

Newman CLI

Prerequisites and Steps to Run Postman Collections Using Newman CLI

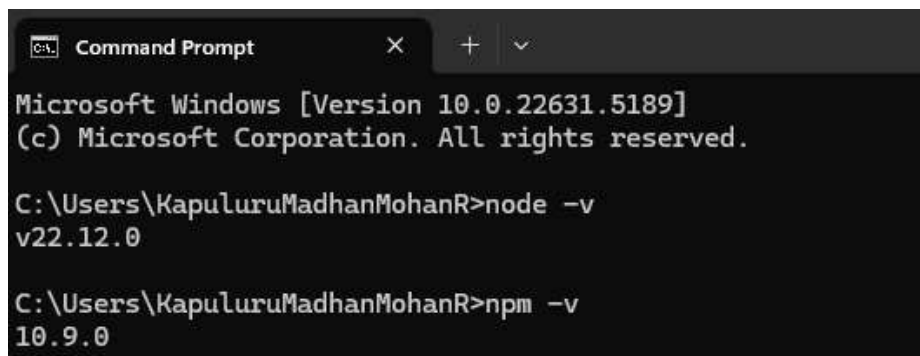
Before you can run and test Postman collections using Newman CLI, ensure you have the following installed

1. **Install Node.js**
 - a. Download and install Node.js from the official Node.js website.
2. **Install Newman**
 - a. Newman is the CLI tool for running Postman collections.
3. **Install newman-reporter-htmlextra**
 - a. This reporter generates detailed HTML reports for Newman runs.
4. **Export Postman Collection.**
5. **Run Collection in Command Prompt (Locally)**
6. **Push Collections to Github**
7. **Run Postman collection in Jenkins (from Github)**

Verify Node.js Installation

→ Once Node.js is installed, open a command prompt or terminal and check if Node.js and npm (Node Package Manager) are installed. Run the following commands

- ◆ node -v
- ◆ npm -v



```
Microsoft Windows [Version 10.0.22631.5189]
(c) Microsoft Corporation. All rights reserved.

C:\Users\KapuluruMadhanMohanR>node -v
v22.12.0

C:\Users\KapuluruMadhanMohanR>npm -v
10.9.0
```

Install Newman and HTML Extra Reporter

→ Run the following commands to install Newman and the htmlextra reporter globally

- ◆ npm install -g newman

```
C:\Users\KapuluruMadhanMohanR>npm install -g newman
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm warn deprecated @faker-js/faker@5.5.3: Please update to a newer version.

added 120 packages in 13s
```

- ◆ npm install -g newman-reporter-htmlextra

```
C:\Users\KapuluruMadhanMohanR>npm install -g newman-reporter-htmlextra
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm warn deprecated @faker-js/faker@5.5.3: Please update to a newer version.

added 166 packages in 20s
```

→ Ignore any warnings that may appear during the installation process.

Export Postman Collections

- Before using Newman, you must export the desired collection and any associated data from Postman.
 - ◆ Open Postman. Select the collection you want to run.
 - ◆ Click on the "Export" option.
 - ◆ Save the collection as a .json file.
- Note
 - ◆ Export additional files, like environment variables or data files (CSV/JSON), if needed.

Run Collections in Command Prompt locally

→ Collection Execution

- ◆ To execute your exported Postman collection locally, use the newman run command.
- ◆ **Command Syntax**
 - `newman run <collection-file> [options]`
- ◆ **Including Data Files and Variables**
 - `-d`: Use this option to specify a data file (e.g., CSV or JSON).
 - `-g`: Use this option to include a global variables file.
 - `-e`: Use this option to include a environmental variables file.
- ◆ **Example Command**
 - `newman run Products.postman_collection.json -d product_data.json -g workspace.postman_globals.json`

→ Generating Reports

- ◆ To generate detailed reports, use the `-r` option with the reporter name.
- ◆ For example, to use the `htmlextra` reporter
 - `newman run Products.postman_collection.json -d product_data.json -g workspace.postman_globals.json -r htmlextra`
- ◆ **htmlextra CLI Options**
 - **Report Path**
 - Default: `newman/` in the current working directory.
 - To specify a custom path:
 - ◆ `--reporter-htmlextra-export ./results/report.html`
 - **Browser Title**
 - Default: `Newman Summary Report`
 - To customize the browser title:
 - ◆ `--reporter-htmlextra-browserTitle "Store API Test Report"`
 - **Main Title**
 - Default: `Newman Run Dashboard`
 - To set a custom main title:
 - ◆ `--reporter-htmlextra-title "Store API Test Summary"`

◆ Example Command

- `newman run Products.postman_collection.json -d product_data.json -g workspace.postman_globals.json -r htmlextra --reporter-htmlextra-export ./results/report.html --reporter-htmlextra-browserTitle "Store API Test Report" --reporter-htmlextra-title "Store API Test Summary"`

Push Collections to GitHub

1. Create a Remote Repository on GitHub

a. Example repository URL:

- i. <https://github.com/Madhan-091296/StoreAPI.git>

2. Initialize a Local Repository

a. `git init`

3. Connect the Local Repository to Remote

- a. `git remote add origin https://github.com/Madhan-091296/StoreAPI.git`

4. Add Files to Staging

a. `git add -A`

5. Commit Files Locally

- a. `git commit -m "committing products"`

6. Push Files to Remote Repository

- a. `git push origin master`

Run Postman Collections in Jenkins (From GitHub)

→ Prerequisite

- ◆ Log in to Jenkins.
- ◆ Install the **NodeJS** and **HTML Publisher** plugins.

