



EL OYENTE JENKINS PLUGIN

User guide

Authors:
Juan Luis Pardo González
Isabel Fernández Díaz

Contents

1	GENERAL INFORMATION	3
1.1	Plug-in Overview	3
1.2	Prerequisites	3
2	CONFIGURATION AND USE	4
2.1	Installation guide	4
2.2	Using the plug-in	6
2.2.1	Main configuration	6
2.2.2	Configuring a job	7
2.2.3	Setting up filters	9
2.2.4	Using environment variables	10
3	IMPROVEMENTS	11

1 General Information

1.1 Plug-in Overview

ElOyente (the listener in Spanish) is a plug-in for [Jenkins](#) that allows Jenkins jobs to:

- Subscribe for XMPP Pub/Sub events, to trigger the job.
- Add filter expressions in XPath notation, to limit the triggering to particular events.
- | ▪ Extract data from the events with XPath expressions, and make it a [_](#)available as environment variables.

1.2 Prerequisites

It is required to set up or to have access to an XMPP server in charge of handling the events and messages. There exist several available servers that could be used. Follow this [website](#) to find some examples.

For the development of the plug-in, it has been used **Openfire**. This sever is quite easy to set up and use. For more information, visit its [website](#), where you can find the [installation guide](#).

2 Configuration and use

2.1 Installation guide

To install ELOyente, just follow these easy steps:

