Log Aggregation Project

Cribl Take Home Project submitted by Yves do Régo

Take Home Timeline and Design

Friday April 22rd, 2022

Got the project objectives and requirements. Decided to approach it like a Hackathon.

Saturday April 23rd, 2022

* Read requirements and went on an 8k walk thinking about design approaches.

* Asked for clarifications on some requirements.

* Started on Events in reverse Order

* API design end-user
* Docker/Compose
* Coding/Debugging (API)

Sunday April 24th, 2022

* Coding/Debugging

* 10k Bike ride

* Lots of issues with UT environment (DI)…

* Design and Coding of Bonus Requirements

Monday April 25th, 2022

Put together design document/slides.

* More coding/debugging refinements.

* Got feedback from Harry

* Implements events filtering.

* Add Error handling for cluster get-events.

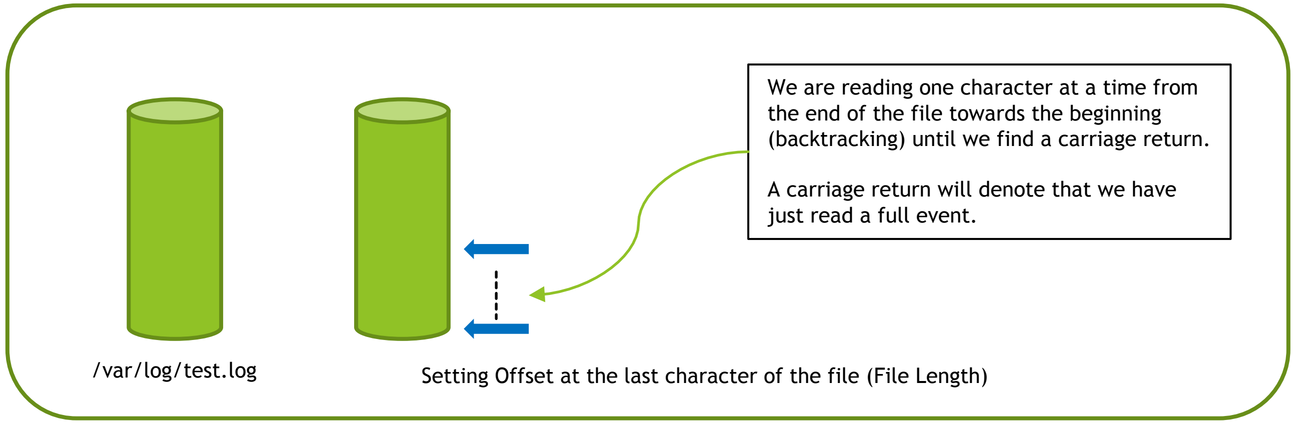
* Finalized documentation

Design

Design Approach

Events in reverse time order

- Out of the requirements the first one to tackle was the events in reverse time order.
  - Main consideration was finding an efficient approach memory and performance wise. The Solution was to use a random access file, and read from the end of the file while back tracking.



We are reading one character at a time from the end of the file towards the beginning (backtracking) until we find a carriage return.
A carriage return will denote that we have just read a full event.
/var/log/test.log
Setting Offset at the last character of the file (File Length)

