# Integration with Automation for Jira

- Overview
- Concepts
- Usage Examples
  - Jenkins
    - Trigger a Jenkins project build from an issue
      - Jenkins configuration
      - Automation for Jira configuration
    - Trigger a Jenkins project build from a Test Plan and report the results back to it
      - Jenkins configuration
      - Automation for Jira configuration
    - Azure DevOps
      - Trigger a Azure DevOps pipeline from a Test Plan and report the results back to it
        - Azure DevOps configuration
        - Automation for Jira configuration
    - Travis Cl
      - Trigger a TravisCI project build from a Test Plan and report the results back to it
        - TravisCl configuration
        - Automation configuration
    - Generic automation of processes
       Copy fields from requirement/Story to T
      - Copy fields from requirement/Story to Test whenever creating a Test or linking it to a story
        - Automation configuration
        - Reopen/transitionTests linked to a requirement whenever the requirement is transitioned or changed
           Automation configuration
- References

## Overview

The Automation for Jira app/feature enables users to easily extend and implement automation in Jira without having to code.

This way, users can implement rules that are triggered upon some event, executed if certain condition(s) are met and that perform certain action(s).

Rules can also be triggered manually or may be scheduled.

#### Please note

Automation for Jira has become part of Jira Datacenter offering, enabling automation of flows for all Jira users. Please check Atlassian's Jira Automation documentation for more info.

Since Xray uses issue types for most of its entities and since Xray provides many JQL functions that allow you to obtain testing-related information, Automation for Jira can be used with Xray in a very straightforward way.

Automation rules are available and, can be created, from the project settings, namely from the "Automation" tab.

<del>,</del>	Project settings								4
	Summary Details Re-Index project Delete project	Automation	Namo *		Global administration Owner -	Filter rules	Create n Project:	ite știt Enabled :	
	Issue types Acceptance Criterion Bag Capatity Epis Epis Provoment Instance New Feature New Feature New Contion Requirement Requirement Requirement Requirement Requirement Versions Versions Versions Versions Versions Components Users and roles Pervents	<ul> <li>Al rules</li> <li>Project rules</li> <li>Circler rules</li> <li>Circler rules</li> <li>Add label</li> </ul>	Troger Junkins job end link to Teel Plan e		💿 Adm	inistrator	Calculator (CALC)		
	Notifications HipOhat Integration Project automation								

#### Please note (i)

The following examples are provided as-is, no warranties attached; use them carefully.

Please feel free to adapt them to your needs.

Note: We don't provide support for Automaton for Jira; if you have doubts concerning its usage, please contact Automation for Jira's support.

### Concepts

Jira Automation allows project administrators to implement rules that can make certain processes automated, guaranteeing efficiency and consistency.

The main concepts of Jira Automation follow a very simple approach for defining an automation rule: if a certain "thing" happens (trigger) and certain cond itions are met, then execute one or more actions.

- Trigger: Triggers start the execution of a rule. Triggers can listen for events or be scheduled to run.
  - manual
  - upon field or workflow status changes
  - upon releasing
  - ° periodic
  - ° ...
- · Condition: Actions will only execute if all conditions preceding them pass.
  - ° "If" statement ° Issue fields condition
  - ° ...
- Action: Actions perform changes to a system.
  - change fields on issues
  - transition issues
  - ° web request
  - ° log
  - ° ...

It's also possible to run actions on issues that are related to the issue that triggered the rule, using "branches".



Automation rules run asynchronously. Some actions can run in parallel but usually they're sequential. There is no interaction with the user (except if the trigger was set off manually from Jira's UI).

# Usage Examples

### Jenkins

### Trigger a Jenkins project build from an issue

In this very simple scenario, we'll implement a rule, triggered manually, that will trigger a Jenkins project/job. The action will be available from within the "More" menu, in all issues of the selected project.

We're assuming that:

- you just want to trigger a Cl job, period; this job may be totally unrelated to the issue from where you triggered it
  what the Cl job will do, including if it will report the results back to Xray or not, is not relevant

#### Jenkins configuration

In Jenkins, we need to generate an API token for some user, which can be done from the profile settings page.

🚱 Jenkins		
Jenkins 🕨 admin		
1 People	Full Name	admin
🔍 Status	Description	
📄 Builds		
🐡 Configure		
🌲 My Views		
条 Credentials	API Token	
	User ID	admin
	API Token	fa02840152aa2e4da3d8db933ec708d6

At the project level, we need to enable remote build triggers, so we can obtain an "authentication token" to be used in the HTTP request afterwards.

Build Triggers						
Trigger builds remotely (e.g., from scripts)						
Authentication Token	iFBDOBhNhaxL4T9ass93HRXun2JF161Z					
	Use the following URL to trigger build remotely: JENKINS_URL/job/java-junit-calc-triggered/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.					

The project itself is a normal one, without parameters.

General Source C	Code Management Build Trig	rs Build Environment Build Post-build Actions	
Description	creates a new Test Execut	with the results from 4 junit tests. The revision field is populated with the build $\theta$	
	[Plain text] Preview		
Discard old builds			
Strategy	Log Rotation		+
	Days to keep builds		
		I not empty, build records are only kept up to this number of days	
	Max # of builds to keep	3	
		f not empty, only up to this number of build records are kept	
			Advanced.
GitHub project			
This project is para	rmeterized		
Throttle builds			
Disable this project	t		
Execute concurren	t builds if necessary		
			Advanced
Source Code	Management		
None			

#### Automation for Jira configuration

1. create a new rule and define the "When" (i.e. when it to should be triggered ), to be "Manually triggered"

Au	tomation DRAFT		Return	n te
Trigger Jenkins job Rule details		Rule details		
(1)	Rule details	Name*	Trigger Jenkins job	
0	Audit log	Description	Trigger Jenkins job "java-junit-calc-local-git"	
8	When: Manually triggered	Projects	Calculator (CALC)	
	All logged in users can run rule.	Enabled	Projects can only be modified in the global administration.	
8	Then: Send webhook	Allow rule trigger	Check to allow other rule actions to trigger this rule. Only enable this if you need this rule to execute in response to another rule.	
	POST http://192.168.56.102:8081/job/java-	Notify on error	Don't notify	
	junit-calc-local-git/build? token=iFBDOBhNhaxL4T9ass93HRXun	Created	3 hours ago	
	2JF161Z	Owner	Administrator	
Ó	Add component		The owner will receive emails when the rule fails.	
		Updated	3 hours ago	
		Actor	Administrator	
			Actions defined in this rule will be performed by the user selected as the actor.           Save         Cancel	

2. define an action (i.e. the "Then") as "Send webhook" and configure it as follows

Automation ENABLED							
Trigger Jenkins job	ß Send webhook ⊫						
Rule details	This action will send a HTTP POST to the url specified below: Webhook URL*						
Audit log	http://192.168.56.102:8081/job/java-junit-calc-local-git/build?token=iFBDOBhNt						
When: Manually triggered     All logged in users can run rule.	Headers (optional)       Content-Type       application/json       Authorization       Basic YWRtaW46YWRtaW4=						
Then: Send webhook POST http://192.168.56.102:8081/job/java- junit-calc-local-git/build? token=iFBDOBhNhaxL4T9ass93HRXun 2JF161Z	Add HTTP method POST Webhook body Empty Save Cancel						
Add component							

the Webhook URL provided above follows this syntax:
 <jenkins\_base\_url>/job/<name\_of\_jenkins\_project\_job>/build?token=<token>

• besides the "Content-Type" header that should be "application/json", define also an "Authorization" header having the value "Basic <auth>", where the base64 encoded <auth> can be generated using your Jenkins API credentials

After publishing the rule, you can go to the screen of an issue and trigger the Jenkins project/job.