1. Start Jenkins, and once in the dashboard, click on **Manage Jenkins**

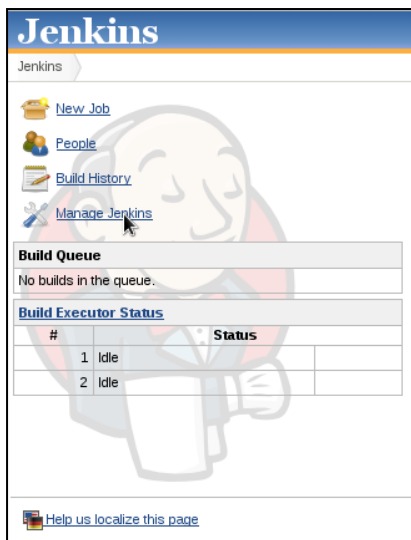


Figure 1: Jenkins dashboard menu

2. Then select **Manage Plugins**

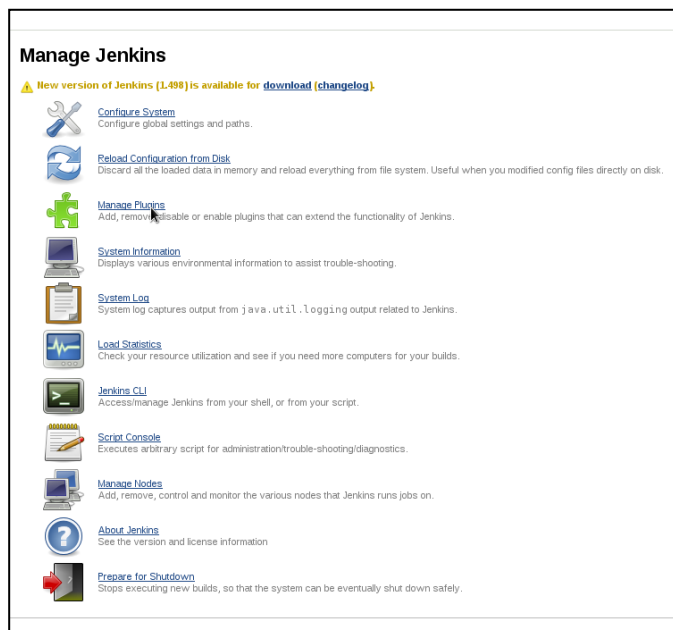


Figure 2: Jenkins main menu

3. Select the **Available** tab.



Figure 3: Available plugins in Jenkins

4. Scroll down to the section **Build Triggers**, there you'll find ELOyente.

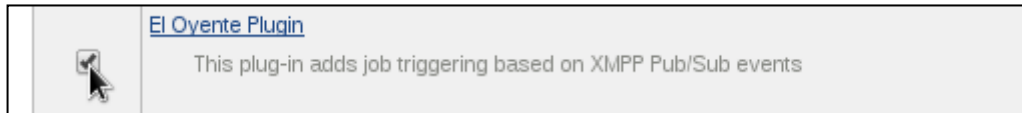


Figure 4: El Oyente plugin available

5. After clicking **Download and install after restart** the plug-in will be installed and ready to be used.

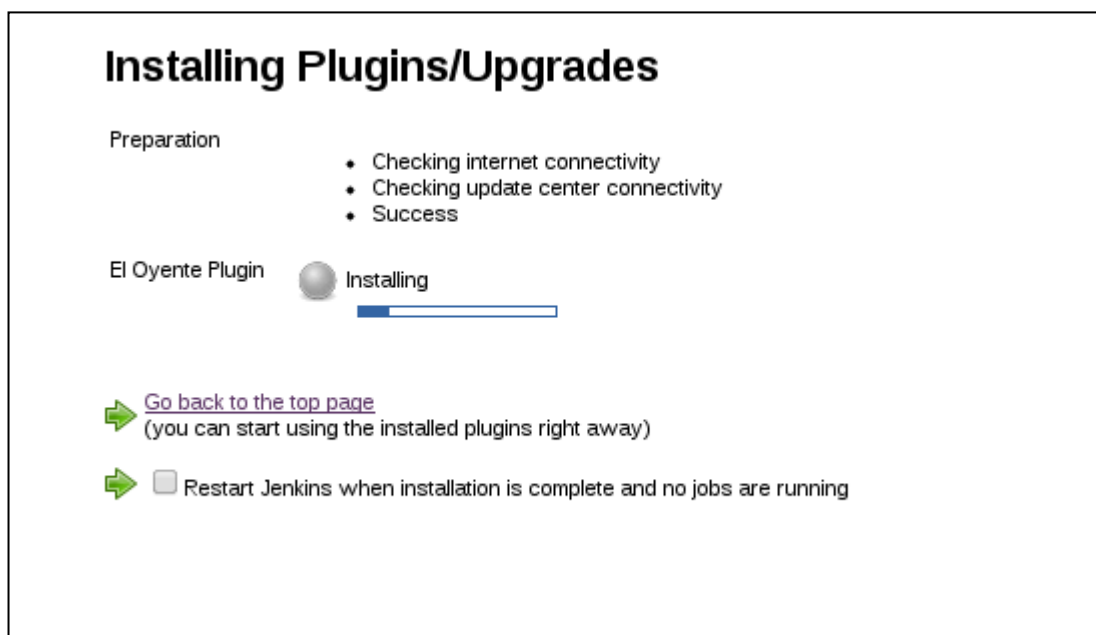


Figure 5: Plug-in installing

2.2 Using the plug-in

2.2.1 Main configuration

In order to set the plug-in to work it is necessary to fill the parameters of the server we are going to use, this is done in the main configuration.

To access the main configuration of Jenkins, click in **Manage Jenkins** and after that in **Configure System**.

