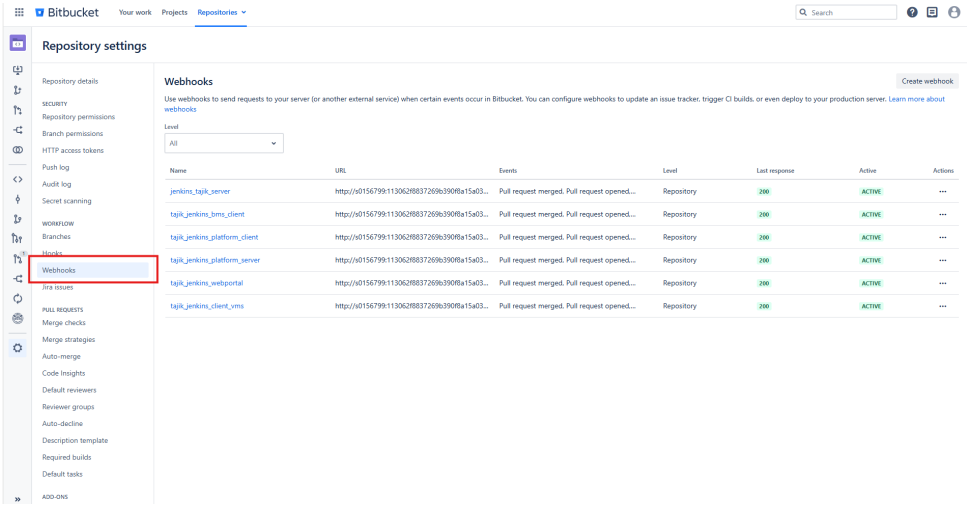
Setup Jenkins webhook triggers for a project (WIP).

First we will be needing a user that has permissions to launch the jenkins jobs. The user for continuous integration is SG_MAD_EJOURNEY_CI.
It is already setup for our jenkins instance. If a project is hosted in a different jenkins instance, then we would be needing to talk with somebody from IT (probably Alberto Robles), and ask him to generate a token for such user, and give it to us.

Setup Webhooks in BitBucket.

For this you will be needing administration permissions in Bitbucket for the project you need to setup. Anyone with these permissions will be able to modify these webhooks.

First thing, go to the webhooks tab for your repository, inside the repository settings:



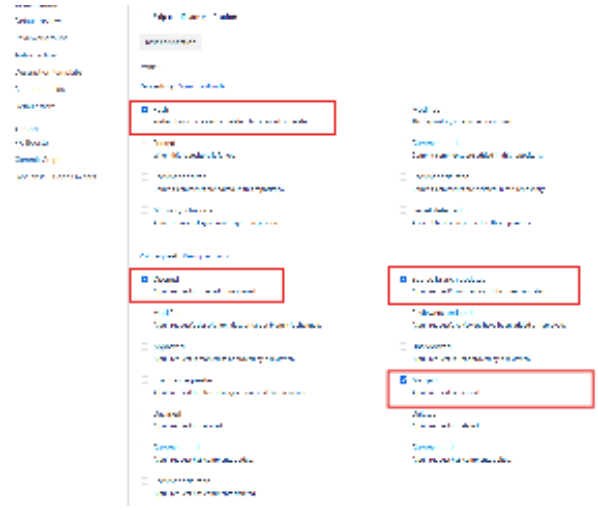
Then push the button `Create Webhook` :

Here you will fill in

- the name for your hook,
- the url that needs to be invoked,
- the events that will trigger the webhook.

Of these three, the most important, and the most sensitive one is the url.

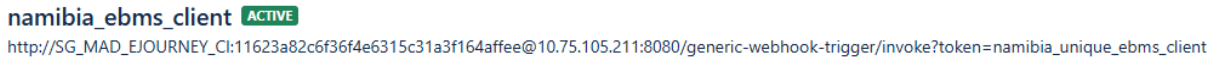
First let's choose the events that will trigger the webhook (bottom of the page):



with the checks shown the following events will trigger a build:

1. When there is a new commit in a branch.
2. When a new pull request is open, in the "from" branch.
3. When a pull request is updated with new code (amend, new commit, etc).
4. When a pull request is merged.

