

# PROJEQTOR INTEGRATION

**TECHNICAL DOCUMENTATION** 

VERSION: V 1.0

**REFERENCE: PLUGIN – INTEGRATION** 

# SOMMAIRE

INTRODUCTION	3
SETUP	3
GITHUB INTEGRATION	4
Configuration on Github side Configuration on Projeqtor side How to use Constraints	

# **INTRODUCTION**

The objective of this document is to explain how plugin "Integration" works.

This plugin enables to retrieve into projector some information from remote applications.

Information retrieved will depend on remote application and remote application configuration. On current version, this plugin covers integration with:

• GitHub

# **SETUP**

When acquiring a plugin, you get a .zip file. To install it, go to the Plugins Management screen.

rinstalled plug-ins	<ul> <li>available plugins (local)</li> </ul>
no plugin available	plug-ins (zip files) are searched in directory '/plugin'
	no plugin available
	add local plugin file Browse
	<ul> <li>vailable plugins (remote)</li> </ul>
	feature not available on this version

In the section of installed plugins, you have the list of plugins already in place with the version of the latter and on which version of ProjeQtOr you installed it.

In the available plugins section, click on the browse button to upload your .zip file.

Once the file is uploaded, it appears in a table.

Note: if your zip file is a package containing several plugins, all plugins will appear in the table.

ns (zip files	) are searched in directory '/plugin'		
	file	date	size
+ ® 前	integration_V1.0.zip	08/12/2022	28.7 KB
+ • m	webhook_V1.0.zip	08/12/2022	465.2 KB

Click on 🕂 to install your plugin

Click on <sup>(®)</sup> to view the plugin metadata.

Click on  $\boxed{10}$  to cancel the installation of this plugin.

When the zip is installed, the application restarts. Go to the Plugin menu to access it. After installation, the menus are available for the profile "administrator".

# **GITHUB INTEGRATION**

GitHub integration will allow to retrieve some information from message entered on push command:

- Comments from commit message
- Work from commit message
- Transition from commit message

### **CONFIGURATION ON GITHUB SIDE**

Fist, you need an account on Github. Use Github registration form. When it's done you need to have a project.

On your project, select "Settings"

<> Code		រឿ Pull requests	Actions	III Projects	🛱 Wiki	Security	🗠 Insights	lôg Settings
---------	--	------------------	---------	--------------	--------	----------	------------	--------------

#### Then select Webhooks

Options   Manage access   Security & analysis   Branches   Webhooks   Notifications   Integrations   Deploy keys   Actions   Environments   Secrets   Pages   Moderation settings	
Manage access   Security & analysis   Branches   Webhooks   Notifications   Integrations   Deploy keys   Actions   Environments   Secrets   Pages   Moderation settings	Options
Security & analysis Branches Webhooks Notifications Integrations Deploy keys Actions Environments Secrets Pages Moderation settings	Manage access
Branches Webhooks Notifications Integrations Deploy keys Actions Environments Secrets Pages Moderation settings ate a new webhook	Security & analysis
Webhooks         Notifications         Integrations         Deploy keys         Actions         Environments         Secrets         Pages         Moderation settings	Branches
Notifications         Integrations         Deploy keys         Actions         Environments         Secrets         Pages         Moderation settings	Webhooks
Integrations Deploy keys Actions Environments Secrets Pages Moderation settings ate a new webhook	Notifications
Deploy keys Actions Environments Secrets Pages Moderation settings	Integrations
Actions Environments Secrets Pages Moderation settings	Deploy keys
Environments Secrets Pages Moderation settings ate a new webhook	Actions
Secrets Pages Moderation settings ate a new webhook	Environments
Pages Moderation settings ate a new webhook	Secrets
Moderation settings ate a new webhook	Pages
ate a new webhook	Moderation settings
	ate a new webhook

Webhooks

Add webhook

Define payload to point to /plugin/integration/githubPayload.php on your server, in Json format

Payload URL *	
https://yourserver/plugin/in	tegration/githubPayload.php
Content type	
application/json	\$

Be sure to define a "Secret" and note it, you'll need it to configure Projector side. You may send all events or just push event (but at least pushes are required)

Which events would you like to trigger this webhook?

Just the push event.

Send me everything.

O Let me select individual events.

And be sure your webhook is active

Active We will deliver event details when this hook is triggered.

Save. That's all.

