**JMeter Automation Tool**

Step-by-step usage guide

By Adam Nogueira

Project Intro

This file contains instructions for setting up a machine and directory for running the JMeter Automation Tool. This tool was created with the purpose of making it easy to run JMeter tests over an API with minimal setup and specific knowledge, is composed mainly by a Jenkinsfile and a few default JMeter test-files.

For getting everything setup, you will simply need to download Java, Jenkins and Jmeter, have a repository with the project files, and choose a few parameters in a user-friendly manner.

Important: this tool works exclusively on the Windows OS.

Downloads

- Java

<https://www.java.com/en/download/>

Simply follow the link above, and installing make sure to select the “add to Path” option, as this will allow us to run java from the cmd. Remember where your installation is, as you will need it for the Jenkins setup.

After that reset your PC for the path to setup properly.

- JMeter

<https://jmeter.apache.org/download_jmeter.cgi>

This time follow the link above and click on the apache-jmeter-X.X.zip installation. After that extract the zip to wherever you want your JMeter setup to be, for ease of use I will be extracting it to the root of my machine, and with the 5.5 version I will be left with a path “C:\apache-jmeter-5.5”.

- Jenkins

<https://www.jenkins.io/download/thank-you-downloading-windows-installer/>

By simply following this link, the download should start. When installing just make sure to choose the option depicted in the following image, unless you want to setup with Account and Password, and choose the correct java installation path. After that a jenkins.war file should appear wherever you executed your installer, which I moved to my root again for ease of access.

A screenshot of a computer

Description automatically generated with medium confidence

Jenkins Setup

- Start Jenkins

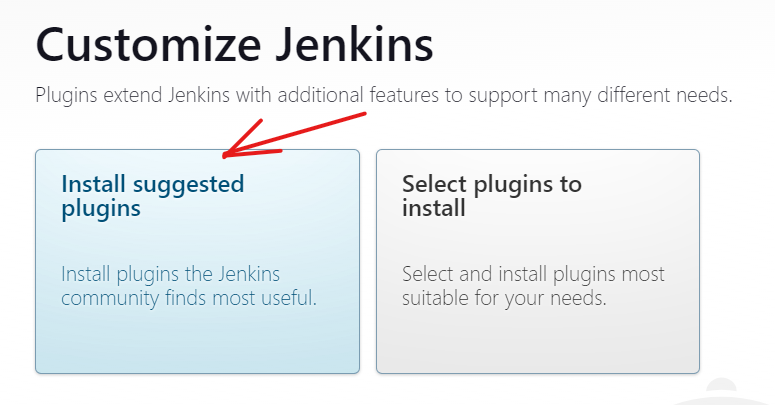
Using the cd command, go to the directory where the jenkins.war file is located. There, run the following command: “java -jar jenkins.war”.

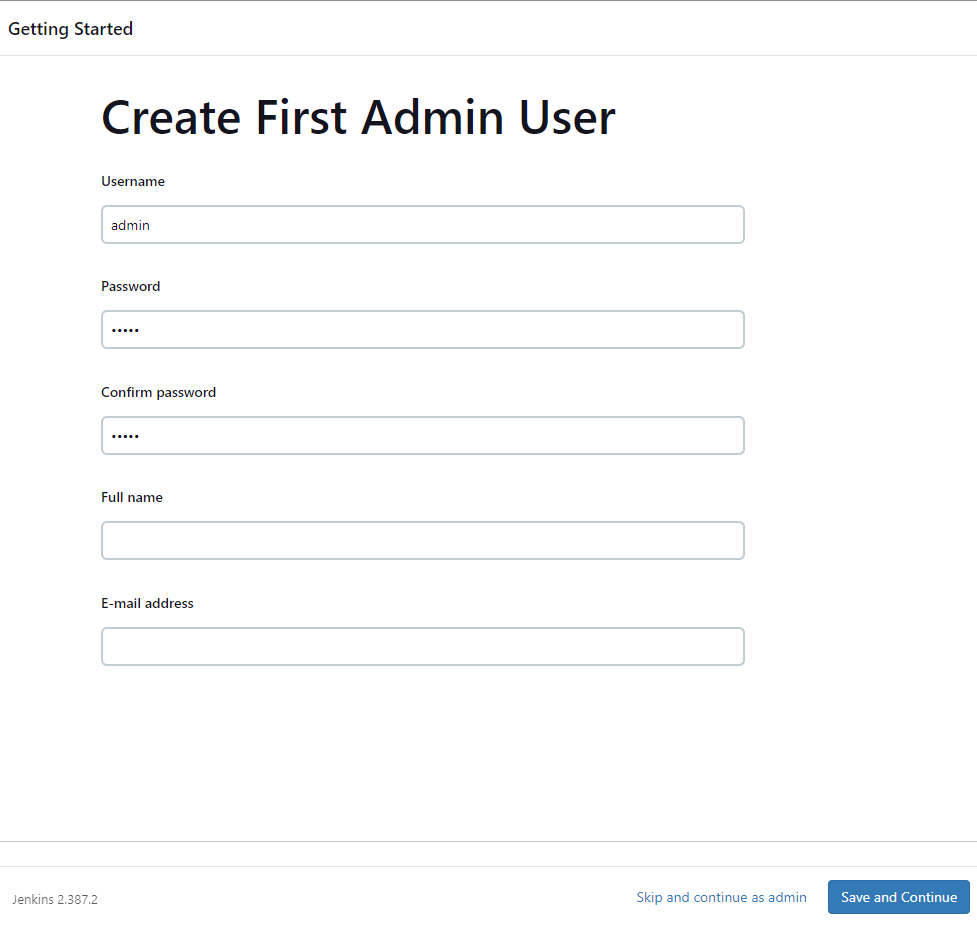
Something like this should be left in the cmd:

A screen shot of a computer screen

Description automatically generated with low confidence

Now open the localhost you chose in the Jenkins installation (should be localhost:8080 if you have not changed it), where you should input the password saved in the file specified in the cmd output. After that select the following option to install the base Jenkins plugins.



After that you should be on a window similar to the following one, where you should create an admin user, which will be used to log in every time you start Jenkins from now on:

After that you just confirm the localhost url, and you have Jenkins ready to operate.

- Configure Jenkins

First, however, you should make sure the performance plugin is downloaded and working, for that go to Manage Jenkins -> Manage Plugins -> Available Plugins, where you will look for the following plugin:

A screenshot of a computer

Description automatically generated with low confidence

After selecting it, click at the bottom of the page “Install without restart”.

Now all we are missing is setting up the email notification sender, for which you can go to Manage Jenkins -> Configure System -> Extended E-mail Notification. Here you will need to provide the access to the email that Jenkins will use to send the results of the tests, in my case it looks like this:

A screenshot of a computer

Description automatically generated with low confidence

If you want to use a free gmail account, I recommend following this guide for getting the email password, and filling the other values as depicted:

<https://support.google.com/mail/answer/185833?hl=en>

Also don’t forget to click “Apply” and “Save”, which will send you back to the Jenkins front page.

- First and Second Run

The last part of the setup is to create the pipeline and run it for a first time, in order to get the Jenkinsfile and know which parameters it requires, and a second time, for you to input said parameters for the first time.

A screenshot of a computer

Description automatically generated with low confidenceTo create the pipeline, you can simply click New Item, from there input a name for your pipeline and select the pipeline option, as follows:

Now you are on the configurations page for your pipeline. From there you can add a description if you want, but you must go to the Pipeline section, and choose the Pipeline Script from SCM option, inputting your repository with the Jenkinsfile and whatever credentials needed to access it.

Also make sure to select the correct branch, but in my case, as my repo is public and the branch is called master, it looks like this:

A screenshot of a computer

Description automatically generated with medium confidence

Now click “Apply” and “Save” and you are good for a first execution, by simply clicking “Build Now”. Note that the run will fail, you can delete it later if it’s a bother.

However, if you refresh the page after the completion, you will see that the Build Now button has become a Build with Parameters button, which you can click on to open the parameters page of the build.

Here you can set the parameters as whatever you want (as long as it’s valid), they are mostly self-explanatory, but for this first build with parameters you **must** select a *triggerMode*, or else the build will fail:

A screenshot of a computer

Description automatically generated with low confidence

Another important parameter is the *jmeterBinPath*, which in my case is “C:\apache-jmeter-5.5\bin”, but just needs to be the directory where your JMeter executable is.

And now you are good to go! Use this pipeline however you want.

Further Modifications

- Custom test file

If you want to use a custom jmeter test-file, feel free to put it as a parameter, but know that as Jenkins executes it with our pre-set command, I recommend using the following variables given in the command line in a HTTP Request Defaults element: *serverName*, *pathName*, *protocolType*, and *bodyData* for non-Get requests.

You can also remove them from the command by modifying the line 49 of the jenkinsfile.

- Custom email

We make use of a java parser to convert csv into a script-less html with in-line styles, so that it can be sent over email. If you use different types of listeners, you may generate different types of data, and want to change the email template, for that just change the CSVtoHTMLConverter.java file, and then in cmd go to the directory where it resides and run “javac CSVtoHTMLConverter.java”.

This should create a new CSVtoHTMLConverter.class file, which will be used by Jenkins in the future runs for email generation.

- Custom trigger modes

If you want to change the frequency at which the scm is polled in the “Once every Commit” triggerMode, you can do that in the line 117 of the jenkinsfile, by changing the cron expression for whatever you’d like.

In lines 125 and 126 you can also change the time of day at which the “Daily” triggerMode runs, by inputting a new hour and minute, which are preset to midnight.

Conclusion

If everything worked out as it should, you must have sent/received an email like this:

A screenshot of a computer

Description automatically generated with low confidence

It is worth noting that this guide is up-to-date and written in 16-06-2023, and this project has a lot of room for improvement. Next steps in development should include better emails, multiple test-file execution, and creating a machine in a cloud service with the Jenkins and JMeter installations, so that this setup becomes even simpler and easy to deploy.