Design Approach
REST API

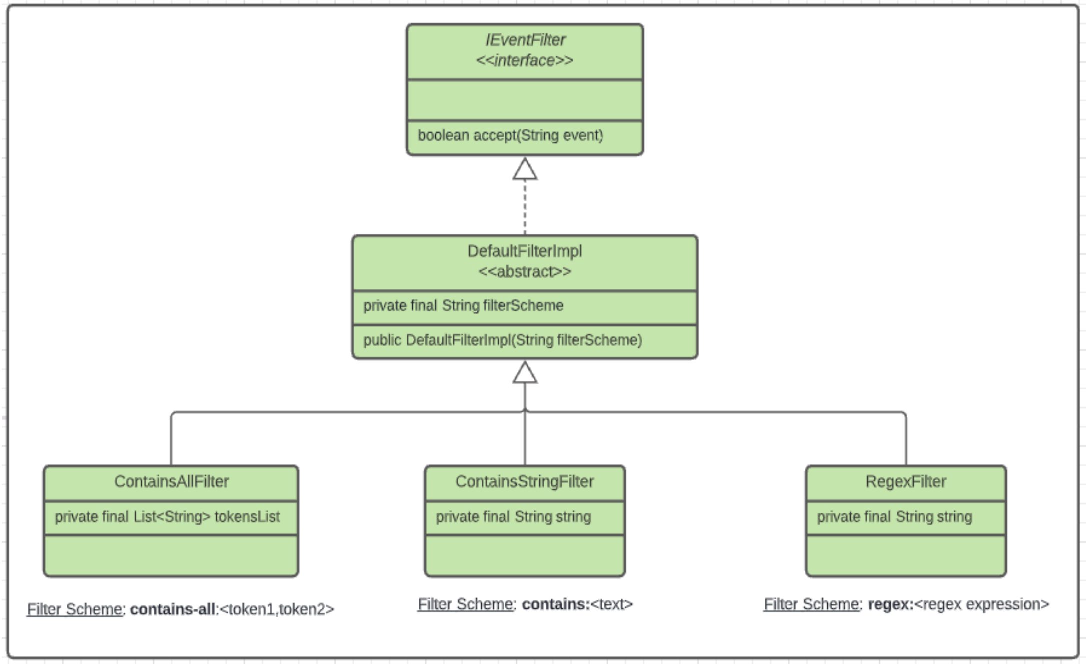
- get-files endpoint: We need to offer an endpoint to list the files present in a given directory. The returned information should be sufficient to facilitate the selection of a specified file.
  - Operation HTTP GET
    - Parameters:
      - Absolute path for the directory.
      - Only return files matching specified extensions. (Optional parameter)

- get-events endpoint: We relaxed the restriction around only retrieval from /var/log/ and request the target directory from the user.
  - Operation HTTP GET
    - Parameters:
      - Absolute path for the file to process
      - Maximum number of Events to return. [1-250]
      - Filter for events. (Optional parameter)

Design Approach
Event Filter

- Chose an implementation based on inheritance to be able to have different type of filters.
- 3 implementations were done for the take home project.
  - **contains-all:** Event must contain all the provided tokens.
  - **contains:** Event must contain the provided text.
  - **regex:** Event must evaluate to true for the passed regex expression.

Usage Example Query Parameter
filter=contains:test


IEventFilter
<<interface>>
boolean accept(String event)
DefaultFilterImpl
<<abstract>>
private final String filterScheme
public DefaultFilterImpl(String filterScheme)
ContainsAllFilter
private final List<String> tokensList
ContainsStringFilter
private final String string
RegexFilter
private final String string
Filter Scheme: contains-all:<token1,token2>
Filter Scheme: contains:<text>
Filter Scheme: regex:<regex expression>

Development

Development
Technology Stack and Solution Deployment

- We are choosing to develop the solution using Java and SpringBoot as the framework.
  - Project is using Java 11+, maven based and using vscode IDE
  - For some of the REST parameters validation we've made used of the SpringBoot embedded validation framework (@NotNull, @Min, @Max)

- The solution will be deployed using docker image and docker-compose.

Development

Java Project Structure

- REST Controller implementation
  - ControllerAdvice provides global error handling capabilities
  - Controller delegates to service layer for processing.

