**newman *the cli companion for postman***

Using Newman, one can effortlessly run and test a Postman Collections directly from the command-line.

To run Newman, ensure that you have NodeJS >= v4. A copy of the NodeJS installable can be downloaded from <https://nodejs.org/en/download/package-manager>.

The easiest way to install Newman is using NPM. If you have NodeJS installed, it is most likely that you have NPM installed as well.

$ npm install newman --global;

The newman run command allows you to specify a collection to be run. You can easily export your Postman Collection as a json file from the [Postman App](https://www.getpostman.com/apps) and run it using Newman.

$ newman run examples/sample-collection.json;

## ***Command line Options***

### **newman run <collection-file-source> [options]**

1. **-e <source>, --environment <source>**  
   Specify an environment file path or URL. Environments provide a set of variables that one can use within collections.
2. **-g <source>, --globals <source>**  
   Specify file path or URL for global variables. Global variables are similar to environment variables but has a lower precedence and can be overridden by environment variables having same name.
3. **-d <source>, --iteration-data <source>**  
   Specify a data source file (CSV) to be used for iteration as a path to a file or as a URL.
4. **-n <number>, --iteration-count <number>**  
   Specifies the number of times the collection has to be run when used in conjunction with iteration data file.
5. **--folder <name>**  
   Run requests within a particular folder in a collection.
6. **--export-environment <path>**  
   The path to the file where Newman will output the final environment variables file before completing a run.
7. **--export-globals <path>**  
   The path to the file where Newman will output the final global variables file before completing a run.
8. **--export-collection <path>**  
   The path to the file where Newman will output the final collection file before completing a run.
9. **--timeout-request <ms>**  
   Specify the time (in milliseconds) to wait for requests to return a response.
10. **-k, --insecure**Disables SSL verification checks and allows self-signed SSL certificates.
11. **--ignore-redirects**  
    Prevents newman from automatically following 3XX redirect responses.
12. **--delay-request**  
    Specify the extent of delay between requests (milliseconds).
13. **--bail**  
    Specify whether or not to stop a collection run on encountering the first error.
14. **-x, --suppress-exit-code**  
    Specify whether or not to override the default exit code for the current run.
15. **--no-color**  
    Newman attempts to automatically turn off color output to terminals when it detects the lack of color support. With this property, one can forcibly turn off the usage of color in terminal output for reporters and other parts of Newman that output to console.
16. **--disable-unicode**  
    Specify whether or not to force the unicode disable option. When supplied, all symbols in the output will be replaced by their plain text equivalents.

#### **Configuring Reporters**

Reporters provide information about the current collection run in a format that is easy to both: disseminate and assimilate.

1. **-r <reporter-name>, --reporters <reporter-name>**  
   Specify one reporter name as string or provide more than one reporter name as a comma separated list of reporter names. Available reporters are: cli, json, html and junit. [ junit reports are in xml ]  
     
   [NOTE: Spaces should **not** be used between reporter names / commas whilst specifying a comma separted list of reporters. For instance:  
     
   ✅ -r html,cli,json,junit   
   ❌ -r html, cli , json,junit ]
2. **--reporter-{{reporter-name}}-{{reporter-option}}**  
   When multiple reporters are provided, if one needs to specifically override or provide an option to one reporter, this is achieved by prefixing the option with --reporter-{{reporter-name}}-.  
     
   Example: --reporters cli,html --reporter-cli-silent would silence the CLI reporter only.
3. **--reporter-{{reporter-options}**}  
   If more than one reporter accepts the same option name, they can be provided using the common reporter option syntax.   
   For example, ... --reporters cli,html --reporter-silent passes the silent: true option to both HTML and CLI reporter.

##### ***CLI reporter options***

These options are supported by the CLI reporter, use them with appropriate argument switch prefix. For example, the option no-summary can be passed as --reporter-no-summary or --reporter-cli-no-summary.

CLI reporter is enabled by default, you do not need to specifically provide the same as part of --reporters option. However, enabling one or more of the other reporters will result in no CLI output. Explicitly enable the CLI option in such a scenario.

| **CLI Option** | **Description** |
| --- | --- |
| --reporter-cli-silent | The CLI reporter is internally disabled and you see no output to terminal. |
| --reporter-cli-no-summary | The statistical summary table is not shown. |
| --reporter-cli-no-failures | This prevents the run failures from being separately printed. |
| --reporter-cli-no-assertions | This turns off the request-wise output as they happen. |
| --reporter-cli-no-console | This turns off the output of console.log (and other console calls) from collection's scripts. |

##### ***JSON reporter options***

The built-in JSON reporter is useful in producing a comprehensive output of the run summary. It takes the path to the file where to write the file. The content of this file is exactly same as the summary parameter sent to the callback when Newman is used as a library.

To enable JSON reporter, provide --reporters json as a CLI option.

| **CLI Option** | **Description** |
| --- | --- |
| --reporter-json-export <path> | Specify a path where the output JSON file will be written to disk. If not specified, the file will be written to newman/ in the current working directory. |

##### ***HTML reporter options***

The built-in HTML reporter produces and HTML output file outlining the summary and report of the Newman run. To enable the HTML reporter, provide --reporters html as a CLI option.

| **CLI Option** | **Description** |
| --- | --- |
| --reporter-html-export <path> | Specify a path where the output HTML file will be written to disk. If not specified, the file will be written to newman/ in the current working directory. |
| --reporter-html-template <path> | Specify a path to the custom template which will be used to render the HTML report. This option depends on --reporter html and --reporter-html-export being present in the run command. If this option is not specified, the [default template](https://github.com/postmanlabs/newman/blob/develop/lib/reporters/html/template-default.hbs) is used |

Custom templates (currently handlebars only) can be passed to the HTML reporter via --reporter-html-template <path> with --reporters html and --reporter-html-export. The [default template](https://github.com/postmanlabs/newman/blob/develop/lib/reporters/html/template-default.hbs) is used in all other cases.

##### ***JUNIT/XML reporter options***

Newman can output a summary of the collection run to a JUnit compatible XML file. To enable the JUNIT reporter, provide --reporters junit as a CLI option.

| **CLI Option** | **Description** |
| --- | --- |
| --reporter-junit-export <path> | Specify a path where the output XML file will be written to disk. If not specified, the file will be written to newman/ in the current working directory. |