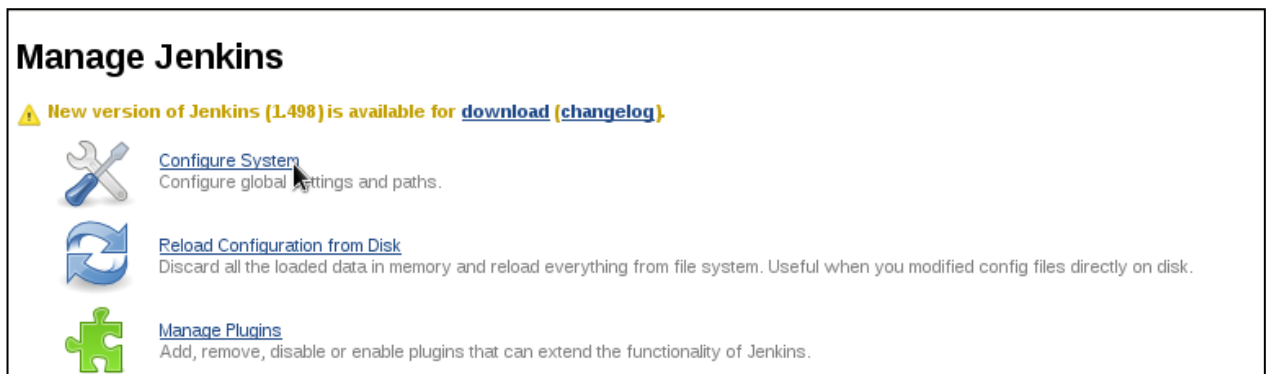


Figure 6: Configure System

This will open a menu like the one below, search for the section XMPP job triggering. There you can specify the server address of your XMPP server, a user and a password.

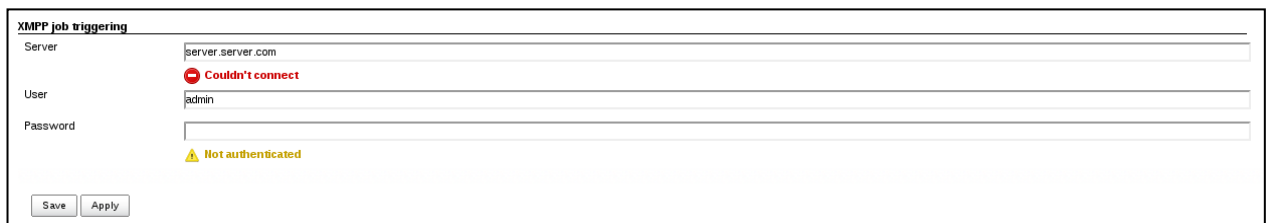


Figure 7: Main EOyente configuration

As you can see the interface will give you feedback about the connection and authentication. In the example, as the server is not valid, it cannot connect, and as the user and password are not correct it shows that you are not authenticated.

Once these parameters have been set we are ready to use the plug-in within our projects.

! *If these values are empty or not correct, still you'll be able to use the plug-in in the projects, but of course the connection won't be created and they won't react to any possible events. You can set the main configuration parameters later and once you click Save, you'll start receiving the XMPP events.*

2.2.2 Configuring a job

These are the steps needed to create a job and set up ELOyente plug-in for listening to XMPP events.

First create the job and give it a name, it doesn't matter the type of job you're creating, the plug-in will always be available.

Jenkins All

New Job

People

Build History

Manage Jenkins

Build Queue

No builds in the queue.

Build Executor Status

#	Status
1	Idle
2	Idle

Job name: TestJob

☒ **Build a free-style software project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system.

☐ **Build a maven2/3 project**
Build a maven2 project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

☐ **Build multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments.

☐ **Monitor an external job**
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. [details](#).

☐ **Copy existing Job**
Copy from:

OK

Figure 8: Creating a job

Once created, you'll see the configuration screen, scroll down to the section Triggers, and click on the checkbox labelled as ***XMPP triggered plug-in*** to activate it. Then click on **Add subscriptions** and you are ready to subscribe to the nodes from which you'll receive your events.

Just write on the textbox the name of one of the nodes available in your XMPP server, you can add more nodes if needed, just clicking on **Add subscription** as many times as you need. In case you need to delete one of the nodes just click on the **Delete subscription** button on the right side under the subscription you want to delete.

Once you click the **Save** button the plug-in will be listening for the XMPP events over the nodes you selected. The job will be triggered as soon as you receive an event from those nodes.

NOTE: It is possible to subscribe to nodes that still don't exist, once they are created the plugin will start listening when clicking on the **Save** button in the configuration of that particular job, or in the **Save** button of the main configuration.