<b>III X</b>	🗛 Dashboards <del>-</del> Projects - Issues - Boards	→ Structure → Power Apps Db
<b>5</b>	Calculator / CALC-3214 all my sum related tests fo	r v3.0
•••		
_	Zedit Comment	More - Stop Progress R
f	Details	Trigger Bamboo Build w
	Type: 🗾 Test Plan	Trigger Jenkins Build
<u>b_</u>	Priority: 🔦 Major	Synchronize Tests from
Πα	Affects Version/s: None	Assign
UUa	Labels: None	Log work
PE	Sprint: Sprint 1	Agilo Roard
23	Test Count: 7	Agile Board
(=)		Rank to Bottom
6	Description	Attach files
	Risks/sensible areas to cover:	Voters
efe	addition operation in basic mode	Stop watching
T_	addition operation in scientific mode	Watchers
Þ	Tests	Create sub-task
		Convert to sub-task
$\odot$	F lest Plan Board	Move
		Link
	Overall Execution Status	Clone
		Labels
	PASS	Delete
		Trigger Jenkins job

#### Trigger a Jenkins project build from a Test Plan and report the results back to it

In this simple scenario, we'll implement a rule, triggered manually, that will trigger a Jenkins project/job. The action will be available from within the "More" menu, for all Test Plan issues of the selected project.

We're assuming that:

- you just want to trigger a CI job, period; this job may be totally unrelated to the issue from where you triggered it
- the results will be submitted back to Xray, if the project is configured to do so in Jenkins

#### Jenkins configuration

In Jenkins, we need to generate an API token for some user, which can be done from the profile settings page.

没 Jenkins		
Jenkins 🕨 admin		
1 People	Full Name	admin
🔍 Status	Description	
📄 Builds		
🐡 Configure		
鵗 My Views		
A Credentials	API Token	
	User ID	admin
	API Token	fa02840152aa2e4da3d8db933ec708d6

At the project level, we need to enable remote build triggers, so we can obtain an "authentication token" to be used in the HTTP request afterwards.

Build Triggers							
Trigger builds remotely (e.g., from scripts)		0					
Authentication Token	iFBDOBhNhaxL4T9ass93HRXun2JF161Z						
	Use the following URL to trigger build remotely: JENKINS_URL/job/java-junit-calc-triggered/build?token=TOKEN_NAME or /buildWithParameters?token=TOKEN_NAME Optionally append &cause=Cause+Text to provide text that will be included in the recorded build cause.						

The project itself is a normal one; the only thing relevant to mention is that this project is a parameterized one, so it receives TESTPLAN, that in our case will be coming from Jira.

🎡 Jenkins			3	Qsearch
Jenkins 🕑 java-junit-calc-local-git-report-to-testplan 💛				
Genera	Source Code	e Management Build Trig	gers Build Environment Build Post-build Actions	
Der	scription	creates a new Test Execu	ion with the results from 4 junit tests. The revision field is populated with the build $\#$	
	ard old builde	[Plain text] <u>Preview</u>		
Str	ategy	Log Potation		
		Log Hotation		
		Days to keep builds		
			if not empty, build records are only kept up to this number of days	
		Max # of builds to keep	3	
			if not empty, only up to this number of build records are kept	Advanced
				Advanced
Gitl-	lub project			
<ul> <li>This</li> </ul>	project is parame	terized		X
		String Parameter		•
		Name TES	TPLAN	
		Default Value		0
		Description		•
		Plain	text] Preview	
		⊛ Tr	im the string	0
L		Add Parameter 👻		
		[Plain ⊘ Tr Add Parameter ▼	text] Preview im the string	<u>//</u>

#### Automation for Jira configuration

1. create a new rule and define the "When" (i.e. when it to should be triggered ), to be "Manually triggered"

Au	tomation		
Trigger Jenkins job and link to Test Plan		Rule details	Trigger Jenkins job and link to Test Plan
()	Rule details	Description	Trigger Jenkins job "java-junit-calo"
	Audit log When: Manually triggered All logged in users can run rule.	Projects Enabled	Calculator (CALC)         Projects can only be modified in the global administration.
×	If: Compare two values Checks if: {{issue.issuetype.name}} equals Test Plan	Allow rule trigger Notify on error Created Owner	<ul> <li>Check to allow other rule actions to trigger this rule. Only enable this if you need this rule to execute in response to another rule.</li> <li>Don't notify          <ul> <li>a hours ago</li> <li>Administrator</li> <li>The owner will receive emails when the rule fails.</li> </ul> </li> </ul>
8	Then: Send webhook POST http://192.168.56.102:8081/job/java- junit-calc-local-git-report-to- testplan/buildWithParameters? token=iFBDOBNNhaxL4T9ass93HRXun 2JF161Z&TESTPLAN={{issue.key}}	Updated Actor	3 hours ago          Administrator         Actions defined in this rule will be performed by the user selected as the actor.         Save       Cancel
¢	Add component		

2. define the condition so that this rule can only be executed from Test Plan issue

Αι	Automation ENABLED						
Trig Pla	gger Jenkins job and link to Test n	😕 Compare condition 🝵					
Rule details		Compares a value to another using value substitutions and regular expressions. First value*					
	Audit log	{{issue.issuetype.name}}					
	When: Manually triggered	Condition Equals \$ Second value					
~	If: Compare two values	Test Plan Save Cancel					
	Checks if: {{issue.issuetype.name}} equals Test Plan	> What values can I compare?					
8	Then: Send webhook POST http://192.168.56.102:8081/job/java- junit-calc-local-git-report-to- testplan/buildWithParameters? token=IFBDOBINIhaxL4T9ass93HRXun 2JF161Z&TESTPLAN={{issue.key}}						
Ċ	Add component						

3. define an action (i.e. the "Then") as "Send webhook" and configure it as follows

Αι	Itomation ENABLED					
Trigger Jenkins job and link to Test Plan		(	🔏 Se	nd webhook 🍵		
	Rule details		This action	on will send a HTTP POST	Γ to the url specified below:	
			s?token	=iFBDOBhNhaxL4T9ass9	3HRXun2JF161Z&TESTPLAN={{is	sue.key}}
9	, duit log	-	Headers	(optional)		
*	When: Manually triggered		Content	-Туре	application/json	Î
	All logged in users can run rule.		Authorization		Basic YWRtaW46YWRtaW4=	] <del>_</del>
			Add			
2	If: Compare two values		HTTP me	əthod		
	Checks if: {{issue.issuetype.name}} equals Test Plan		POST		\$	
			Webhool	< body		
			Empty		\$	
8	Then: Send webhook		Save	Cancel		
	http://192.168.56.102:8081/job/java-					
	junit-calc-local-git-report-to- testplan/buildWithParameters?					
	Loken=iFBDOBNNaxL4T9ass93HRXun 2JF161Z&TESTPLAN={{issue.key}}					
0	Add component					

- the Webhook URL provided above follows this syntax:

   <jenkins\_base\_url>/job/<name\_of\_jenkins\_project\_job>/buildWithParameters?token=<token>&TESTPLAN={{issue.key}}

   besides the "Content-Type" header that should be "application/json", define also an "Authorization" header having the value "Basic <auth>, where the base64 encoded <auth> can be generated using your Jenkins API credentials

After publishing the rule, you can go to the screen of an issue and trigger the Jenkins project/job.

	P Edit Comment	More -	Stop Progress	Resolve Issue	Close Issue	Admin 🗸	
	Details         Type:       Image: Test Plan         Priority:       Major         Affects Version/s:       None         Component/s:       None         Labels:       None         Sprint:       Sprint 1         Test Count:       7         Description       Risks/sensible areas to cover:         •       addition operation in basic mode         •       addition operation in scientific mode	Trigger Ba Trigger Jei Synchroniz Assign Log work Agile Boar Rank to To Rank to To Rank to Bo Attach files Voters Stop watch Watchers	mboo Build w hkins Build hkins build ze Tests from d p ottom s hing	Status: Resolution: Fix Version/s:		U U V2	PROGRESS presolved
0	F Test Plan Board	Convert to Move Link	sub-task				
	7 PASS	Clone Labels Delete					
	TOTAL TESTS: 7	Trigger Jer Trigger Jer	nkins job nkins job an			_	7

### Azure DevOps

### Trigger a Azure DevOps pipeline from a Test Plan and report the results back to it

#### Azure DevOps configuration

We need to create a service connection, using the "incoming webhook" template, so that we can use Azure DevOps API later on.