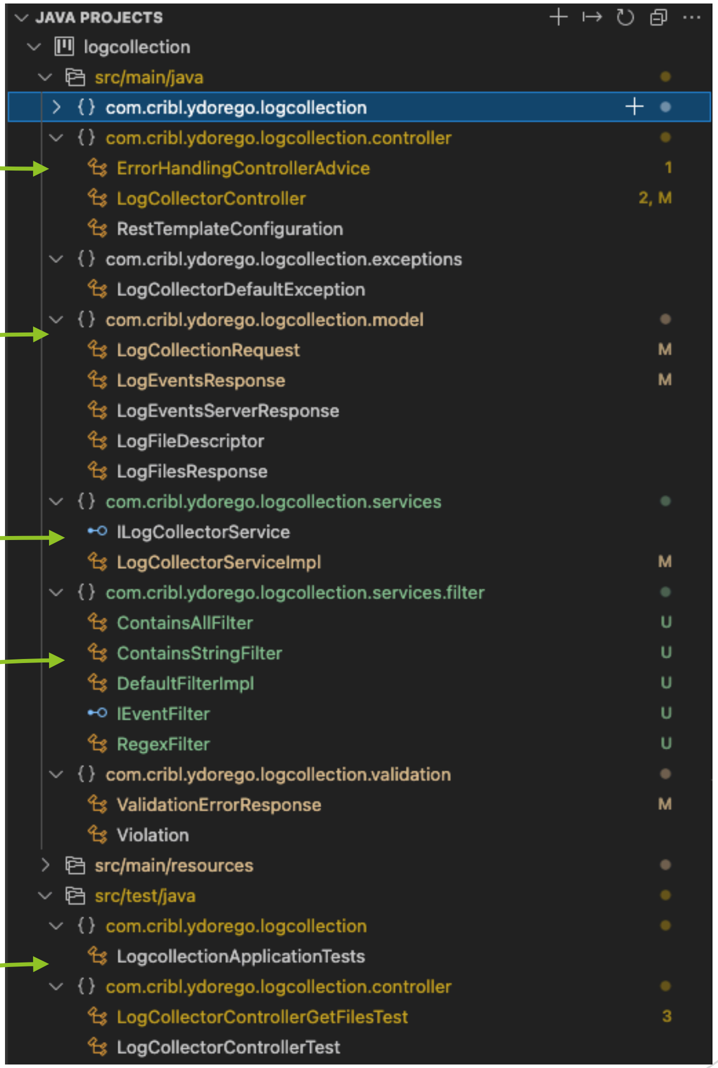
- Model classes uses by the API layer for request/response

- Service layer responsible for the implementation of the business logic.
  - Interface based development

- Filter implementation

- Validation provides uniform error format and handling.

- Unit tests



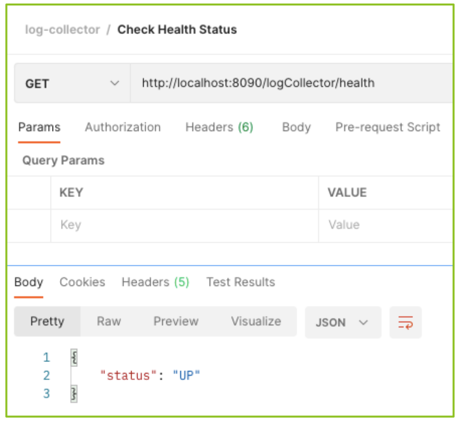
```
JAVA PROJECTS
  logcollection
    src/main/java
      com.cribl.ydorego.logcollection
      com.cribl.ydorego.logcollection.controller
        ErrorHandlingControllerAdvice                 1
        LogCollectorController                        2, M
        RestTemplateConfiguration
      com.cribl.ydorego.logcollection.exceptions
        LogCollectorDefaultException
      com.cribl.ydorego.logcollection.model
        LogCollectionRequest                          M
        LogEventsResponse                             M
        LogEventsServerResponse
        LogFileDescriptor
        LogFilesResponse
      com.cribl.ydorego.logcollection.services
        ILogCollectorService
        LogCollectorServiceImpl                       M
      com.cribl.ydorego.logcollection.services.filter
        ContainsAllFilter                             U
        ContainsStringFilter                          U
        DefaultFilterImpl                             U
        IEventFilter                                  U
        RegexFilter                                   U
      com.cribl.ydorego.logcollection.validation
        ValidationErrorResponse                       M
        Violation
    src/main/resources
    src/test/java
      com.cribl.ydorego.logcollection
        LogcollectionApplicationTests
      com.cribl.ydorego.logcollection.controller
        LogCollectorControllerGetFilesTest            3
        LogCollectorControllerTest
```