You can also subscribe several times to the same node in one particular job, in that case a build will be triggered for each of the subscriptions.

☒ XMPP triggered plugin

Subscriptions

Node

TestNode

Filter

Environment variables

Add variable

Delete subscription

Node

TestNode2

Filter

Environment variables

Add variable

Delete subscription

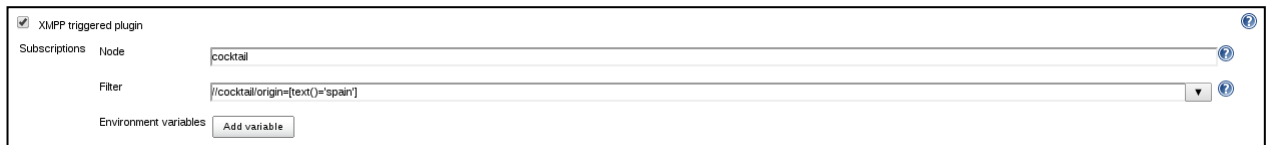
Add subscription

Figure 9: Listening to nodes

2.2.3 Setting up filters

Under the node subscriptions we see a textbox where you can add a filter for that particular subscription. If empty, the job will be triggered for any event occurring on the selected node. The filter expression allows the job to be triggered only if the event matches a certain criteria. The filter is an expression in XPath notation, which is applied on the XMPP event. If the expression evaluates to true, the job is triggered. See XEP-0060 for more information about XMPP Pub/Sub.

In the image below you can see a subscription to a node "cocktails", the filter is set so only the cocktails from Spain will trigger the job. When an XMPP message is received it is filtered, if the plug-in finds the tag origin with a value of Spain, the triggered will be fired, in other case, the job will not be triggered.



The screenshot shows a configuration window for the "XMPP triggered plugin". It has a "Subscriptions" section with a "Node" field containing the text "cocktail". Below this is a "Filter" field containing the XPath expression `/cocktail/origin=[text()='spain']`. At the bottom, there is an "Environment variables" section with an "Add variable" button. The window includes standard UI elements like a checkbox for the plugin, a help icon, and a dropdown arrow for the filter field.

Figure 10: Filtering events

2.2.4 Using environment variables

The plug-in also allows you to save information from the event into environment variables that can be referred in the rest of the job, you can add as many as you want.

To set an environment variable you must click in the button **Add variable** and two fields will appear.

The first field tagged as **Name** lets you choose a name for the variable, this variable can be referred in the rest of the job with the \$ operator.

The second field **Value selection** allows specifying an XPath expression, which will be applied on the XMPP event. The result of this will be used as the value for the environment variable.

In the example it is extracted the name of the cocktail, a list of steps for preparation, an URL that contains the image and a list of ingredients. All this information is extracted from the payload of the XMPP message received and could be used for example to fill an HTML file that shows the cocktail characteristics.

The screenshot displays the configuration window for an 'XMPP triggered plugin'. It features a sidebar with 'Subscriptions' and 'Node' (set to 'cocktail'). The 'Filter' field contains the XPath expression '//cocktail/origin[text()='spain']'. The main area is titled 'Environment variables' and contains four variable definitions:

- Variable 1:** Name: NAME, Value selection: //cocktail/name/text().
- Variable 2:** Name: PREPARATION, Value selection: //cocktail/preparation.
- Variable 3:** Name: IMAGEURL, Value selection: //cocktail/imageurl/text().
- Variable 4:** Name: INGREDIENTS, Value selection: //cocktail/ingredients.

Each variable definition includes a 'Delete variable' button. At the bottom, there are buttons for 'Add variable', 'Delete subscription', and 'Add subscription'.

Figure 11: Environment variables

3 Improvements

- Adding dropdown menu for nodes subscriptions with the available nodes in the server
- Giving feedback to the user in the job configuration in case the configuration of the server is not filled or it is not possible to establish the connection.

END OF DOCUMENT