The url that needs to be filled looks like this:



http://SG_MAD_EJOURNEY_CI:11623a82c6f36f4e6315c31a3f164affee@10.75.105.211:8080/generic-webhook-trigger/invoke?token=namibia_unique_ebms_client

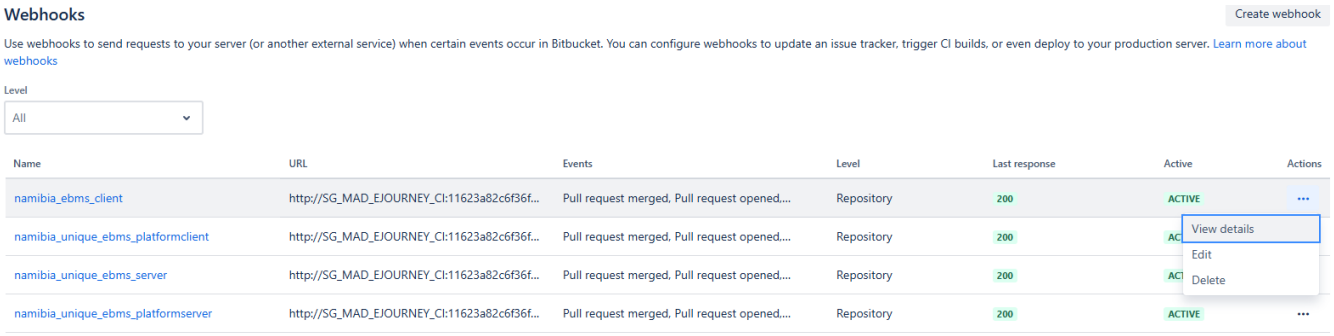
Url analysis:

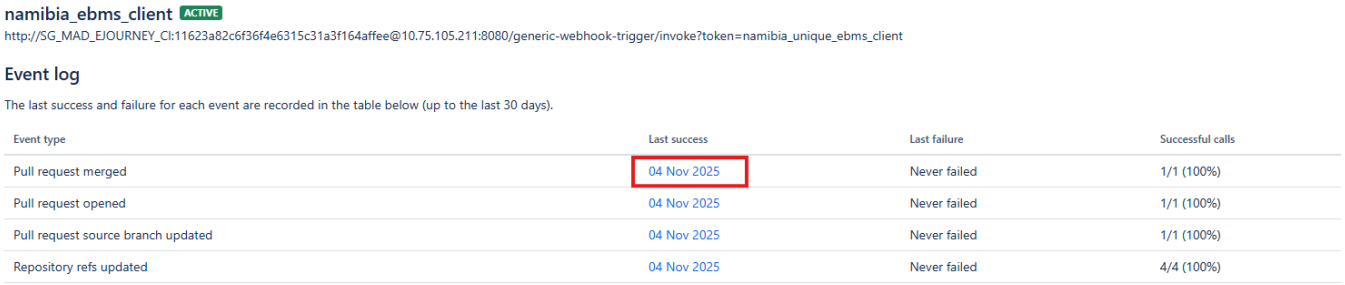
SG_MAD_EJOURNEY_CI	This is the jenkins user with permissions to trigger the jobs. This one we used here is the designated one for continuous integration.
11623a82c6f36f4e6315c31a3f164affee	This is the jenkins token for such user. Must be generated in jenkins. More information in the jenkins configuration part. This must be done for each user that triggers the builds and for each server. It is not inherited.
10.75.105.211:8080	This is the jenkins instance where your jobs are located.
generic-webhook-trigger/invoke?token=	This is a fixed part of the url that the plugin we are using imposes. That plugin is generic-webhook-trigger : Link
namibia_unique_ebms_client	This is an unique identifier of the job that will trigger. More information in jenkins part.

For each job you need to trigger, you will need one webhook, as we have seen there is a unique identifier that only identifies one job. So, if you have several jobs that need to be triggered when updating one repository, as is our case, then you will be needing several webhooks in one repository.