Development

Log-Collector-Server Execution

Start a single node log-collector-server: docker-compose -f docker/docker-compose.yml up

Check application status

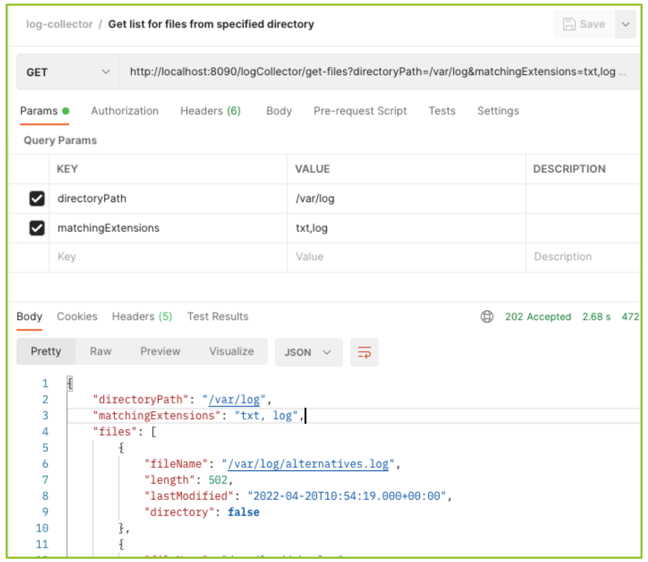


```
GET  http://localhost:8090/logCollector/health
```

KEY	VALUE
Key	Value

```
{
    "status": "UP"
}
```

Get list of files from directory

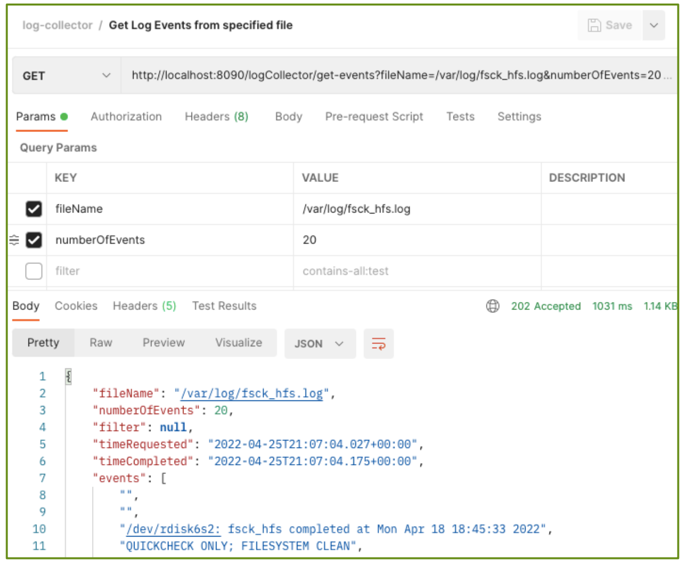


```
GET  http://localhost:8090/logCollector/get-files?directoryPath=/var/log&matchingExtensions=txt,log ...
```

	KEY	VALUE	DESCRIPTION
✓	directoryPath	/var/log	
✓	matchingExtensions	txt,log	
	Key	Value	Description

```
{
    "directoryPath": "/var/log",
    "matchingExtensions": "txt, log",
    "files": [
        {
            "fileName": "/var/log/alternatives.log",
            "length": 502,
            "lastModified": "2022-04-20T10:54:19.000+00:00",
            "directory": false
        },
        {
```

