

Automating Workflows

for

API Testing



Dec 2024

Mohsen Madani & Newsha Amiri

Why?

Response Time



Reliability



Postman

- Creating API test **collections**
- Writing **test** scripts





Newman

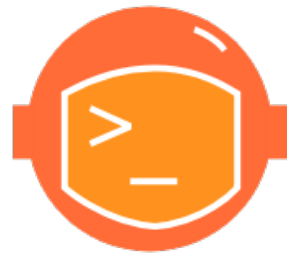
- Running collections via CLI

```
$ newman run examples/sample-collection.json -r cli,json
```

collection

reporters

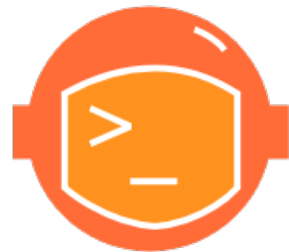
Newman



- Flow

```
1 for (const file of files) {
2   try {
3     logger.info({
4       Scope: 'FileReader',
5       Message: `Reading JSON file: ${file}`,
6     })
7   } catch (err) {
8     logger.error({
9       Scope: 'FileReader',
10      Message: `Error parsing JSON file: ${file}`,
11      Error: err?.message,
12    })
13  }
14  await runNewman(file)
15  await uploadToMinio(file)
16 }
17 postToMattermost(files)
```

Newman



- Run Newman

```
1 newman.run({  
2   collection: require(collectionPath),  
3   environment: require(CONSTANTS.NEWMAN.ENVIRONMENT_PATH),  
4   reporters: CONSTANTS.NEWMAN.REPORTERS,  
5   iterationCount: 3,  
6   delayRequest: 1000  
7 })
```

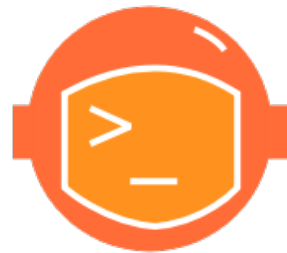
Newman



- Upload to MinIO

```
1 minio
2   .fPutObject(
3     CONSTANTS.MINIO.BUCKET_NAME,
4     `${CONSTANTS.MINIO.REPORT_PATH}/${
5       process.env.CI_PIPELINE_ID
6     }/${CONSTANTS.NEWMAN.EXPORT_FILE(collectionFileWithoutExt)}.html`,
7     `${CONSTANTS.NEWMAN.EXPORT_PATH(collectionFileWithoutExt)}.html`,
8     CONSTANTS.MINIO.HTML_METADATA
9   )
10  .then(() => {
11    logger.success({
12      Scope: 'MinIO',
13      Message: 'Uploaded successfully',
14    })
15    resolve()
16  })
17  .catch((e) => {
18    logger.error({
19      Scope: 'MinIO',
20      Error: e?.message,
21    })
22  })
```

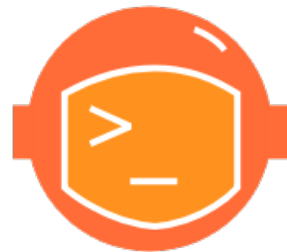
Newman



- Post to Mattermost

```
1 mattermost.createPost({
2   channel_id: process.env.MATTERMOST_CHANNEL_ID,
3   message,
4   metadata:
5     failuresLength >= 10
6     ? {
7       priority: {
8         priority: PostPriority.URGENT,
9       },
10    }
11    : undefined,
12   props: {
13     attachments: [
14       {
15         title: `📎 ${CONSTANTS.MATTERMOST.ATTACHMENT_TITLE}`,
16         text: collectionFilesWithoutExt
17         .map(
18           (collectionFile) =>
19             `[${collectionFile}](${CONSTANTS.MINIO.END_POINT}/${CONSTANTS.MINIO.BUCKET_NAME
20             }/${CONSTANTS.MINIO.REPORT_PATH}/${process.env.CI_PIPELINE_ID
21             }/${CONSTANTS.NEWMAN.EXPORT_FILE(collectionFile)}.html)`
22           .join('\n\n'),
23         color: CONSTANTS.MATTERMOST.ERROR_COLOR,
24       },
25     ],
26   },
27 },
28 },
29 },
30 })
```

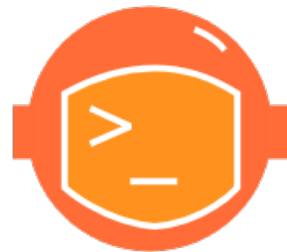

Newman



- Run pipeline

```
1 production:
2   stage: build
3   needs:
4     - job: 'install-dependencies'
5   rules:
6     - if: $CI_PIPELINE_SOURCE == "push"
7     - if: $CI_PIPELINE_SOURCE == "schedule" && $BUILD_ENV ==
      "production"
8   before_script:
9     - rm -rf "newman/"
10    - mkdir "newman"
11  script:
12    - yarn workspace newman-reporter report
13    - mv apps/newman-reporter/exports/ "newman/exports"
14    - mv apps/newman-reporter/collections/ "newman/collections"
15    - mv apps/newman-reporter/environments/ "newman/environments"
16  artifacts:
17    when: always
18    name: "$CI_PIPELINE_ID"
19    paths:
20      - "newman/"
```