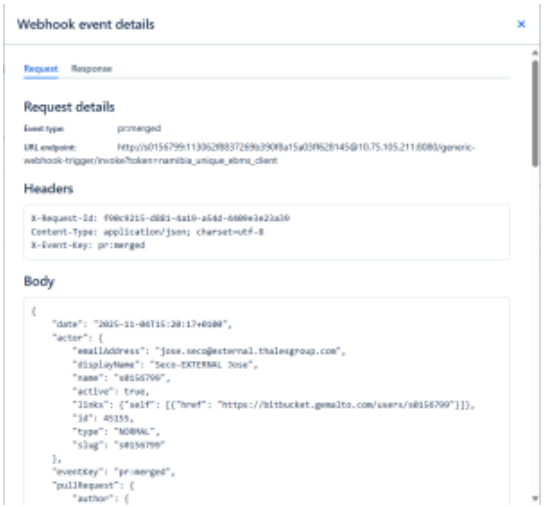
See the events requests.

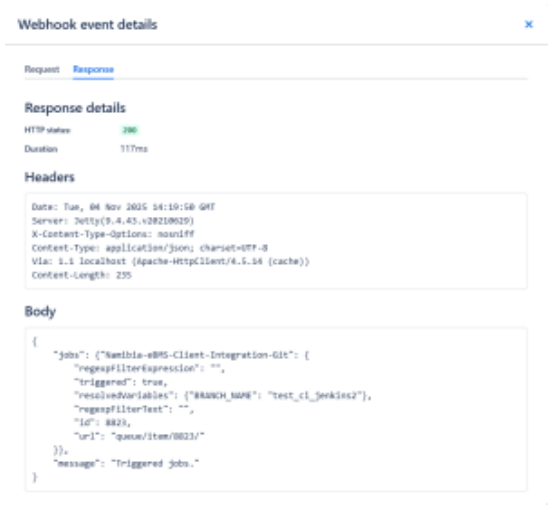
Once such events are fired, the last one of them will be recorded, so that you can review it:





And there, you can review the request, and the response from jenkins. In a request that works, we would see something like the following images, but more of the capture later.

