# **Run Postman Collection on Jenkins**

## **Run Postman Collection on Jenkins**

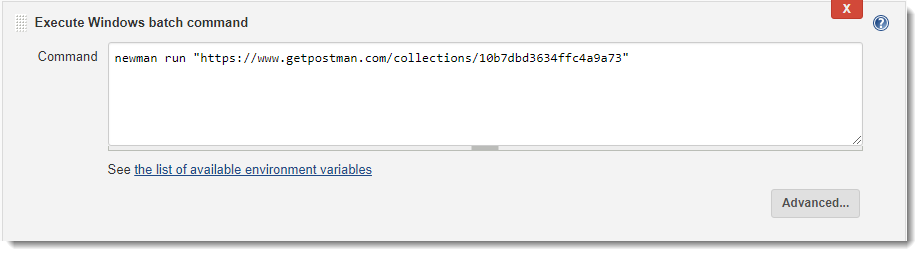
***Run Postman Collection on Jenkins*** is same as running it over the shell/command prompt.

***Prerequisites***

* *A collection running in Postman correctly with at least 1 test (Refer*[***How to Create & Run Collections in Postman***](https://toolsqa.com/postman/collections-in-postman/)*)*
* *Newman installed and knowledge of Newman (Refer*[***Newman Tutorial***](https://toolsqa.com/postman/what-is-newman-in-postman/)*)*
* *Knowledge of Newman commands for running a collection (Refer*[***Running Collections using Newman***](https://toolsqa.com/postman/running-collection-using-newman/)*)*

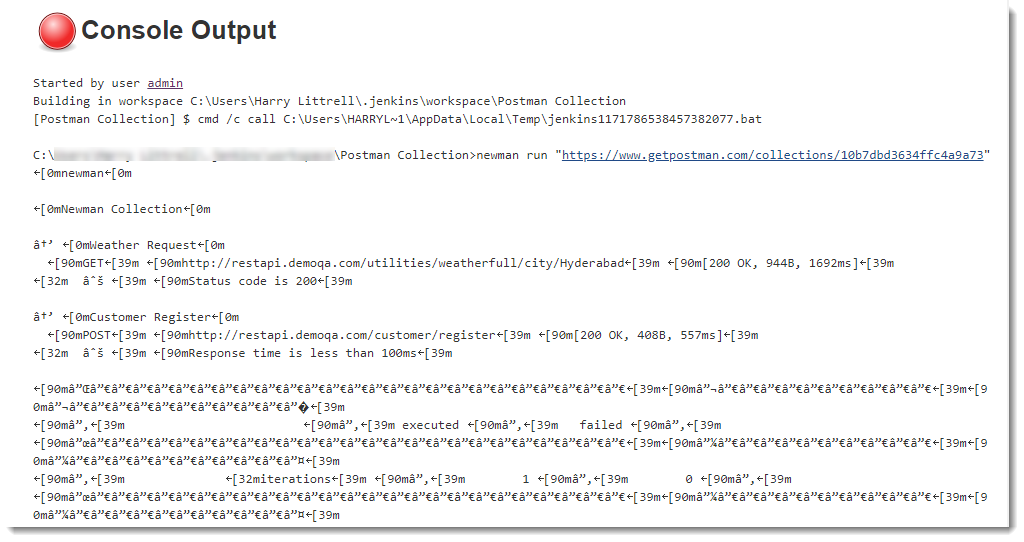
### ***How to Run Postman Collection on Jenkins using Newman Commands?***

1. To run a Collection, go to your ***Postman Collection*** and get the ***Share Link***. ***Refer***[***Running Collections using Newman***](https://toolsqa.com/postman/running-collection-using-newman/)***to learn more***.
2. Go to the ***Jenkins Job*** and under ***build section***, write the command to run the Collection: ***newman run "<link>***"



1. ***Save*** the changes and click on ***build now***. This will start the Jenkins job, which will perform all the actions defined with in the job. As we have specified an action to run the *Postman Collection using Newman command*, this will do the same.
2. Check the ***Console Output*** in the Jenkins.

In my system, I get this output while running the collection. ***This is correct output plus unicode for some of the texts***. If you look carefully at the output, you will notice that it has produce some decent logs as well, which tell the result of the execution. It shows the API detail, status, time taken & Size.



***Note****: You must be familiar with the****Unicode****system in computers.****A Unicode system is the basic standard used for converting the texts for the propagation to another system.****With so many languages, it is hard to make every system compatible to every other language out there. So, a basic standard Unicode system is created for propagation. But here, we see Unicode we don't understand and it really looks very messy. In the next section we will clear it out. If you have got the structured result, you can skip the tutorial or learn for knowledge, in case*.