¢	sergiofreire / NUNITDEMO /	Settings / Service connections	Edit service connection
N +	Project Settings	< MyWebhookConnection	
_		Overview Usage history	Authentication
	General		WebHook Name
	B Overview	Details	MyWebhookTrigger
	<sup>ሳ</sup> ጜ Teams	Service connection type Cre	Secret (ontional)
8	Permissions	Incoming WebHook using no authentication	*******
2	Notifications		Optional secret for the webhook. WebHook service will use this secret to calculate the newload checksum
	5 Service hooks		lite Joy de (notional)
-	I Dashboards		
Δ.	Boards		Http header name on which checksum will be sent
	Project configuration		
	🔏 Team configuration		Details
	O GitHub connections		Service connection name
	Pinelines		MyWebhookConnection
			Description (optional)
	Address Addres		
	Parallel jobs     Octoberge		
	Tost management		Security
	Release retention		Grant access permission to all pipelines
	o <sup>o</sup> Service connections		
	AML build services		
_	кероз		
۲	Repositories		
»	Artifacts		Troubleshoot Cancel Save

Create a Personal Access Token (PAT), so you can use it as the password in API requests, along with the "organization name" as username.

Azure DevOps					𝒫 Search		SF
User settings Sérgio Freire	Personal Access Tokens These can be used instead of a passwo	rd for applications like Git or can be par	ssed in the authorization header to access REST A	PIs		Preview features	
Account	+ New Token / Revoke / Ed	it () Regenerate		A second sectors		Solution Time and Locale	~
9E Destile	Token name 👃		Status	Organization	Expires or	O Permissions	
Ch Time and Lease	mytoken1 Full access		Active	sergiofreire	16/05/20	Notifications	
Permissions	mytoken?					Theme	
	Build (Read & execute); Code (Full); Co	de (Read & write); Packaging (Read, w	Active	sergiofreire	05/05/20	bh Usage	
Preterences						& Personal access tokens	
Notifications						SSH public keys	
© Theme						<ul> <li>Alternate credentials</li> </ul>	
Ub Usage							
Security							
𝒫 Personal access tokens							
N SSH public keys							
<ul> <li>Alternate credentials</li> </ul>							
Authorizations							
Create a new perso	onal access token	×					
Name							
mytoken							
Organization							
org sergiofreire		~					
Expiration (UTC)							
30 days	~	24/06/2021					
ser Scopes Authorize the scope of acc Scopes Full access Custom define	cess associated with this token ed						
Work items, queries, back	logs, plans, and metadata						
C Read C Read	& write 📄 Read, write, & manag	je					
Code							
Source code, repositories	, pull requests, and notifications						
Read write	manage	Full Status					
Build Artifacts, definitions, requ	uests, queue a build, and updated build p	roperties					
🕑 Read 🕑 Read	& execute						
Palaasa							
Read, update, and delete	releases, release pipelines, and stages						
Read Read	, write, & execute 🛛 🗌 Read, write	, execute, & manage					
Test Management Read, create, and updated	d test plans, cases, and results						
Read Read	& write						
Show all scopes (28 more	)						
Create Cancel							
Cancel							

Then, in your Azure DevOps repository containing the project's code and tests, create a pipeline /azure-pipelines.yml; this pipeline will be triggered using Azure DevOps API.

In the following example, the pipeline will receive the Test Plan issue key as an input parameter. It will then run the build, including the automated tests, and in the end it will report the results back to Xray using "curl" utility.

We need to define a resources section, that contains a reference to the webhook configured earlier.

```
/azure-pipelines.yml
parameters:
- name: "testplan"
  type: string
 default: ""
trigger:
- main
resources:
  webhooks:
    - webhook: "MyWebhookTrigger"
                                              ### Webhook alias
     connection: "MyWebhookConnection"
                                             ### Incoming webhook service connection
pool:
  vmImage: ubuntu-latest
steps:
- bash: |
   echo ${{ parameters.testplan }}
  displayName: '(debug) print testplan parameter'
- script: dotnet restore
 displayName: 'install build dependencies'
- script:
   dotnet test -s nunit.runsettings
  displayName: 'Run tests'
- bash:
    set -x
    curl -o - -H "Content-Type: multipart/form-data" -u '$(jira_user):$(jira_password)' -F "file=@./bin/Debug
/net5.0/TestResults/nunit_webdriver_tests.xml" "$(jira_server_url)/rest/raven/2.0/import/execution/nunit?
projectKey=$(project_key)&testPlanKey=${TESTPLAN}"
  displayName: 'Import results to Xray server'
```

Xray endpoint's base URL and the API key credentials (i.e. client id + client secret) are defined in Azure DevOps as variables. These may be marked as secret.

Azure DevOps	sergiofreire / NUNITDEMO / Pipelines / Builds / NUNITDEMO / YAML	earch	≣ 🖞 Ø Å SF
NUNITDEMO +		Va	ariables Run
🛃 Overview			
💐 Boards	<sup>8</sup> / <sub>2</sub> main ∨ ♦ NUNITDEMO / azure-pipelines.yml	Tasks	
😢 Repos	1 parameters: 2 - name: "testplan" 3 - type:string		
Pipelines	4 - default:"" 5	dotnet .NET Core Build, test, pa	ackage, or publish a dotnet applic
Pipelines	6 trigger: 7main	Android sig	ning n Android APK files
Environments	8 9 resources:	Ant	
🔊 Releases	10 •webhooks: 11 •••••webhook:"NvWebhookTrigger" ### Webhook alias	Build with Ap	ache Ant
II∕\ Library	12	App Center Distribute ap	distribute p builds to testers and users via
E Task groups	14 pool: 15 ···vmImage:·ubuntu-latest	App Center	test
Deployment groups	16	Test app pac	kages with Visual Studio App Cen
👗 Test Plans	1/ steps: 18 19 - bash: I	Archive file Compress file	s es into .7z, .tar.gz, or .zip
Artifacts	<pre>20 echo \${{ parameters.testplan }} 21 displayName.'(debug) print testplan parameter' 22</pre>	ARM templa Deploy an Az	ate deployment ure Resource Manager (ARM) te

Variables	×
♀ Search variables	+
<i>fx</i> <b>jira_password</b> = !q"w#€	
<i>fx</i> <b>jira_server_url</b> = https://sandbox.xpand-it.com	
<i>fx</i> <b>jira_user</b> = sfreire	
<i>fx</i> <b>project_key</b> = CALC	

#### Automation for Jira configuration

1. create a new rule and define the "When" (i.e. when it to should be triggered ), to be "Manually triggered"



2. define the condition so that this rule can only be executed from Test Plan issue