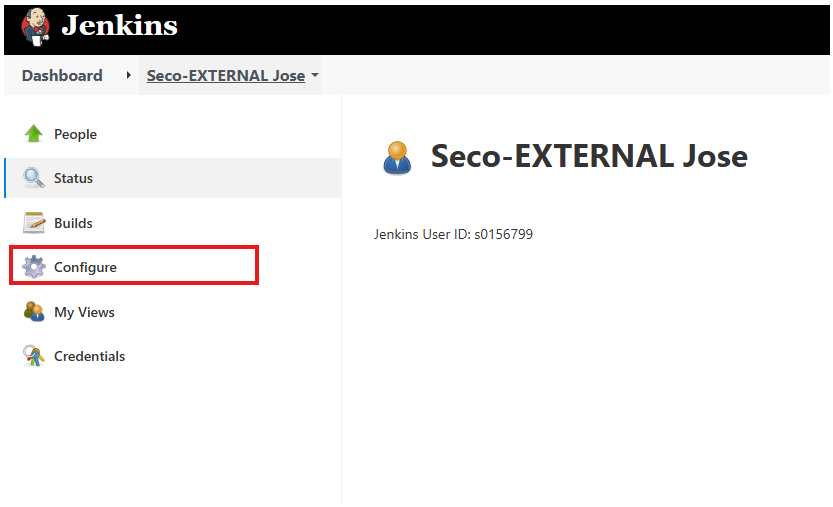




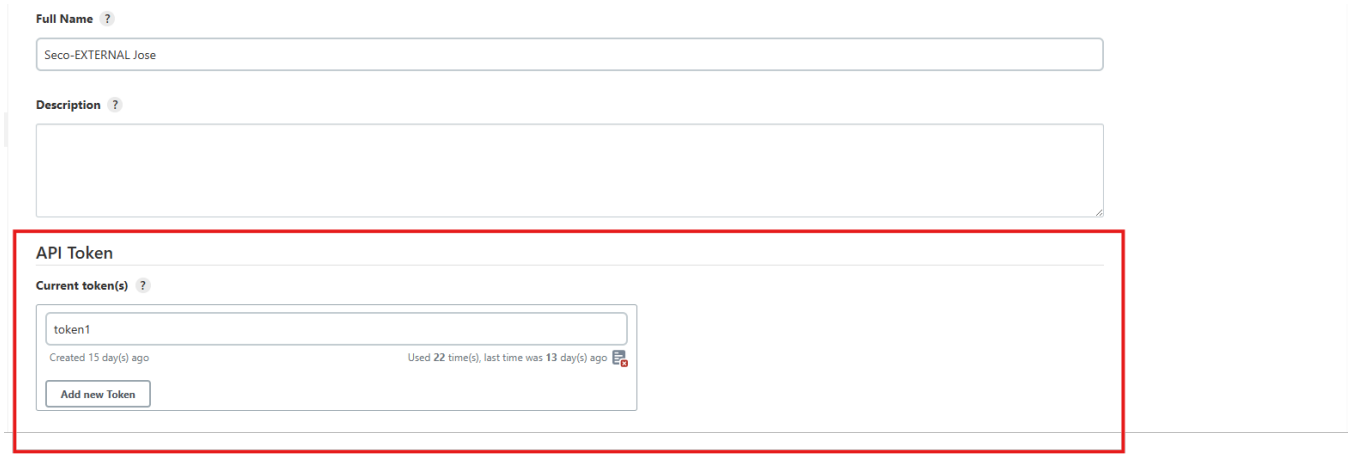
Jenkins Setup

Generate user token

Tokens are generated in your user configuration. So first you must login with the desired user, then push on such user name (up-right). The screen you will se will be like this:



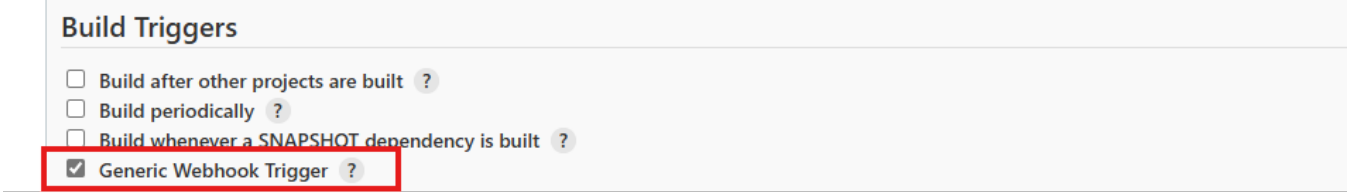
If you push "Configure", there will be a section for generating a token:



Configure plugin for each of the jobs.

This needs to be done in each of the jobs that will be triggered.
The general way of working is that it will capture data from the body of the request that is sent from Bitbucket to Jenkins. The body of such request is in json format. With the data you capture, you can override the values of the variables of the job, or define new variables that you want to use for something.

In the configure tab of the job, you will get a new Build Trigger, that has been added by this plugin:



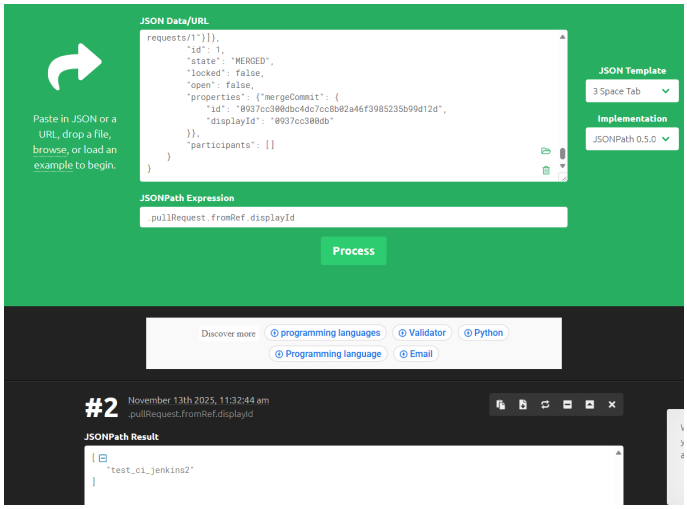
Don't have any other trigger active. Specially don't have this one, that makes the whole thing not work:

That way, it will be setup to be triggered just manually, or with this webhook trigger.

When you push the "Generic Webhook Trigger" check, it will expand for the configuration of such webhook trigger.

First thing you will usually want to capture, is the branch, the variable name will change from repository to repository. As we have different events that we are capturing in Bitbucket, the requests will have different bodies, so we will need to be search for such branch in different places. As we don't know which request we are receiving, we need to define the same variable as many times as needed to match all of the requests.