Get events from specified file



```
GET  http://localhost:8090/logCollector/get-events?fileName=/var/log/fsck_hfs.log&numberOfEvents=20 ...
```

	KEY	VALUE	DESCRIPTION
✓	fileName	/var/log/fsck_hfs.log	
✓	numberOfEvents	20	
	filter	contains-all:test	

```
{
    "fileName": "/var/log/fsck_hfs.log",
    "numberOfEvents": 20,
    "filter": null,
    "timeRequested": "2022-04-25T21:07:04.027+00:00",
    "timeCompleted": "2022-04-25T21:07:04.175+00:00",
    "events": [
        "",
        "",
        "/dev/rdisk6s2: fsck_hfs completed at Mon Apr 18 18:45:33 2022",
        "QUICKCHECK ONLY; FILESYSTEM CLEAN",
```

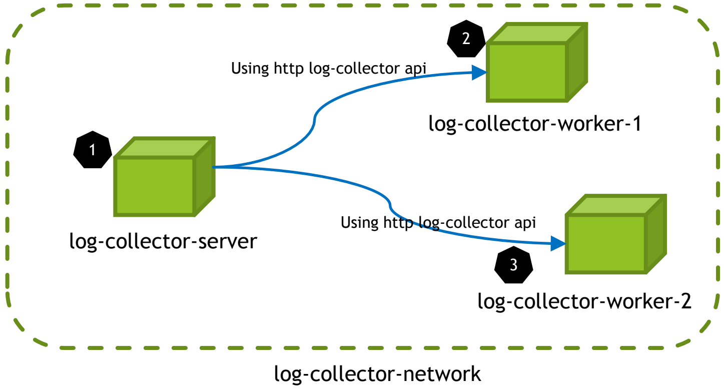
Bonus Requirement

Bonus Requirement

Main log-collector-server delegate to workers nodes.

- We chose an approach were we made each log-collector-server instance have the capability to delegate requests to workers present in the same network.
  - Add a new get-events-from-server API endpoint.
    - Operation HTTP GET
      - Parameters:
        - Absolute path for the file to process
        - Maximum number of Events to return. [1-250]
        - List of servers to get data from. (Comma separated list of workers hostname)
        - Filter for events. (Optional parameter)

1. Client request logs from worker1 and worker2 on log-collector-server API
2. Log collector server loops through server list:
   1. Request/Response from worker1
   2. Request/Response from worker2
3. Return aggregated response to client.



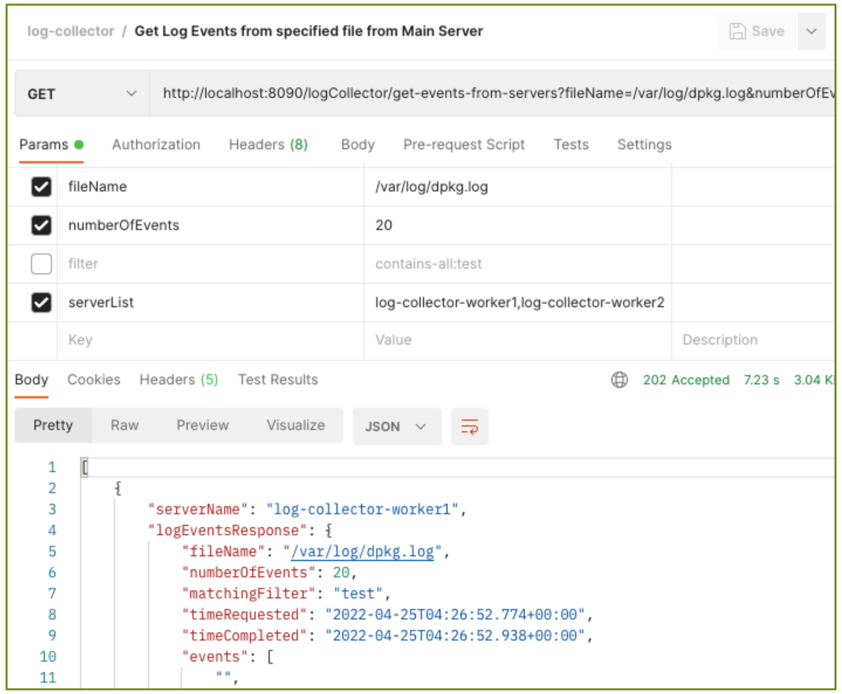
Bonus Requirement

Log-Collector-Server Execution

Start a single node log-collector-server: docker-compose -f docker/docker-compose-cluster.yml up