utomation <b>ENABLED</b>	
igger Azure DevOps pipeline for is Test Plan	Issue fields condition 🍵
	Check whether an issue's field meets a certain criteria
Rule details	Field*
Audit log	Issue Type 🗸
Addit log	Condition*
When: Manually triggered	equals 🗸
All logged in users can run rule.	Value Field
If: Issue Type equals	🔁 Test Plan
Test Plan	
utomation ENABLED	Send web request
5 Test Fian	
	This action will send a HTTP request to the url specified below:
Rule details	This action will send a HTTP request to the url specified below: Webhook URL*
Rule details Audit log	This action will send a HTTP request to the url specified below: Webhook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.
Rule details       Audit log	This action will send a HTTP request to the url specified below: Webhook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {{value.urlEncode}}. Headers (optional)
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.	This action will send a HTTP request to the url specified below: Webhook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {{value.urlEncode}}. Headers (optional) Content-Type
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.	This action will send a HTTP request to the url specified below:         Webhook URL*         https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni         Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.         Headers (optional)         Content-Type         Authorization         Basic c2VyZ2IvLmZyZWlyZUB4c
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals	This action will send a HTTP request to the url specified below:         Webhook URL*         https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni         Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.         Headers (optional)         Content-Type         Authorization         Basic c2VyZ2lvLmZyZWlyZUB4c         Add
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan	This action will send a HTTP request to the url specified below:         Webbook URL*         https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni         Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.         Headers (optional)         Content-Type         Authorization         Basic c2VyZ2lvLmZyZWlyZUB4c         Add         HTTP method
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan	This action will send a HTTP request to the url specified below: Webhook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {{value.urlEncode}}. Headers (optional) Content-Type application/json = Authorization Basic c2VyZ2lvLmZyZWlyZUB4c = Add HTTP method POST
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan	This action will send a HTTP request to the url specified below: Webhook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {{value.urlEncode}}. Headers (optional) Content-Type application/json = Authorization Basic c2VyZ2lvLmZyZWlyZUB4c = Add HTTP method POST Webhook body
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan         Then: Send web request         POST	This action will send a HTTP request to the url specified below:   Webhook URL*   https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni   Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.   Headers (optional)   Content-Type   application/json   Authorization   Basic c2VyZ2lvLmZyZWlyZUB4c   Add   HTTP method   POST   Webhook body   Custom data
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan         Then: Send web request         POST         https://dev.azure.com/sergiofreire/NUNIT         DEMO(_apig/build/build/2)	This action will send a HTTP request to the url specified below:   Webbook URL*   https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni   Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.   Headers (optional)   Content-Type   application/json   Authorization   Basic c2VyZ2lvLmZyZWlyZUB4c   Add   HTTP method   POST   Webhook body   Custom data   Wait for response
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan         Then: Send web request         POST         https://dev.azure.com/sergiofreire/NUNIT         DEMO/_apis/build/builds?         ignoreWarnings=true&api-version=6.0	This action will send a HTTP request to the url specified below: Webbook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {(value.urlEncode}). Headers (optional) Content-Type application/json for the second s
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan         Then: Send web request         POST         https://dev.azure.com/sergiofreire/NUNIT         DEMO/_apis/build/builds?         ignoreWarnings=true&api-version=6.0	This action will send a HTTP request to the url specified below: Webbook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {(value.urlEncode)}. Headers (optional) Content-Type application/json • Authorization Basic c2VyZ2lvLmZyZWlyZUB4c • Add HTTP method POST • Webhook body Custom data • Walt for response Delay execution of subsequent rule actions until we've received a response for this webhook Custom data*
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan         Then: Send web request         POST         https://dev.azure.com/sergiofreire/NUNIT         DEMO/_apis/build/builds?         ignore/Warnings=true&api-version=6.0         Add component	This action will send a HTTP request to the url specified below: Webbook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {(value.urlEncode)}. Headers (optional) Content-Type application/json = Authorization Basic c2VyZ2lvLmZyZWlyZUB4c = Add HTTP method POST Webhook body Custom data Wati for response Delay execution of subsequent rule actions until we've received a response for this webhook Custom data* { ''arameters': "{ ''testplan\": !''{(issue.key})'' }", ''definition": { ''testplan\": !''{(issue.key})'' }",
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan         Then: Send web request         POST         https://dev.azure.com/sergiofreire/NUNIT         DEMO/_apis/build/builds?         ignore/Warnings=true&api-version=6.0         Add component	This action will send a HTTP request to the url specified below: Webhook URL* Inttps://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {(value.urlEncode)}. Headers (optional) Content-Type application/json Authorization Basic c2VyZ2lvLmZyZWlyZUB4c Add HTTP method POST Webhook body Custom data Wait for response Delay execution of subsequent rule actions until we've received a response for this webhook Custom data* { "parameters": "{ ''testplan\'': \'{{issue.key}}\'' }", "definition": { "id": 3 }
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan         Then: Send web request         POST         https://dev.azure.com/sergiofreire/NUNIT         DEMO/_apis/build/builds?         ignoreWarnings=true&api-version=6.0         Add component	This action will send a HTTP request to the url specified below: Webbook URL* https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWarni Request parameters must be url encoded, smart values should use: {(value.urlEncode}). Headers (optional) Content-Type application/json for application Basic c2VyZ2lvLmZyZWlyZUB4c for Add HTTP method POST V Webhook body Custom data V Wait for response Delay execution of subsequent rule actions until we've received a response for this webhook Custom data* $\begin{cases}                                    $
Rule details         Audit log         When: Manually triggered         All logged in users can run rule.         If: Issue Type equals         Test Plan         Then: Send web request         POST         https://dev.azure.com/sergiofreire/NUNIT         DEMO/_apis/build/builds?         ignore/Warnings=true&api-version=6.0         Add component	<pre>This action will send a HTTP request to the url specified below: Vebbook URL*  https://dev.azure.com/sergiofreire/NUNITDEMO/_apis/build/builds?ignoreWami Request parameters must be url encoded, smart values should use: {\value.urlEncode}. Headers (optional)  Content-Type application/json Authorization Basic c2VyZ2lvLmZyZWlyZUB4c Add HTTP method POST Vebhook body Custom data Valt for response Delay execution of subsequent rule actions until we've received a response for this webhook Custom data*  {     parameters": "{ \'testplan\': \'{{issue.key}}\'' ",     "idfinition": {     "Idfinition: {     "Idfinition": {     Idfinition: {</pre>

- the web request URL provided above is from Azure DevOps API, for queueing builds, and follows this syntax:

  https://dev.azure.com/<organization\_name>/<project>/\_apis/build/builds?ignoreWarnings=true&api-version=6.0

  authentication is done using the organization name plus the personal access token, created earlier in Azure DevOps, as the login:password pair used to calculate the Base64 content of the Authorization header
  the "Content-Type" header should be "application/json"
  the HTTP POST body content, defined in the "Custom data" field, will be used to identify the build definition and also the original Test Plan issue login:

key;

```
custom data (i.e. HTTP body content)
```

```
{
    "parameters": "{ \"testplan\": \"{{issue.key}}\" }",
    "definition": {
        "id": 3
        }
}
```

Note: to find the definition id, you can click on the pipeline in Azure DevOps and its id is shown as part of the URL

← → C ( a dev.azure.com/se	rgiofreire/NUNITDEMO/_buik ?definitionId=3		*
Azure DevOps	sergiofreire / NUNITDEMO / Pipelines / NUNITDEMO		🔎 Search 🚝 🗂 💿 🙈 💶
NUNITDEMO +			Edit Run pipeline
Overview	Runs Branches Analytics		
Boards	Description	Stages	
😰 Repos	#20210525.12 Update azure-pipelines.yml for Azure Pipelines	0	B Yesterday
Pipelines	R Manually triggered for 🕑 P main 🕴 71eed16 🖉	•	⊙ 46s ·
kini Pipelines	#20210525.11 Update azure-pipelines.yml for Azure Pipelines A Manually triggered for Strain $2$ Theorem 7.1	0	C3 Yesterday ⓒ 51s
Environments	#20210525.10 Update azure-pipelines.yml for Azure Pipelines	•	[] Yesterday
∯ Releases	🖉 Individual Ci for 🕘 🎾 main 🕴 71eed16 🖈	0	() 63s
II\ Library	#20210525.9 Update azure-pipelines.yml for Azure Pipelines ⊘ Individual CI for ● 3º main ♦ 78135b7	0	[3] Yesterday ⊙: 46s
AND - I			

After publishing the rule, you can go to the screen of an issue and trigger a pipeline run in Azure DevOps.

CA Te	est Plan for	curren	t sprint							
🖋 Edit	Q Comment	Assign	More 🗸	To Do	In Pro	ogress	Done	Admin 🗸		
Details     Type:     Priority:     Labels:		Test Plan Trivial one	Log work Agile Board Rank to To	þ		5	Status: Resolution:		TO DO (View Workflow) Unresolved	
Descript     Click to a	Description     Click to add description			ottom s						
✓ Tests	est Plan Board		Attach Scr Voters Stop watch	eenshot						+ Add ~
This test	plan is not assoc	viated with t	Watchers Create sub Convert to	o-task sub-task	<					Add Test Executions
This test	plan is not assoc ents	ciated with 1	Move Link Clone							
			Labels Delete			s to att	ach, or bro	wse.		
> Structur	e		trigger Azı	ure DevO	ps p					0