Newman



- Scheduled pipeline

Description

Run production tests every hour

Cron timezone

[UTC+3.5] Tehran

Interval Pattern

- ☐ Every day (at 2:02am)
- ☐ Every week (Sunday at 2:02am)
- ☐ Every month (Day 26 at 2:02am)
- ☒ Custom

0 * * * *

Set a custom interval with Cron syntax. [What is Cron syntax?](#)

Select target branch or tag

main

Variables

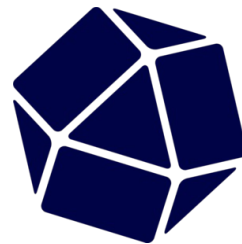
Variable

BUILD_ENV





InfluxDB



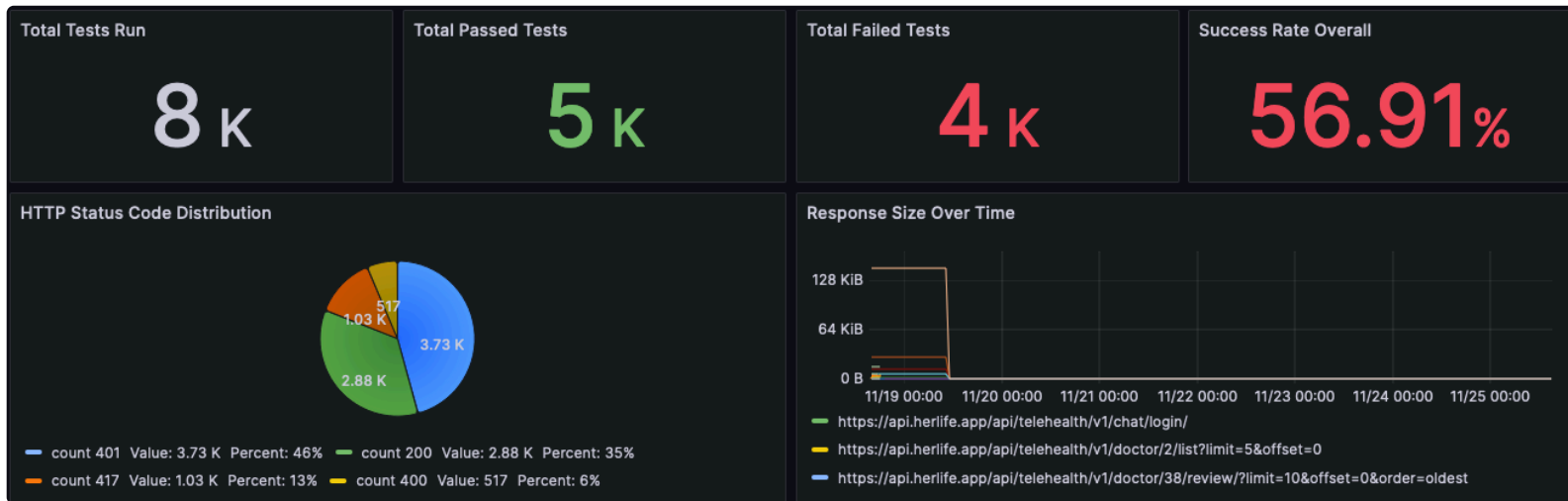
- Store results and metrics

request_name	response_size	response_time	skipped_count	status	test_status
group string	group string	group string	group string	group string	group string
Banner	183	16	0	Unauthorized	PASS
Banner	183	18	0	Unauthorized	PASS
Banner	183	19	0	Unauthorized	PASS
Banner	183	20	0	Unauthorized	PASS
Banner	183	21	0	Unauthorized	PASS
Banner	183	17	0	Unauthorized	PASS
Banner	183	18	0	Unauthorized	PASS

Grafana



- Visualizing test results



Grafana



- Response Time Over Time per URL

```
from(bucket: "Herlife QA API Test")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r._measurement == "api_results" and exists
r.response_time)
  |> filter(fn: (r) => r.url == "${urlVariable}" or "${urlVariable}" ==
"All")
  |> map(fn: (r) => ({
    _time: r._time,
    _value: float(v: r.response_time),
    url: r.url
  })))
```

URL

Grafana

- Define URL variable



Q&A

Ask me question if you have any

The END

Mohsen Madani



Newsha Amiri



References

- Postman ([link](#))
- Newman ([link](#))
- InfluxDB ([link](#))
- Grafana ([link](#))