### ***How to Remove Unicode from Jenkins Console Output logs?***

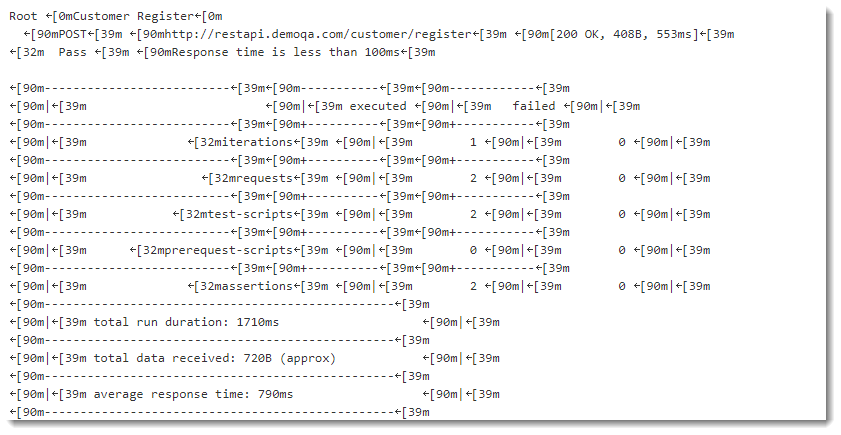
It is very easy to remove Unicode from the Console Output logs. We need to filter out the Unicode symbols, so ***we define a flag along with the command.*** Some people also refer to flags as "***parameters***" in Jenkins. Newman provides the flag ***--disable-unicode*** to remove the Unicode text from the logs.

1.Go to the ***build*** section.

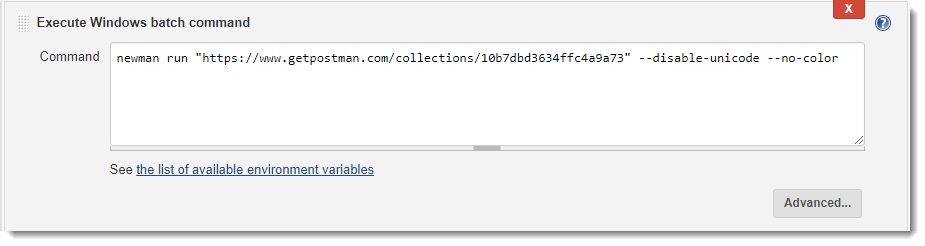
1. Write the flag after the command which is already written in the previous section.



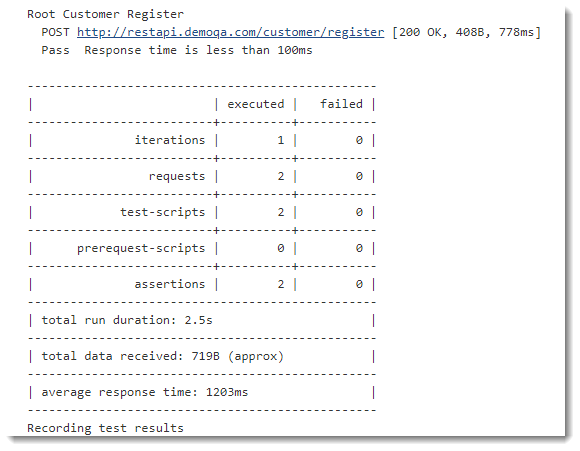
1. ***Save*** t*he changes and click again on* ***Build Now*** to run the new command.
2. Check the *Console Output* now.



Now, ***unicode texts are removed***. But the output in the console is not as tidy as it was when we were running the collection in the shell/command prompt. Some of you must have got the response in a better tabular way. But if not, you can try with adding additional parameter to the previous command ***--no-color:***



***Save****the changes and click again on****Build Now***. Now, the console gives a expected visually structured output.



***Note****: Although --no-color is used to disable the color output, but the compendium of these things is that Jenkins behave differently on different systems*.

### ***What are different Parameters available in Newman?***

Newman provides a rich set of options to customize a run. You can retrieve a list of options by running it with the -h flag. There is no need to learn all of it, but it is better to know the capabilities of these parameters. So that when the need arise, you know what to do and how to do.

Options: Utility: -h, --help output usage information -v, --version output the version number Basic setup: --folder [folderName] Specify a single folder to run from a collection. -e, --environment [file|URL] Specify a Postman environment as a JSON [file] -d, --data [file] Specify a data file to use either json or csv -g, --globals [file] Specify a Postman globals file as JSON [file] -n, --iteration-count [number] Define the number of iterations to run Request options: --delay-request [number] Specify a delay (in ms) between requests [number] --timeout-request [number] Specify a request timeout (in ms) for a request Misc.: --bail Stops the runner when a test case fails --silent Disable terminal output --no-color Disable colored output -k, --insecure Disable strict ssl -x, --suppress-exit-code Continue running tests even after a failure, but exit with code=0 --ignore-redirects Disable automatic following of 3XX responses