

How To - Automated Issue Tracking with JIRA and Jenkins

Part 1

Jira side customization

Jira webhook integration:

Click on the “Cog wheel” on the right hand side, then click on “System” this time, then, scroll down (in the very bottom actually), and on the left hand side pane, look for “Webhooks”. Click on that. Then, click on “+Create a Webhook” on the right hand side. Here, add following SUFFIX after your Jenkins server url: /jira-trigger-webhook-receiver/. So for example, your Jenkins server is running on <http://10.0.0.1:8080>, the webhook will be <http://10.0.0.1:8080/jira-trigger-webhook-receiver/>.

WebHooks

+ Create a WebHook ?

New WebHook Listener

New Jenkins

New WebHook Listener

New WebHook Listener

Name* New WebHook Listener

Status* Enabled Disabled

URL* ins-632459431.ap-south-1.elb.amazonaws.com/jira-trigger-webhook-receiver/

You can use the following additional variables in the URL: \${board.id}, \${comment.id}, \${issue.id}, \${issue.key}, \${mergedVersion.id}, \${modifiedUser.key}, \${modifiedUser.name}, \${project.id}, \${project.key}, \${sprint.id}, \${version.id}

Read more

Description

Events Issue related events

Events for issues and worklogs. You can specify a JQL query to send only events triggered by matching issues.

All issues

Syntax help

Worklog

☐ created

☐ updated

☐ deleted

Issue

☒ created

☒ updated

☐ deleted

☐ worklog changed

Issue link

☐ created

☐ deleted

Comment

☒ created

☐ updated

☐ deleted

Additionally, tick the checkbox “created” and “updated” under Issue, and “created” under Comment in the Events section. What we are essentially configuring is that whenever new issue is created or updated, or a comment is created, Jenkins is triggered by Jira. So essentially, by updating our issues or by adding comments, we can control Jenkins pipelines. This is how Jira triggers Jenkins pipeline.

Finally, scroll down and click on CREATE to create this webhook. We can integrate multiple Jenkins pipeline using this feature, as we can create webhooks for multiple Jenkins server URLs.

Unfortunately, webhooks is just one part. You would also need TICKET PATTERN. As we know, in Jira, everything is an issue (including stories, epics etc.). So to keep track of it, Jira system generates a prefix before every issue. Think of it as a Primary Key value for Jira.

To get this Ticket pattern, click on the “Cog Wheel” on upper right side, then click on Project. Click on the project which you want to integrate. Then click on “Details” on the left side. The KEY is the TICKET PATTERN required to integrate with Jenkins.

Project settings

Details

Name* TEST

Key* DEMO ⓘ

URL

Project type* Software ⓘ

Project category None ⓘ

Avatar* select image

Description

You would also be needing the Jira server URL, Jira server name and the admin username/password. Except the admin username/password (which you should remember), server URL and name can be found in System Settings.

Plugins to be installed to achieve this

- Jira Plugin
- Generic Webhook Trigger Plugin
- Jira Trigger Plugin

Jenkins side customization

Ensure you have the below configuration done inside Manage Jenkins > Configure System.

Configuring Jira sites:

JIRA Steps

JIRA sites

Name Jira

URL http://novartis.devops.altimetrik.io:8080

Connection Timeout(ms) 10000

Read Timeout(ms) 10000

Choose Login Type:

☒ Basic

User Name admin

Password

☐ OAuth

List of JIRA sites

Jira trigger related configuration:

JIRA Trigger Configuration

JIRA root URL	<input type="text" value="http://novartis.devops.altimetrik.io:8080/"/>
JIRA username	<input type="text" value="admin"/>
JIRA password	<input type="password" value="*****"/>
Reply to JIRA for build updates?	<input type="checkbox"/>

Creating Jenkins Freestyle Job:

Step 1 Provide your GIT repository details which is going to be used here to run any script etc.

Source Code Management

☐ None
☐ AWS CodePipeline
☒ Git

Repositories

Repository URL

Credentials

Branches to build

Branch Specifier (blank for 'any')

Repository browser

Additional Behaviours

☐ Mercurial
☐ Multiple SCMs

Step 2 Provide what do you want to do with Jira using this job for example here we are filtering particular project and then if the issue status from that project gets updated (here anything In Progress) then a configured web hook will trigger a job.

Build Triggers

☐ Trigger builds remotely (e.g., from scripts)
☐ Build after other projects are built
☐ Build periodically
☐ Build when a change is pushed to BitBucket
☐ Build when a change is pushed to GitLab. GitLab webhook URL:
☐ Build when a comment is added to JIRA
☒ Build when an issue is updated in JIRA

JQL filter

Changelog matcher

JIRA Field Matcher

JIRA field ID

☒ Compare new value
New value
☐ Compare old value

Parameter mapping

☐ Build when another project is promoted
☐ Enable Artifactory trigger
☐ Gerrit event
☐ GitHub hook trigger for GITScm polling
☐ Gitlab Merge Requests Builder
☐ Monitor Docker Hub/Registry for image changes
☐ Poll SCM

Step 3 This step is just for a demo purpose here and trying to execute a script from repository.

Build

Execute shell

Command

```
echo "This job will be triggered when there is change in JIRA ticket status"
sh ./run.sh
```

See [the list of available environment variables](#)


Advanced...


Add build step





After configuring as shown above, you will need to go to any issue from the above configured project and change its status to see if the webhook is working as expected and triggering a particular job in Jenkins.






GitLab side customization:


Created a sample repository in GitLab.




Jenkins-Jira-integration 
Project ID: 12


  Star 0  Fork 0  Clone

 Add license  6 Commits  1 Branch  0 Tags  215 KB Files


master 


jenkins-jira-integration / 


History


 Find file


Web IDE





 Update run.sh
Administrator authored 20 minutes ago


27090c26 


 README



 Add CHANGELOG


 Add CONTRIBUTING

 Enable Auto DevOps

 Add Kubernetes cluster

 Set up CI/CD

Name	Last commit	Last update
 README.md	Initial commit	18 hours ago
 run.sh	Update run.sh	20 minutes ago

 README.md

Jenkins-Jira-integration

Sample shell script which is creating some folder structure there and providing permissions to it.

master

jenkins-jira-integration / run.sh

Find file

Blame

History

Permalink

 Update run.sh

Administrator authored 20 minutes ago

27090c26



run.sh 379 Bytes



EditWeb IDEReplaceDelete

1 echo "Welcome to this world"

2 echo "=====

3 echo "Creating directory and listing down the created folder structure with permissions set"

4 echo "=====

5 mkdir -p jira-webhook/builds/test

6 chmod -R 777 jira-webhook/*

7 ls -ltr *

Automated it using Jenkinsfile as below:

```
timestamps {

node () {

    stage ('jira_jenkins_integration - Checkout') {
        checkout([$class: 'GitSCM', branches: [[name: '*/master']],
doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg:
[], userRemoteConfigs: [[credentialsId: 'gitlab', url:
'http://novartis.devops.altimetrik.io:9080/root/jenkins-jira-integration
' ]]])
    }
    stage ('jira_jenkins_integration - Build') {

// Shell build step
sh """
echo "This job will be triggered when there is change in JIRA ticket
status"
sh ./run.sh
    """
    }
}
}
```

References docs and links :

<https://www.youtube.com/watch?v=DktKQZVLhtE>

https://medium.com/@shrut_terminator/devops-usecase-jira-jenkins-integration-4051413446a9