

One good and tested method is to ask a member of SRE team to grant such access. One such person is Wojciech Gasperowicz. Open Edit Workflow page

Once global admin rights are obtained, click Projects in the JIRA Administration menu?



Find and click PLENG.

You'll find many configuration options in that page. What we're interested in at the moment are the Workflows. You can either click Workflows in the left panel or go straight to editing the one you're interested in.

Either way, in case of Epic workflow, you'll end up here:

i You are editing a draft. There are unpublished changes.

[Publish](#)
[Discard](#)
[View Original](#)

i This draft was last edited by **you** at 26/Nov/18 17:11.

Diagram **Text** Export ▾

Step Name (id)	Linked Status	Transitions (id)	Actions
Done (11)	DONE	Business Validation (91) >> BUSINESS VALIDATION Draft (101) >> DRAFT	Add transition Delete Transitions Edit View Properties
Draft (12)	DRAFT	Refinement (11) >> REFINEMENT Done (51) >> DONE	Add transition Delete Transitions Edit View Properties
Refinement (13)	REFINEMENT	Ready for Implementation (21) >> READY FOR IMPLEMENT... Draft (61) >> DRAFT	Add transition Delete Transitions Edit View Properties

You can switch between Diagram and Text views.

Add a post function

To add a Post Function to a transition (Text view is preferable in this case) and go to Post Functions tab.

Screen: None - it will happen instantly

Triggers **0** Conditions **0** Validators **0** **Post Functions 6**

Click Add post function on the right, choose Script Post-Function [ScriptRunner] and click Add.

Automating Creating issues during a transition

Do not be discouraged by the name of the script - it's actually this one that will help us automate creating issues during a transition:

➔ **Clones an issue, and links** ⓘ

Clones this issue to another issue, optionally another project and issue type, and creates a link.

This is the screen where all the fun happens, ie. Groovy scripts and actual settings. Contrary to script's description, it can actually create a completely new issue, depending on what you set up.

Confirm your script by clicking Update.

Remember to move the just-created script to the proper place in Post Functions list (bottom in most cases).

Save draft

Your changes will not be applied to the workflow until you Publish them. After clicking Update in the previous step, you should now see an orange notification box giving you an option to Publish or Discard your changes.

Changing field default values

To change default field values in various issue types, you need to use **Behaviours** plugin. To access it, open Administration menu in the top-right corner and click add-ons and then Behaviours in the left panel.

There, click Fields for the field behaviour you want to change.

Hints, tips and tricks

In general, using ScriptRunner can be tricky to use, so I'll give some tips in this section.

You can also find hints for specific tasks in example scripts in [jira-scriptrunner repository](#).

IDE

It's actually really worth it to setup your IntelliJ to help you with writing scripts even if it means copying them to the browser with each change. See [links](#) section on instructions how to do it.

Overzealous script window

There are some situations where the script field will unnecessarily underline parts of your code in red. One such example is `passesCondition` variable in `Condition` field - according to documentation, it's already predefined for Conditions but is considered undeclared anyway. But the scripts run well anyway.

Order of post-functions and actions

For your post-functions to work properly, you need to move them below *"Fire a **Generic Event** event that can be processed by the listeners."* in the Post-functions list.

Remember that scripts added to "Additional issue actions" field in a "Clones an issue, and links" post-function are executed AFTER a new issue is created, so adding create/don't create conditions to this field won't work.

Issue key

An issue created by a "Clones an issue, and links" post-function is not given an ID until the whole post-function is done, so you cannot actually use it to e. g. adding its ID to summary.

Helpful links and sources

<https://scriptrunner.adaptavist.com/latest/jira/quickstart.html>?- ScriptRunner documentation

<https://scriptrunner.adaptavist.com/4.3.5/bitbucket/DevEnvironment.html>?- instruction how to set up a dev env on your machine (not tested yet)

<https://community.atlassian.com/t5/Jira-questions/How-to-validate-field-depending-on-component-selection-simple/qaq-p/352621>?- Adaptavist (ScriptRunner creators) people are quite active on Atlassian forums, so many questions are already answered.

<https://bitbucket.itcm.oneadr.net/projects/FIRE/repos/jira-scriptrunner/browse>?- repository with example scripts?