Azure DevOps	sergiofreire / NUNITDEMO / Pipelines /	NUNITDEMO / 20210525.12	P Search I≣ ☐ ⑦ 冬 SF				
NUNITDEMO +	← Jobs in run #20210525	🔗 Run tests p					
Cverview	NUNITDEMO						
📑 Boards	Jobs	1 Starting: Run tests 2					
😰 Repos	✓ Ø Job 41s	3 Task : Command line 4 Description : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows 5 Vertice : 2 182 8					
Pipelines	Checkout NUNITDEM 3s	6 Author : Hicrosoft Corporation 7 Help : <u>https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line</u>					
Pipelines	<ul> <li>(debug) print testplan 1s</li> </ul>	8					
Environments	install build depende 12s	11 dotnet test -s nunit runsettings					
🖉 Releases	Run tests 17s	13 /usr/bin/bashnoprofilenorc /home/vsts/work_temp/60e5fbfe-b993-4b55-8ca5-dfb268e40leb.sh 14 Determining projects to restore					
III\ Library	(debug) list files in w <1s	15 All projects are up-to-date for restore. 16 /home/vsts/work/1/s/Google/PageObjects/HomePage.cs(18.29): warning CS0169: The field 'HomePage.el	em submit button' is never used [/home/vsts/work/1/s/nunit ·				
E Task groups	Import results to Xra <1s	17 /home/vsts/work/1/s/Webdem/PageObjects/LoginPage.cs(22,29): warning C50649: Field 'LoginPage.sut 18 /home/vsts/work/1/s/Goodle/PageObjects/HomePage.cs(13,29): warning C50649: Field 'HomePage.elem s	mitButtonElement' is never assigned to, and will always hav earch text' is never assigned to, and will always have its				
Deployment groups	Import results to Xray 4s	19 /home/vsts/work/1/s/Webdemo/PageObjects/LoginPage.cs(18,29): warning CS0649: Field 'LoginPage.pas 20 /home/vsts/work/1/s/Webdemo/PageObjects/LoginPage.cs(14,29): warning CS0649: Field 'LoginPage.use	swordElement' is never assigned to, and will always have it rnameElement' is never assigned to, and will always have it				
📥 Test Plans	Post-job: Checkout <1s	21 nunit_webdriver_tests -> /home/vsts/work/1/s/bin/Debug/net5.0/nunit_webdriver_tests.dll 22 Test run for /home/vsts/work/1/s/bin/Debug/net5.0/nunit webdriver tests.dll (.NETCoreApp.Version=	-v5.0)				
Artifacts	Finalize Job <1s	23 Microsoft (R) Test Execution Command Line Tool Version 16.9.4 24 Copyright (c) Microsoft Corporation. All rights reserved.					
	Report build status <1s	<ol> <li>Starting test execution, please wait</li> <li>A total of 1 test files matched the specified pattern.</li> </ol>					

In this case, since the pipeline was configured to report results back to Xray, a new Test Execution would be created and linked back to the source Test Plan where the automation was triggered from.

	CALC-18	esults - nı	unit_we	bdriver_tests	.xml -	[16220	26982605	]			
🖋 Edit	Q Comment	Assign M	Nore 👻 T	o Do In Progress	Done	Admin 👻					
✓ Details											
Type: Priority: Labels:		Test Executio Trivial Ione	n				Status: Resolution:	TO DO Unreso	(View Workflow) blved		
Test Plar Test Env	n: C ironments: N	Ione									
<ul> <li>Descript</li> </ul>	tion										
Executio	n results impor	ted from extern	al source								
✓ Tests											+ Add ~
Overall Ex	xecution Status										
<b>4</b> pas	S										
Total Test	ts: 4										
Ŧ	Filter(s)										
										Show 100 🗸 entries	Columns -
	A Rank	🖕 Кеу	🖕 Summa	ry	4	Test Type	#Req	#Def	Assignee	🖕 Status	
	1	CALC-13	BasicTex	tSearchNoPOM		Generic	0	0	Sergio Freire	PASS	• …
	2	CALC-11	BasicTex	tSearch		Generic	0	0	Sergio Freire	PASS	• …
	3	CALC-12	InvalidLo	igin		Generic	1	0	Sergio Freire	PASS	• …
	4	CALC-10	ValidLog	in		Generic	1	0	Sergio Freire	PASS	• ••
Showing	1 to 4 of 4 entrie	S								First Previous	1 Next Last

	CALC / CALC-15						
Y	Test Plan for current sprint						

🖋 Edit	Q Comment	Assign More 🖌	To Do	In Progress	Done	Admin ¥					
<ul> <li>Details</li> </ul>											
Type:		Test Plan					Status:	TO DO (View Wo	orkflow)		
Priority Labels:	No No	Trivial ne					Resolution:	Unresolved			
<ul> <li>Descrip</li> <li>Click to</li> </ul>	ption add description										
<ul> <li>Tests</li> </ul>									_		
- F 1	Fest Plan Board								+	Create Test Execution	~ + Add ~
Overall	Execution Status										
Л											
<b>4</b> ра	SS										
Total Te	sts: 4										
	⇒ Filter(s)										
	∓ Filter(s)							Show	v 10 🗸 entries	s All Environments	<ul> <li>Columns •</li> </ul>
	∓ Filter(s) Key	Summary			Requireme	nts	#Test Executions	Shov Issue Assig	v 10 🗸 entries	s All Environments Latest Status	- Columns -
. ·	Filter(s) Key CALC-13	Summary BasicTextSearch	hNoPOM		Requireme	nts	#Test Executions	Shov Issue Assig Xpand IT A	r (10 💙 entrie: inee .dmin	s All Environments Latest Status PASS	- Columns -
	Filter(s) Key CALC-13 CALC-11	Summary BasicTextSearch BasicTextSearch	hNoPOM h		Requireme	nts	#Test Executions	Shov Issue Assig Xpand IT A Xpand IT A	r 10 💙 entries Inee dmin dmin	s All Environments Latest Status PASS PASS	• Columns • 
	Filter(s) Key CALC-13 CALC-11 CALC-12	Summary BasicTextSearcl BasicTextSearcl InvalidLogin	hNoPOM h		Requireme CALC-2	nts	#Test Executions	Shov Issue Assig Xpand IT A Xpand IT A Xpand IT A	r 10 ♥ entries Inee dmin dmin dmin	All Environments Latest Status PASS PASS PASS	• Columns • 
	Key CALC-13 CALC-12 CALC-12 CALC-10	Summary BasicTextSearcl BasicTextSearcl InvalidLogin ValidLogin	hNoPOM h		Requireme CALC-2 CALC-2	nts	#Test Executions	Shov Issue Assig Xpand IT A Xpand IT A Xpand IT A Xpand IT A	r (10 v) entries nee dmin dmin dmin dmin	All Environments Latest Status PASS PASS PASS PASS PASS	• Columns •
E v	₩         ₩           Key         CALC-13           CALC-11         CALC-12           CALC-12         CALC-10           g 1 to 4 of 4 entries	Summary BasicTextSearcl BasicTextSearcl InvalidLogin ValidLogin	hNoPOM h		Requireme CALC-2 CALC-2	nts	#Test Executions	Shov Issue Assig Xpand IT A Xpand IT A Xpand IT A	e 10 entries nee dmin dmin dmin dmin	All Environments Latest Status PASS PASS PASS PASS First Previous	<ul> <li>Columns -</li> <li></li> <li></li> <li></li> <li>1 Next Last</li> </ul>
Showing	<pre>Key CALC-13 CALC-11 CALC-12 CALC-10 g 1 to 4 of 4 entries recutions</pre>	Summary BasicTextSearcl BasicTextSearcl Invalid.ogin Valid.ogin	hNoPOM h		Requireme CALC-2 CALC-2	nts	#Test Executions	Shov Issue Assig Xpand IT A Xpand IT A Xpand IT A	r (10 ♥) entries Innee Idmin Idmin Idmin Idmin	All Environments Latest Status PASS PASS PASS First Previous	<ul> <li>Columns -</li> <li></li> <li></li> <li></li> <li>1 Next Last</li> </ul>
Showing	<pre>Filter(s) Key CALC-13 CALC-11 CALC-12 CALC-10 g 1 to 4 of 4 entries recutions</pre>	Summary BasicTextSearci BasicTextSearci InvalidLogin ValidLogin	hNoPOM h		Requireme CALC-2 CALC-2	nts	#Test Executions	Shov Issue Assig Xpand IT A Xpand IT A Xpand IT A Xpand IT A	r (10 ) entries nee dmin dmin dmin dmin	All Environments Latest Status PASS PASS PASS PASS First Previous Add	- Columns -   1 Next Last rest Executions
Test Ex	<pre>Key CALC-13 CALC-13 CALC-11 CALC-12 CALC-12 CALC-10 g 1 to 4 of 4 entries recutions</pre>	Summary BasicTextSearci BasicTextSearci InvalidLogin ValidLogin	hNoPOM h		CALC-2 CALC-2	nts	#Test Executions	Shov Issue Assig Xpand IT A Xpand IT A Xpand IT A	r ( 0 • ) entrier nee dmin dmin dmin dmin	All Environments Latest Status PASS PASS PASS First Previous Add Show (10 v entries	Columns -     Columns -        Next Last     Columns -     Columns -
Showing ▼ Test Ex	Key Key CALC-13 CALC-13 CALC-11 CALC-12 CALC-12 CALC-10 g 1 to 4 of 4 entries secutions Key	Summary BasicTextSearci BasicTextSearci InvalidLogin ValidLogin	hNoPOM		CALC-2 CALC-2	nts	#Test Executions	Shor Issue Assig Xpand IT A Xpand IT A Xpand IT A	r ( ) v entrier nee dmin dmin dmin dmin	All Environments Latest Status  PASS  PASS  PASS  First Previous  Add  Show (10 v entries  ret Status	Columns -     Columns -              Columns -
■ v Showing v Test Ex	Key CALC-13 CALC-13 CALC-11 CALC-12 CALC-12 CALC-10 g 1 to 4 of 4 entries recutions € Key CALC-18	Summary BasicTextSearci BasicTextSearci InvalidLogin ValidLogin ValidLogin	hNoPOM h	iver_tests.xml	Requireme CALC-2 CALC-2	nts	#Test Executions	Shor Issue Asig Xpand IT A Xpand IT A Xpand IT A Xpand IT A Xpand IT A	r (10 v) entrier nee dmin dmin dmin dmin dmin dmin	All Environments Latest Status ASS ASS First Previous Show (10 ) entries Status	Columns -     Columns -           Columns -