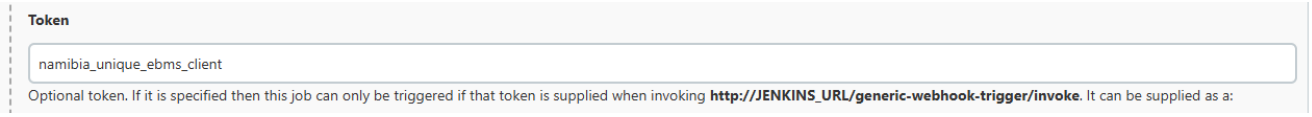
First one would go to this web, to test their capture of the variable, using the request body copied from Bitbucket: Link



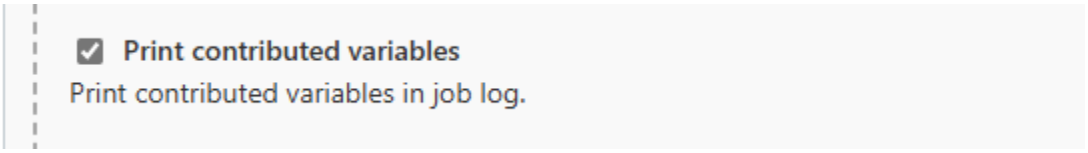
And then would capture that variable, testing with as many bodies of the requests, as evens are. In our case, we only need to capture 3 ways, because for the Pull Requests events, we can find the branch in the same place every time.

Additionally you can redefine other variables that you want, using an incorrect JSONPath, or define new variables that you want for some reason. This way we can make sure Fortify is always triggered, for example:

Then we have the unique token that is used as unique identifier in the request:



And then, finally we have a check that allows us to debug the captured variables in the log, if enabled:



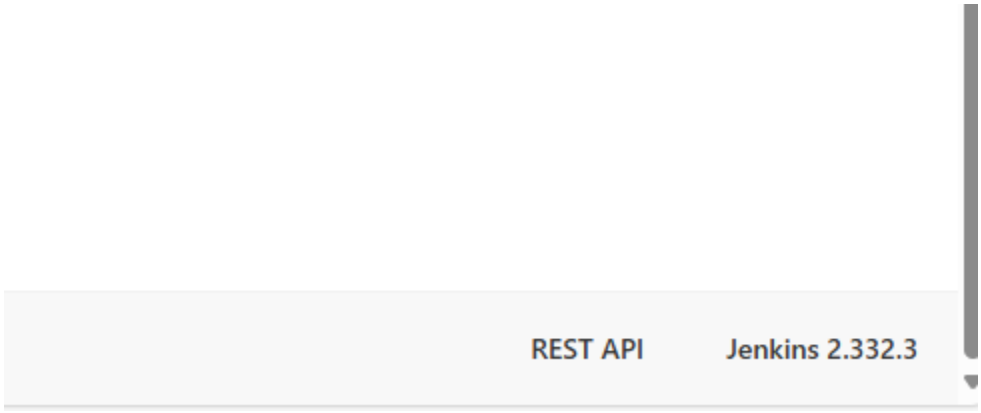

```
Contributing variables:

    BRANCH_NAME = test_ci_jenkins2
    RUN_FORTIFY = true
```

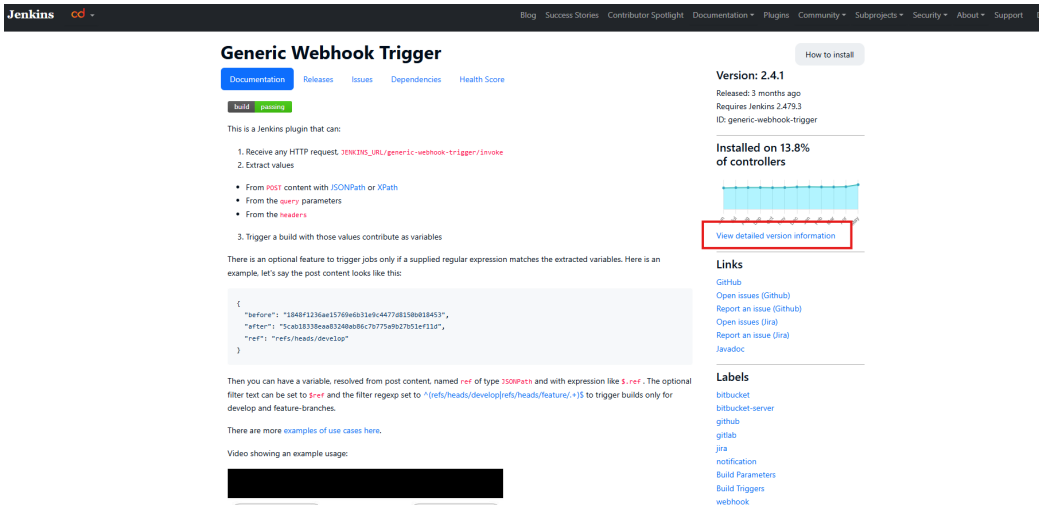
Annex: Installing generic-webhook-trigger plugin