This will start a deployment containing 3 log-collector-nodes with each of them having the same capability to act as a master nodes. But for this exercise only the server will have its port 8090 expose to receive ”public” api requests.

Get events from specified file and from specified servers



log-collector / **Get Log Events from specified file from Main Server**

GET http://localhost:8090/logCollector/get-events-from-servers?fileName=/var/log/dpkg.log&numberOfEv

	Key	Value	Description
✓	fileName	/var/log/dpkg.log	
✓	numberOfEvents	20	
☐	filter	contains-all:test	
✓	serverList	log-collector-worker1,log-collector-worker2	

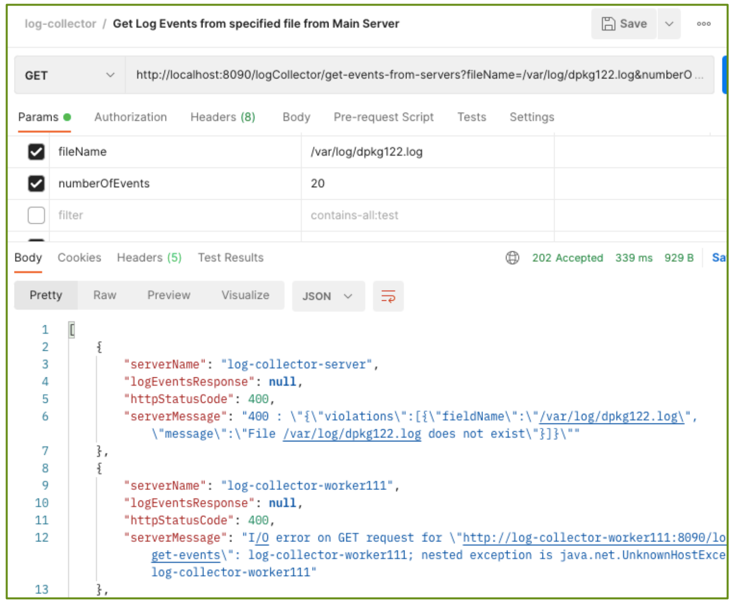
202 Accepted 7.23 s 3.04 K

```
[
    {
        "serverName": "log-collector-worker1",
        "logEventsResponse": {
            "fileName": "/var/log/dpkg.log",
            "numberOfEvents": 20,
            "matchingFilter": "test",
            "timeRequested": "2022-04-25T04:26:52.774+00:00",
            "timeCompleted": "2022-04-25T04:26:52.938+00:00",
            "events": [
                "",
```

Bonus Requirement

Log-Collector-Server Execution (Error Handling)

Get events from specified file and from specified servers: Unknown file and Unknown server