### Travis CI

### Trigger a TravisCI project build from a Test Plan and report the results back to it

In this simple scenario, we'll implement a rule, triggered manually, that will trigger a TravisCI project/job. The action will be available from the "Automation" panel, for all Test Plan issues of the selected project.

We're assuming that:

- you just want to trigger a CI job, period; this job may be totally unrelated to the issue from where you triggered it
  the results will be submitted back to Xray, if the project is configured to do so in TravisCI

#### **TravisCl configuration**

In TravisCI, we need to generate an API authentication token for some user, which can be done from the My Account settings page.

Travis Cl	gelog Documentation Help
MY ACCOUNT Cristiano Morais da Cunha	Cristiano Morais da Cu @CMCunha
C Sync account	Repositories Settings Plan Migrate Plan usage
xray App	API authentication To learn more about using our API, please head to <u>developer.travis-ci.com</u> .
MISSING AN ORGANIZATION? Review and add your authorized organizations.	Token •••••• • view token ••• • view token

Once we have the authentication token we followed the TravisCI API documentation to configure the following steps on the Jira side.

For the Travis CI the important change we must do is in the YAML file that will configure Travis CI pipeline, we use the following configuration to achieve that:

.travis.yml
sudo: false
language: java
jdk:
- openjdk8
cache:
directories:
- "\$HOME/.cache"
jobs:
include:
- stage: test and report to Xray
script:
-
echo "building repo"
mvn clean compile testfile pom.xml
curl -H "Content-Type: multipart/form-data" -X POST -u \$USERNAME:\$PASSWORD -F "file=@target
/surefire-reports/TEST-com.xpand.java.CalcTest.xml" "https://\$JIRASERVER/rest/raven/2.0/import/execution/junit?
projectKey=\$PROJECTKEY&testPlanKey=\$TESTPLAN"
echo "done"

#### For more details about this configuration please check the TravisCI tutorial documentation.

As you can see we are pushing results back to Xray with the last curl command:

curl command
curl -H "Content-Type: multipart/form-data" -X POST -u \$USERNAME:\$PASSWORD -F "file=@target/surefire-reports /TEST-com.xpand.java.CalcTest.xml" "https://\$JIRASERVER/rest/raven/2.0/import/execution/junit? projectKey=\$PROJECTKEY&testPlanKey=\$TESTPLAN"

On this command we are passing the project key in order to report back to a specific Project on the Xray side. Further ahead we will show how it is populated.

- PROJECTKEY The key that identifies the project on the Jira side.
- TESTPLAN The Test Plan key used to identify the Test Plan to associate the execution with.

Once we have the authentication token, we follow the TravisCI API documentation to configure the following steps on the Jira side.

#### Automation configuration

On the Jira side we will use the Automation capabilities that it provides out of the box, so within the administration area go to the automation entry in the system settings and:

1. create a new rule and define the "When" (i.e. when it to should be triggered), to be "Manually triggered"

Automation	ENABLED		
Trigger Travis Cl		🚯 Manual trigger 🥒	
Rule details		Rule is run when it is manually triggered by the user from an issue. Groups that can run trigger	
Audit log		All logged in users	~
🚯 When: Manually tr	ggered	Can	cel Save
All logged in users can	:un rule.		

2.Define a condition, in our case we will define that only Test Plan issue types will be allowed to trigger this pipeline, this is achieved with the following condition:

Automation		
Trigger Travis Cl	🔀 Issue fields condition 🝵	
<ul><li>6 Rule details</li><li>a Audit log</li></ul>	Check whether an issue's field meets a certain criteria Field* Issue Type	
When: Manually triggered All logged in users can run rule.	Condition* equals 🗸	
C If: Issue Type equals	Value Field	
	Cancel 5	Sav

3. define an action (i.e. the "Then") as "Send webhook" and configure it as follows