→ Steps to Execute

◆ Create a New Job

- In Jenkins, create a new job and select the Freestyle project option.

◆ Configure the Git Repository

- Under the Source Code Management section, provide the Git repository URL to fetch the Postman collection and related files.

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

☐ None

☒ Git ?

Repositories ?

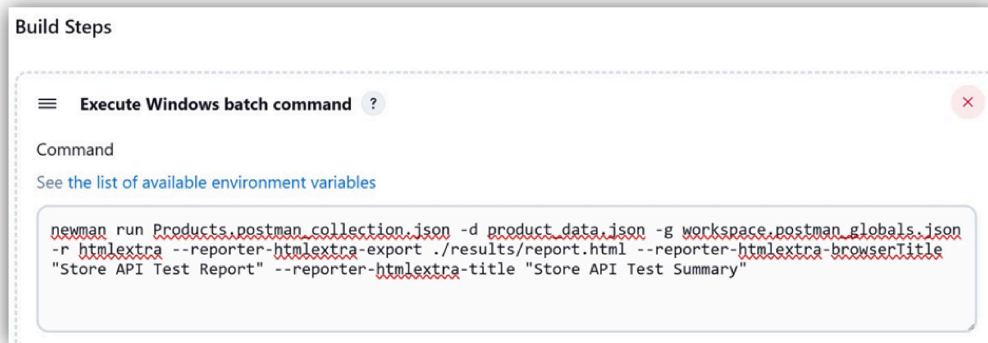
Repository URL ?

<https://github.com/Madhan-091296/StoreAPI.git>

◆ Set Up the Build

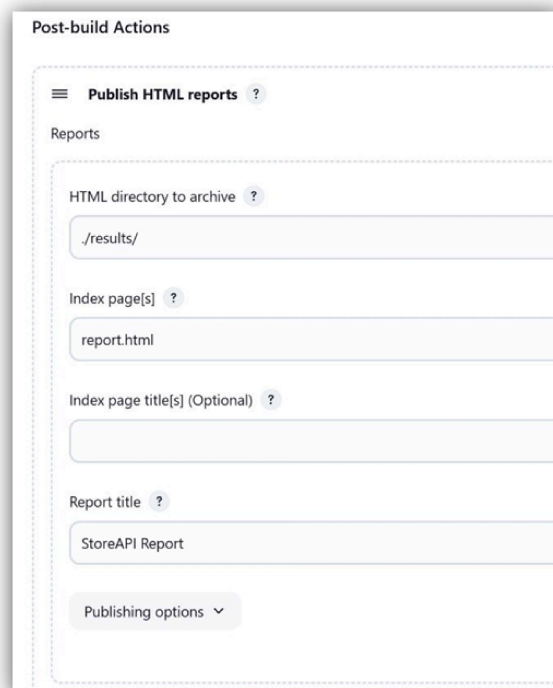
- In the **Build** section
 - Select **Execute Windows batch command** as the build step.

- Add the following script to execute the Postman collection
 - ◆ **newman run Products.postman_collection.json -d product_data.json -g workspace.postman_globals.json -r htmlextra --reporter-htmlextra-export ./results/report.html --reporter-htmlextra-browserTitle "Store API Test Report" --reporter-htmlextra-title "Store API Test Summary"**



◆ **Configure Post-Build Actions:**

- In the **Post-Build Actions** section
 - Select **Publish HTML reports**.
 - Fill in the required fields to point to the HTML report and save the configuration.



◆ **Trigger the Build:**

- Save the job configuration and trigger the build to execute the Postman collection.

→ **Resolving HTML Report Viewing Issues (Optional)**

- ◆ If the HTML report does not display properly (due to a security policy

issue), follow these steps

- Go to **Manage Jenkins** → **Tools and Actions** → **Script Console**.
- Add the following script and run it
 - `System.setProperty("hudson.model.DirectoryBrowserSupport.CSP", "")`
- Return to the project and re-run the job.

Note

→ If **NodeJS** is not installed on your local machine, you can configure the NodeJS platform directly within Jenkins:

- ◆ Go to **Manage Jenkins** → **Tools** → **NodeJS Installations**.
- ◆ Click **Add NodeJS** to configure a new NodeJS installation.

Add NodeJS

NodeJS

Name

mynodejs

☒ Install automatically ?

Install from nodejs.org

Version

NodeJS 23.5.0

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

☐ Force 32bit architecture

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax 'packageName@version'

newman-reporter-htmlextra

Global npm packages refresh hours

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

72

- ◆ Go to Project configuration then add Build Environment.

Build Environment

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☐ Provide Configuration files ?

☐ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☒ Provide Node & npm bin/ folder to PATH

NodeJS Installation

Specify needed nodejs installation where npm installed packages will be provided to the PATH

mynodejs

npmrc file

- use system default -

Cache location

Default (~/.npm or %APP_DATA%\npm-cache)

☐ Terminate a build if it's stuck

☐ With Ant ?