log-collector / Get Log Events from specified file from Main Server

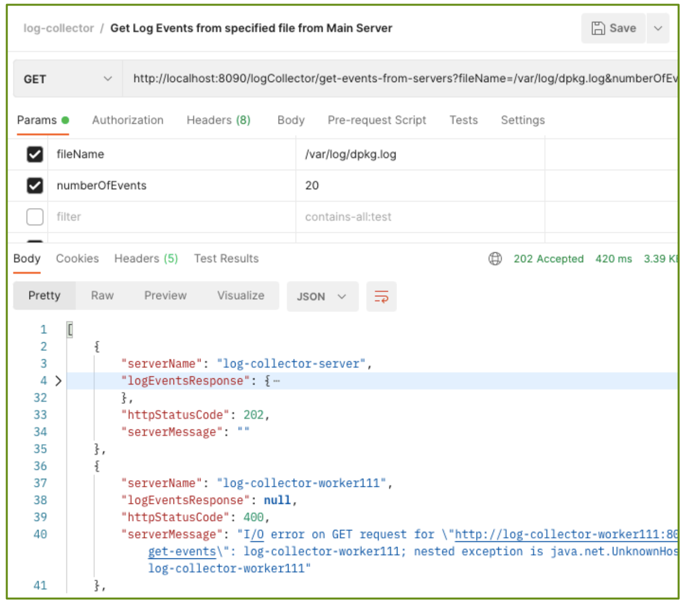
GET http://localhost:8090/logCollector/get-events-from-servers?fileName=/var/log/dpkg122.log&numberO ...

✓	fileName	/var/log/dpkg122.log
✓	numberOfEvents	20
	filter	contains-all:test

202 Accepted 339 ms 929 B

```
[
    {
        "serverName": "log-collector-server",
        "logEventsResponse": null,
        "httpStatusCode": 400,
        "serverMessage": "400 : \"{\"violations\":[{\"fieldName\":\"/var/log/dpkg122.log\",
            \"message\":\"File /var/log/dpkg122.log does not exist\"}]}\""
    },
    {
        "serverName": "log-collector-worker111",
        "logEventsResponse": null,
        "httpStatusCode": 400,
        "serverMessage": "I/O error on GET request for \"http://log-collector-worker111:8090/lo
            get-events\": log-collector-worker111; nested exception is java.net.UnknownHostExce
            log-collector-worker111"
    },
```

Get events from specified file and from specified servers: Unknown server



log-collector / Get Log Events from specified file from Main Server

GET http://localhost:8090/logCollector/get-events-from-servers?fileName=/var/log/dpkg.log&numberOfEv

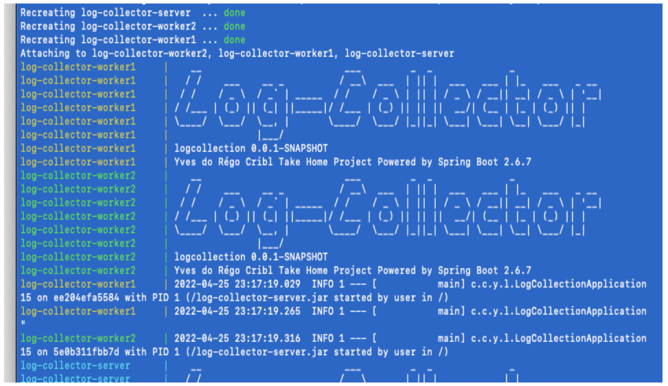
✓	fileName	/var/log/dpkg.log
✓	numberOfEvents	20
	filter	contains-all:test

202 Accepted 420 ms 3.39 K

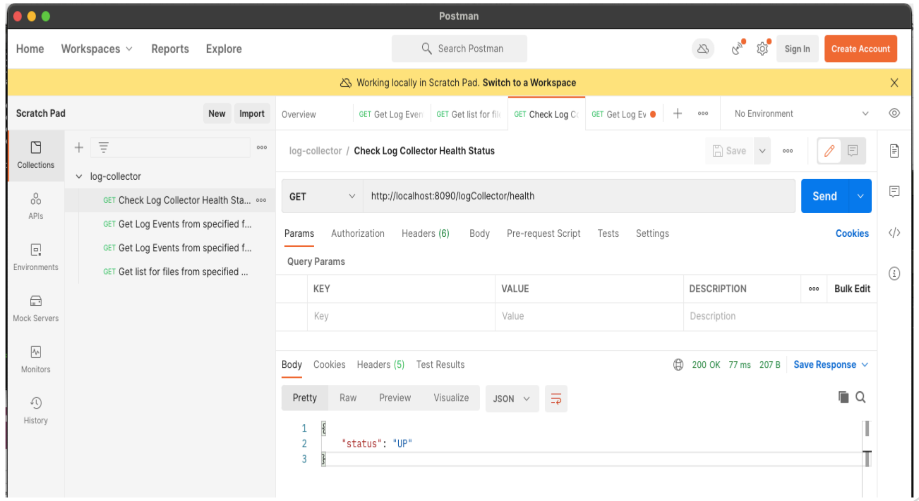
```
[
    {
        "serverName": "log-collector-server",
        "logEventsResponse": {…
        },
        "httpStatusCode": 202,
        "serverMessage": ""
    },
    {
        "serverName": "log-collector-worker111",
        "logEventsResponse": null,
        "httpStatusCode": 400,
        "serverMessage": "I/O error on GET request for \"http://log-collector-worker111:80
            get-events\": log-collector-worker111; nested exception is java.net.UnknownHos
            log-collector-worker111"
    },
```