Trigger Travis CI    Rule details     Audit log     When: Manually triggered All logged in users can run rule.     The: Send web request POST https://apl.travis- c. ordered.travis.     POST https://apl.travis- c. ordered.travis.     Add component                    The: Send web request POST https://apl.travis- c. ordered.travis.     Other than the intervence of the webroe body      Other than the intervence of the webroe body     Other than the intervence of the webroe body     Other than the intervence of the webroe body     Other than the intervence of the webroe body               <	utomation DRAFT	
<ul> <li>Rule details</li> <li>Audit log</li> <li>Audit log</li> <li>When: Manually triggered</li> <li>All loged in users can run rule.</li> <li>If: Issue Type equals</li> <li>Thes: Send web request</li> <li>POST https://apl.travis- d.com/repo/Xray-App%ZPtubrial-java- junit-travisol/requests</li> <li>Add component</li> <li>The details and antitication of subsequent rule actions until we've received a response for this webhoo Eustem data"</li> <li>(intermediation of subsequent rule actions until we've received a response for this webhoo Eustem data"</li> </ul>	igger Travis Cl	🔥 Send web request 🍵
<ul> <li>Audit log</li> <li>Audit log</li> <li>Autionally triggered</li> <li>All logged in users can run rule.</li> <li>If: Issue Type equals</li> <li>Test Plan</li> <li>Then: Send web request</li> <li>POST https://api.travis- d.com/repo/Xray-App%2/Flutorial-java- junit-travisci/requests</li> <li>Add component</li> <li>Add component</li> </ul>	Rule details	This action will send a HTTP request to the un specified below: Webhook URL*
<ul> <li>When: Manually triggered All logged in users can run rule.</li> <li>If: Issue Type equals Test Plan</li> <li>Then: Send web request POST https://api.travis- d.com/repoXray-App%2/Flutorial-java- junit-travisci/requests</li> <li>Add component</li> </ul> Request parameters must be ut encoded, smart values about use: (types.utificnode)): Headers (potional) Accept application/json = . Add HTTP method POST v Webhook body Custom data v Wait for response Delay execution of subsequent rule actions until we've received a response for this webhook Eustom data "recuest": { "request": { [ "request": { "request: { "request": { "request": { "request: { "request: "{ "request": { "request: "{ "requ	Audit log	https://api.travis-ci.com/repo// /requests
<ul> <li>When: Manually triggered All logged in users can run rule.</li> <li>If: Issue Type equals Test Plan</li> <li>Then: Send web request POST https://pit.travis- d.com/report/srav.ApPi.Version 3</li> <li>Then: Send web request POST https://pit.travis- d.com/report/srav.ApPi.Version 3</li> <li>Add component</li> <li>Webhook body</li> <li>Custom data</li> <li>Webhook body</li> <li>Custom data</li> <li>("request": { ""oranit": "(aling", ""(fore)ect.key))", """(fore)ect.key))", """"""""""""""""""""""""""""""""""</li></ul>		Request parameters must be url encoded, smart values should use: {{value.urlEncode}}.
All logged in users can run rule. Content-Type application/json  Accept Actept Accept  Accept Accept  Accept Accept  Accept Accept  Accept	When: Manually triggered	Headers (optional)
If: Issue Type equals   Test Plan     Authorization   token   Travis-API-Version   3   Then: Send web request   POST   oc.com/repox/szPLotorial-java junit-travisc//requests   Add component     Vebhook body   Custom data   Olay execution of subsequent rule actions until we've received a response for this webhook   bustom data* <pre> {</pre>	All logged in users can run rule.	Content-Type application/json
If: Issue Type equals   Test Plan     Then: Send web request   POST   OST   OST   Ost response   Delay execution of subsequent rule actions until we've received a response for this webhoo   Custom data <pre> </pre>		Accept application/json
Then: Send web request   POST https://api.travis-   ci.com/repoXray-Apy%ZPtu/orial-java   junit-travisc/irequests   Add   HTTP method   POST   Vebhook body   Custom data   Delay execution of subsequent rule actions until we've received a response for this webhook   Custom data*   {             "request": {             "request": {             "request": {             "renort: "main",             "renort: "(issue key))",	If: Issue Type equals Test Plan	Authorization token
Add POST https://api.travis- ci.com/repo/Xray-App%2Flutorial-java- junit-travisc/requests Add HTTP method POST V Vebhook body Custom data Vebhook body Delay execution of subsequent rule actions until we've received a response for this webhook Custom data {		Travis-API-Version 3
Iner: send web request       HTTP method         POST hitps://pai.travis- i.com/repo/Xray-App%2Flutorial-java- junit-travisc/irequests       POST         Add component       Custom data         Wait for response       Delay execution of subsequent rule actions until we've received a response for this webhook bustom data         ("request": { "request": { "branch:" main", "config": { "remv"; { "TESTPLAN": "{(issue key})}", "ROJECTKEY": "{(project.key)}"		Add
ci.com/repo/Xray-App%2Ftutorial-java- junit-travisci/requests Add component Add component Wait for response Delay execution of subsequent rule actions until we've received a response for this webboo Custom data* {     "request": {     "config": {     "config": {     "config": {     "TESTPLAN": "{(issue.key)}",     ""ROJECTKEY": "((project.key))"     }     } }	POST https://api.travis-	HTTP method
Junit-travisci/requests Webbook body Uustom data Valit for response Delay execution of subsequent rule actions until we've received a response for this webbook Custom data*  {     "request": {     "oranch": "nain",     "config": {     "envr": {     "onr": {     "envr": {     "forject.Key)}",     "PROJECTKEY": "{[project.Key}])"     } }	ci.com/repo/Xray-App%2Ftutorial-java-	POST 🗸
Add component Wait for response Delay execution of subsequent rule actions until we've received a response for this webboo Custom data  {     "request": {     "oranch;":"main",     "config": {     "env": {         "TESTPLAN": "{(issue.key)}",         ""ROJECTKEY": "{project.key})"     } }	junit-travisci/requests	Webhook body
Add component Wait for response Delay execution of subsequent rule actions until we've received a response for this webhoc Custom data  {     "request": {     "oranch": "nain",     "config": {     "env": {     "env": {     "env": {     "TESTPLAN": "{(issue.key)}",     "PROJECTKEY": "{(project.key)}"     }     } }		Custom data 🗸
Delay execution of subsequent rule actions until we've received a response for this webhod Custom data  {     "request": {         "branch:"main",         "config": {         "erw"; {             "TESTPLAN": "{[issue key]}",         "TESTPLAN": "{[issue key]}",         }         }         }	Add component	Wait for response
Custom data" {     "request": {         "branch": "main",         "config": {         "env": {             "env": {             "env": {             "FROJECTKEY": "{[project.key]}]",             "         }         }		Delay execution of subsequent rule actions until we've received a response for this we
{     "request": {         "branch": "main",         "config": {             "env": {                  "TESTPLAN": "{(issue.key)}",                  "TESTPLAN": "{(iproject.key)}"		Custom data
		<pre>{     "request": {         "branch": "main",         "config"; {             "envr": {                  "envr": {                  "TESTPLAN": "{[issue.key]}]",                  "PROJECTKEY": "{[project.key]]"                  }</pre>
		Ca
Cancel		Validate your webhook configuration
Cancel  Validate your webhook configuration		

• the Webhook URL provided above follows this syntax:

- <TravisCI\_API\_URL>/repo/{slug/id}/requests (The %2F in the request URL is required so that the owner and repository name in the repository slug are interpreted as a single URL segment.)
- besides the "Content-Type" header that should be "application/json", define also an "Authorization" header having the value "token <token>", where you will place the authentication token obtained previously in the TravisCI page and the "Travis-API-Version" header is also mandatory and it will contain the version used.
- Custom data
  - ° We included the simplest possible just to trigger the pipeline from the master branch.

  - Added environment configuration variables to be used later in the TravisCI pipeline
     TESTPLAN that will be automatically filled with the test plan key from where the pipeline is triggered.
    - PROJECTKEY that will be automatically filled in with the project key.

After publishing the rule, you can go to the screen of an issue and trigger the TravisCI project/job.



In this case, since TravicCI was configured to report results back to Xray, a new Test Execution would be created in Jira/Xray.

Comicstore / COM-16 Execution results [1622475015055]											
🖋 Edit	Q Commen	t Assign	More 🖌	To Do	In Progress	Done	Admin	~			
<ul> <li>Details</li> <li>Type:</li> </ul>		Test Exect	ution			s	Status:		TO DO (View	v Workflow)	
Priority: Compon Labels: Test Plar Test Env	ent/s: n: ironments:	O Trivial None None COM-9 None				R	Resolution:		Unresolved		
<ul> <li>Descript</li> <li>Executio</li> </ul>	t <b>ion</b> n results impo	rted from exte	rnal source								
✓ Tests											
Overall E	ecution Status										+ Add ~
<b>4</b> PAS	S										
Total Test	ts: 4 Filter(s)										
	Apply Rank	🔶 Key	🔶 Summary		🖕 Test Type	#Req	ı #De	əf	Assignee	Show 100 ✔ entries	Columns 🗸
	3	CALC-1205	CanSubtra	ct	Generic	0	0		Helder Biscaia	PASS	• …
	4	CALC-1204	CanMultipl	у	Generic	0	0		Helder Biscaia	PASS	• …
	1	CALC-1203	CanDoStuf	f	Generic	0	0		Helder Biscaia	PASS	• …
	2	CALC-1202	CanAddNu	mbers	Generic	0	0		Helder Biscaia	PASS	• …
Showing	1 to 4 of 4 entr	ies								First Previous	1 Next Last

Associated with the Test Plan that we have passed along:

ComicStore / COM-16 Execution results [1622475015055]											
	P Edit	Q Comment	Assign	More 🛩	To Do	In Progress	Done A	dmin 🖌			
~	Details Type: Priority: Compone Labels: Test Plan Test Envi	ent/s: :: ironments:	Test Exect O Trivial None COM-9 None	ution			Statu Reso	s: lution:	TO DO (View Unresolved	Workflow)	
>	Descripti Executior Tests	ion n results impo	rted from exte	rnal source							
	Overall Ex 4 PASS Total Test: =	s: 4 F Filter(s)		_	_		_	_			+ Add ~
	<b>·</b> ·	Apply Rank	¢ Key	Summa	ry	🖕 Test Type	#Req	#Def	Assignee	Show 100 🗸 entries	Columns -
		3	CALC-1205	CanSubt	ract	Generic	0	0	Helder Biscaia	PASS	• …
		4	CALC-1204	CanMulti	ply	Generic	0	0	Helder Biscaia	PASS	•
		1	CALC-1203	CanDoSt	uff	Generic	0	0	Helder Biscaia	PASS	•
		2	CALC-1202	CanAdd	lumbers	Generic	0	0	Helder Biscaia	PASS	•
	Showing 1	1 to 4 of 4 entri	es							First Previous	1 Next Last

### Generic automation of processes

#### Copy fields from requirement/Story to Test whenever creating a Test or linking it to a story

Sometimes it may be useful to copy some fields from the requirement/Story to the Tests that cover it.

