Newman Postman collection runner

CONCEPTS

Export the collection (and environment) from Postman as JSON first; Newman runs those files, it does not edit them.

Postman export collection.json
↓ newman run
Newman run requests + scripts
↓ assertions
Results pass / fail
↓ -r reporters
Reports cli, json, junit
↓ exit code
CI job red or green

Postman now steers users to the Postman CLI; Newman still runs v2 collections but is no longer actively developed.

Passing -r replaces the default cli reporter. Add cli to the list to keep the console output.

INSTALL

npm install -g newman
Install globally (Node 16+)

newman --version
Print the version

brew install newman
Install with Homebrew

npm install -g newman-reporter-htmlextra
Add the HTML Extra reporter

docker run -t postman/newman run *<collection.json>*
Run the official Docker image

newman run --help
List every run option

RUN

newman run *<collection.json>*
Run a collection file

newman run *<url>*
Run a collection from a URL

newman run *<collection.json>* **-e** *<env.json>*
Use an environment file

newman run *<collection.json>* **-g** *<globals.json>*
Use a globals file

RUN (cont.)

newman run *<collection.json>* **-n** *<count>*
Repeat the run N times

newman run *<collection.json>* **--folder** *<name>*
Run only one folder or request

newman run *<collection.json>* **--bail**
Stop at the first failed test

newman run *<collection.json>* **--bail folder**
Stop after a folder with failures

newman run *<collection.json>* **-x**
Exit 0 even if tests fail

newman run *<collection.json>* **--verbose**
Print full request and response data

DATA AND VARIABLES

newman run *<c>* **-d** *<data.csv>*
Iterate over CSV or JSON rows

newman run *<c>* **--env-var** "*<name>*=*<value>*"
Override an environment variable

newman run *<c>* **--global-var** "*<name>*=*<value>*"
Override a global variable

newman run *<c>* **--export-environment** *<file>*
Save the environment after the run

REPORTERS

newman run *<c>* **-r cli,json**
Choose reporters (comma list)

newman run *<c>* **-r junit**
Write a JUnit XML report

newman run *<c>* **-r htmlextra**
Write an HTML report (plugin)

newman run *<c>* **-r json --reporter-json-export** *<file>*
Set the JSON report path

newman run *<c>* **-r junit --reporter-junit-export** *<file>*
Set the JUnit report path

newman run *<c>* **-r htmlextra --reporter-htmlextra-export** *<file>*
Set the HTML report path

REPORTERS (cont.)

newman run *<c>* **--reporter-cli-silent**
Hide cli output

newman run *<c>* **--color off**
Disable colours (log files)

TIMEOUTS AND NETWORK

newman run *<c>* **--delay-request** *<ms>*
Wait between requests

newman run *<c>* **--timeout** *<ms>*
Limit the whole run time

newman run *<c>* **--timeout-request** *<ms>*
Limit each request

newman run *<c>* **-k**
Skip TLS certificate checks

NODE.JS LIBRARY

Run from a script

```
const newman = require('newman');
newman.run({
  collection: require('./
collection.json'),
  reporters: 'cli'
}, (err) ⇒ { if (err) throw
err; });
```

RECIPES

CI run with reports

```
newman run api.json -e
staging.json \
  -r cli,junit \
  --reporter-junit-export
results/junit.xml \
  --bail
```

Data-driven run

```
# users.csv columns become
{{variables}}
newman run api.json -d users.csv
-n 3 \
  --env-var "baseUrl=https://api.
example.com"
```

Fresh token, saved environment

```
newman run login.json -e dev.json
\
  --export-environment dev.json
newman run api.json -e dev.json
```

Run in Docker

```
docker run --rm -v "$PWD:/etc/
newman" \
  postman/newman:alpine run
api.json -e env.json \
  -r cli,junit
```