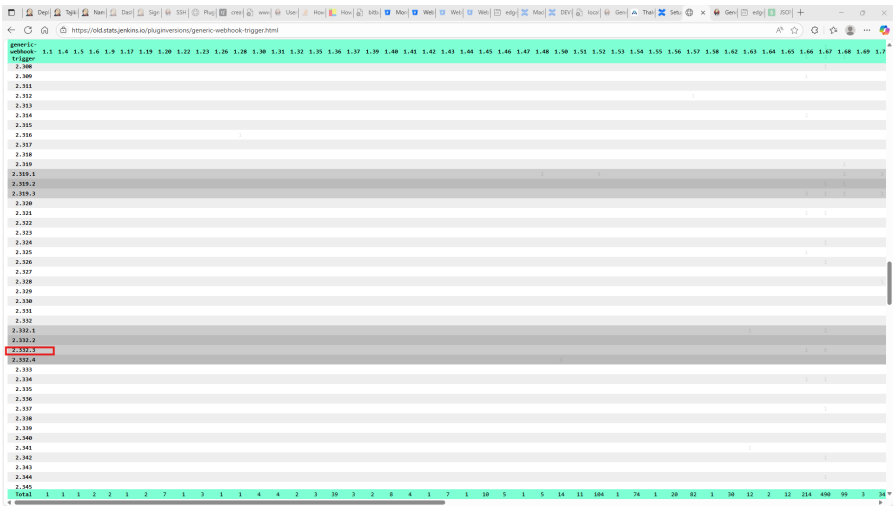
In case you need to install the plugin in a new instance, the most important thing is to install a plugin version that is compatible with your current jenkins version:

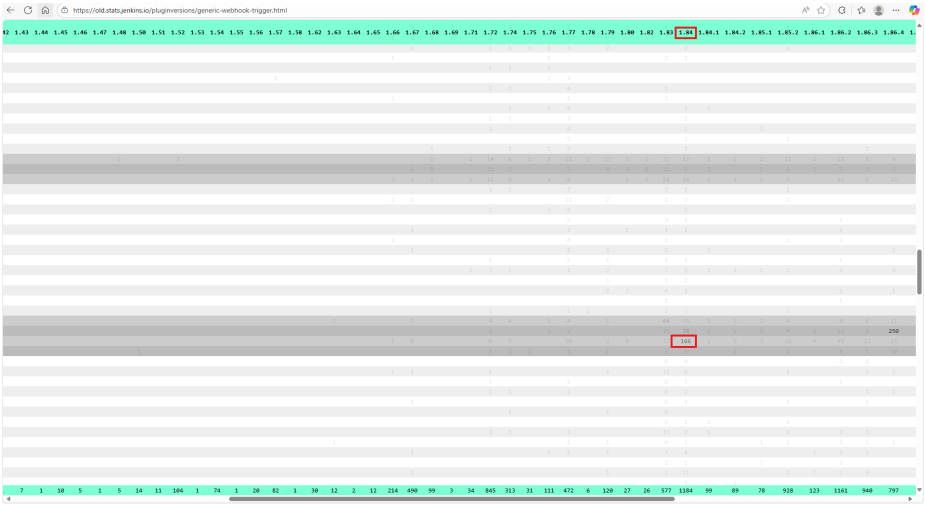
You can check the jenkins version in the bottom right corner:



And check the compatibility chart in the plugin page:







So version of the plugin we must install is 1.84.