#### Automation configuration

On the Jira side we will use the Automation capabilities that it provides out of the box, so within the administration area go to the automation entry in the system settings and:

1. create a new rule and define the "When" (i.e. when it should be triggered) to be "Issue linked". Since Xray, by default, uses the issue link type "Tests" to establish the coverage relation between a Test and the requirement, we can take advantage of that to trigger the rule whenever such issue link is created.

	CALC	Summary Details	Automation Automation rules allow you to autom	nate repetitive tasks based on criteri	is that you set. Here you can manage existing rules and create new ones. Learn more about automation	Return to list
	CALC board     ~       Backog     Actors extrastic       Matching     Actors extrastic       Backog     Season       Status     Season </td <td>Audit tog Be-Index project Delete project Essic Essic Essic Merr Festure Pre-Condition Story Sub-Task Sub-Task Sub-Task Task</td> <td>Copy Fields from requirement to Tests  Test details  Comparison  Method  Met</td> <td>(PAALO)</td> <td>Susception of a state of the state of the ender instant, (issue)] will always in fair for the source issues, in always in state (issue)] will always in fair for the source issues, and (issue)] will be access the link type, use (illen'type)] (e.g. (illen'type)]=Daplacetry. Image: The issue issue issue is always in the source issue issue is always in the source issue. The issue issue issue issue is always in the source issue issue is always in the source issue. The source issue issue is always in the source issue issue is always in the source issue issue is always in the source issue. The issue issue issue issue issue is always in the source issue. The issue issue issue issue issue issue issue is always in the source issue. The issue iss</td> <td></td>	Audit tog Be-Index project Delete project Essic Essic Essic Merr Festure Pre-Condition Story Sub-Task Sub-Task Sub-Task Task	Copy Fields from requirement to Tests  Test details  Comparison  Method  Met	(PAALO)	Susception of a state of the state of the ender instant, (issue)] will always in fair for the source issues, in always in state (issue)] will always in fair for the source issues, and (issue)] will be access the link type, use (illen'type)] (e.g. (illen'type)]=Daplacetry. Image: The issue issue issue is always in the source issue issue is always in the source issue. The issue issue issue issue is always in the source issue issue is always in the source issue. The source issue issue is always in the source issue issue is always in the source issue issue is always in the source issue. The issue issue issue issue issue is always in the source issue. The issue issue issue issue issue issue issue is always in the source issue. The issue iss	
	Automated Steps Library     Add-ons  PROJECT SHORTCUTS Add a link to serul information for your whole team to see.      Add link	<ul> <li>4 more issue types</li> <li>Workflows</li> <li>Screens</li> <li>Fields</li> <li>Priorities</li> <li>Versions</li> </ul>	Add component			
		Components Users and roles Permissions Issue Security Notifications Project links				
а.		Project automation				

2. create a condition to ensure that the rule only runs for Test issues

Copy fields from ENABLED requirement to Tests	😒 Issue fields condition 🝵	
	Check whether an issue's field meets a certain criteria	
Rule details	Field*	
Audit log	Issue Type 🗸 🗸	
	Condition*	
When: Issue linked	equals 🗸	
Types: Tests		

3. use an "Edit issue" action to set the fields on the Test based on the fields of the linked requirement/Story (i.e., the *destination* issue of the linking event). In this example, we'll copy the values of Urgency and Probability custom fields.



This rule will run:

- whenever a Test is created from the requirement/Story issue screen
- · whenever a Test has been initially created and later on linked to the requirement

#### Reopen/transitionTests linked to a requirement whenever the requirement is transitioned or changed

Whenever you change the specification of a requirement/story, you most probably will need to review the Tests that you have already specified.

The following rule tries to perform a transition of **all** Tests linked to a requirement.

#### Automation configuration

On the Jira side we will use the Automation capabilities that it provides out of the box, so within the administration area go to the automation entry in the system settings and:

1. create a new rule and define the "When" (i.e. when it should be triggered) to be "Field value changed"

Project settings		
Summary Details Audit log Re-index project Delete project	Automation Automation rules allow you to automate repetitive tasks base reopen Tests if requirement	d on oriteria that you set. Here you can manage existing rules and create new ones. Learn more about automation
Issue types - Bug - Epic - Improvement - New Feature - Pre-Condition - Story - Sub-task - Sub Test Execution - Task	Rule details     Audit log     When: Value changes for     Description     If: Issue Type is one of     Story, Epic	Triaduda to monitor for changes
Versions Versions Versions Versions Versions Components	<ul> <li>For JQL Issue in requirementTests(((issue key))) </li> <li>Then: Transition the issue to To DO </li> <li>Add component</li> </ul>	
Users and roles Permissions Issue Security Notifications Project links Project automation		

a.b. Note: we could also define the trigger to be based on the transition of the requirement issue to a certain workflow status; in that case we would define it, for example, as shown below.

utomation rules allow you to automate repetitive tasks based (	on criteria that you set. Here you can manage existing rules and create new or	nes. Learn more about a
reopen Tests if requirement DRAFT changes	😡 Issue transitioned 🖉	
	Rule is run when an issue is transitioned through its work	low.
Rule details	From status	
Audit log	Leave blank to match all statuses	~
	To status	
When: Issue transitioned     TO     TO	To Do ×	~
	> More options	]

i.
 2. create a condition to ensure that the rule only runs for Story and Epic issues; adjust these to include all the "requirement" issue types (i.e., the ones that you can cover with Tests)

pen Tests if requirement DRAFT nges	🕫 Issue fields condition 🝵	
	Check whether an issue's field meets a certain criteria	а
Rule details	Field*	
Audit log	Issue Type 🗸 🗸	
	Condition *	
When: Issue transitioned	is one of 🗸 🗸	
то		
To Do	Value Field	
	Story × 4 Epic ×	~

a.

3. create a "branch rule / related issues" to obtain related Tests using JQL and the requirementTests() JQL function and run one, or more, action(s) on them

reopen Tests if requirement ENABLED changes	🕩 Branch rule / related issues 🍵
	Please select which related issue you would like to perform actions against.
Rule details	
Audit log	Type of related issues*
	JQL
When: Value changes for Description	JQL*
	issue in requirementTests({{issue.key}})
	Q Validate q
If: Issue Type is one of	Only include issues that have changed since the last time this rule executed
Story, Epic	A May cause performance issues
	Rule restricted to projects
For JQL issue in requirementTests({{issue.key}})	CALC (CALC)
	Only issues from the above projects will be considered. You can change this restriction in the 'Rule details' section.
	> More options
B Then: Transition the issue to	Cancel

4. under the "For JQL" block, create a action "Transition the issue to" in order to reopen the related Test issues

#### Automation

Automation rules allow you to automate repetitive tasks based on criteria that you set. Here you can manage existing rules and create new ones. Learn more about automation

reopen Tests if requirement ENABLED changes	Transition issue in
<ul><li>Rule details</li><li>Audit log</li></ul>	Transition the issue by: <ul> <li>Selecting the destination status</li> <li>Selecting a specific transition</li> </ul> Destination status*
When: Value changes for     Description	To Do × Ensure a transition from the issue's source status to your selected destination status exists; more info. + add regex to distinguish between multiple transitions to the same status
Story, Epic	<ul> <li>Choose fields to set </li> <li>More options</li> </ul>
For JQL issue in requirementTests({{issue.key}})	c
Then: Transition the issue to     Topo	
Ο	

### References

- Automation for Jira in the Atlassian Marketplace
- Jira Automation in Jira DC
  - Jira smart values issues
     Jira smart values lists

  - Jira smart values text fields

  - Jira smart values users
    Jira smart values conditional logic
    Jira smart values JSON functions
  - Branch automation rules to perform actions on related issues