## **CONFIGURATION ON PROJECTOR SIDE**

# Access menu Plugin > Integration definitions. Create a new Integration Definition and select "github" a new integration source:

	<ul> <li>Description</li> </ul>	
	id	#
	name	
	integration source	github
		Source is the identity of your github <b>repository (without spaces)</b> If set will only integrate hooks from that source.
	ident of the source	
	secret	
	add information on note	
	closed	
		This configuration will receive push commands using smart comment syntax. <item_key> #<command/> <command options=""/></item_key>
		You may send several commands on same commit message. Commands are - comment   add a note - time   add work in format 0w 0d 0h 0m - any status   mode to this status
		Example Ticket-1 #comment this is a comment #time 2h #done will add a note and add 2h of work and move to status done
•	name	Name for this definition
•	integration source	Select "github"
•	ident of the source	Identity of your github repository (without spaces) - optional If the value is set, only hooks from this source will be integrated
•	secret	Secret defined on your github webhook
•	add information on note	<ul> <li>If checked, information about changes will be added as a note to the item (on the note stored as comment if comment is set on message)</li> <li>This information can then be clicked to reach github commit information on github website</li> </ul>
•	closed	enable / disable this definition

### How to use

When you push changes to your github repository, you can add a commit message. Use Smart commit command in the commit message to interact with Projector <u>https://confluence.atlassian.com/fisheye/using-smart-commits-960155400.html</u>

### Syntax for Smart Commit message is:

<ig< th=""><th>nored text&gt; <issue_key></issue_key></th><th><igr< th=""><th>nored text&gt; #<co< th=""><th>MMAND&gt; <command_args></command_args></th></co<></th></igr<></th></ig<>	nored text> <issue_key></issue_key>	<igr< th=""><th>nored text&gt; #<co< th=""><th>MMAND&gt; <command_args></command_args></th></co<></th></igr<>	nored text> # <co< th=""><th>MMAND&gt; <command_args></command_args></th></co<>	MMAND> <command_args></command_args>
•	<issue key=""></issue>	Ref	erence to Projeqt	or element.
	_	Syr	itax is <class>-<ii< td=""><td></td></ii<></class>	
		For	instance Ticket-	refences the Ticket of id 1
		Cla	ss is not case sens	itive (you can either enter ticket or Ticket or TICKET)
		Sor	ne shortcuts are a	llowed for the class:
		0	T, TKT, TIC	for Ticket
		0	A, ACT	for Activity
•	# <command/>	The	command to apply	· · · · · · · · · · · · · · · · · · ·
		Ava	ailable commands	are
		0	#comment	adds a note to the item
		0	#time	adds some real work on the item
		0	# <status></status>	moves the item to corresponding status
		Υοι	ı can enter severa	l commands on same commit message
•	<command_args></command_args>	The	arguments for the	command
		0	for #comment	the text of the new note
		0	for #time	time in format Ow Od Oh Om
				(replace 0 with any numeric value, decimal allowed)
				$_{ m w}$ is weeks, $_{ m d}$ is days, $_{ m h}$ is hours, $_{ m m}$ is minutes
				For instance, to enter 1 hour and a half, you can
				enter 1.5h or 1h 30m or 90m
		0	for # <status></status>	no argument is expected

Any text between the issue key and the Smart Commit command is ignored.

#### Example:

TICKET-1 #comment mise à jour annulées OK #time 2h #done

This commit message will

- Add new note with text "mise à jour annulées OK"
- Add real work for 2 hours (for user = sender, day = today) on the item
- Move item to status "done" (if workflow allows this move)

If option « add information on note" is checked on the Integration Definition, note will look like:

4	36	۵ 🛍	#48	mise à jour annulées OK	07/12 P
				modified branches/V9.3/projeqtor/model/Action.php branches/V9.3/projeqtor/model/Activity.php branches/V9.3/projeqtor/model/Asset.php	

and area displaying modified source code can be clicked to directly move to the commit changes on github.

### **CONSTRAINTS**

Sender of the push command must be known as a user in Projector.

User is searched from email address of the pusher (searching on email of projector user), and if not found from the name of the pusher (searching on name of projector user).

All Projector access rights are controlled over rights of this user on the corresponding item. This includes:

- rights to update the item
- rights to enter work on the item
- rights the move the item to the corresponding status as defined on the workflow

If the user does not have the right to execute the command, the command is rejected.