Conclusion

- The project is overall functional, in a real project I would have added more unit/integration tests to show code coverage etc…



```
Recreating log-collector-server  ... done
Recreating log-collector-worker2 ... done
Recreating log-collector-worker1 ... done
Attaching to log-collector-worker2, log-collector-worker1, log-collector-server
log-collector-worker1     |    __                   ___      _ _           _
log-collector-worker1     |   / /    ___   __ _    / __\___ | | | ___  ___| |_ ___  _ __
log-collector-worker1     |  / /    / _ \ / _` |__/ /  / _ \| | |/ _ \/ __| __/ _ \| '__|
log-collector-worker1     | / /___ | (_) | (_| |_/ /__| (_) | | |  __/ (__| || (_) | |
log-collector-worker1     | \____/  \___/ \__, | \____/\___/|_|_|\___|\___|\__\___/|_|
log-collector-worker1     |               |___/
log-collector-worker1     | logcollection 0.0.1-SNAPSHOT
log-collector-worker1     | Yves do Régo Cribl Take Home Project Powered by Spring Boot 2.6.7
log-collector-worker2     |    __                   ___      _ _           _
log-collector-worker2     |   / /    ___   __ _    / __\___ | | | ___  ___| |_ ___  _ __
log-collector-worker2     |  / /    / _ \ / _` |__/ /  / _ \| | |/ _ \/ __| __/ _ \| '__|
log-collector-worker2     | / /___ | (_) | (_| |_/ /__| (_) | | |  __/ (__| || (_) | |
log-collector-worker2     | \____/  \___/ \__, | \____/\___/|_|_|\___|\___|\__\___/|_|
log-collector-worker2     |               |___/
log-collector-worker2     | logcollection 0.0.1-SNAPSHOT
log-collector-worker2     | Yves do Régo Cribl Take Home Project Powered by Spring Boot 2.6.7
log-collector-worker1     | 2022-04-25 23:17:19.029  INFO 1 --- [           main] c.c.y.l.LogCollectionApplication
15 on ee204efa5584 with PID 1 (/log-collector-server.jar started by user in /)
log-collector-worker1     | 2022-04-25 23:17:19.265  INFO 1 --- [           main] c.c.y.l.LogCollectionApplication
"
log-collector-worker2     | 2022-04-25 23:17:19.316  INFO 1 --- [           main] c.c.y.l.LogCollectionApplication
15 on 5e0b311fbb7d with PID 1 (/log-collector-server.jar started by user in /)
log-collector-server      |    __                   ___      _ _           _
log-collector-server      |   / /                  / __\    | | |         | |
```



log-collector
- GET Check Log Collector Health Sta...
- GET Get Log Events from specified f...
- GET Get Log Events from specified f...
- GET Get list for files from specified ...

log-collector / Check Log Collector Health Status

GET http://localhost:8090/logCollector/health

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body — 200 OK 77 ms 207 B

```
{
    "status": "UP"
}
```