QA Automation Engineer - Test Project BUGS

1. Unable to execute the POST endpoint with same groovy code fails.

Priority: High

Description:

When the same payload is used twice then the API is not functioning properly. It throws internal server error.

Steps to reproduce:

1. Make sure the server is up and running. Use postman or any other API client to post the request.
2. Do a POST operation for the URI localhost:8080/groovy/submit with payload as {"code":"2+3"} and use basic auth user_1/pass_1
3. Now use the same payload that was used above and do a POST operation.

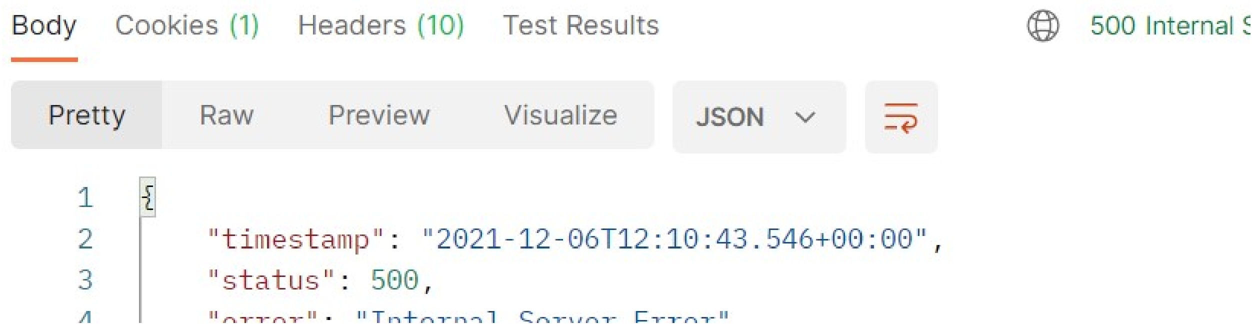
Expected Result:

Should be with 200 ok status and an ID should be generated.

Actual Result:

"status": 500,

"error": "Internal Server Error",



```
{
    "timestamp": "2021-12-06T12:10:43.546+00:00",
    "status": 500,
    "error": "Internal Server Error"
```

2. Without authentication details, POST endpoint creates success response and id is getting generated.

Priority: High

Description:

When the payload without having authentication details in header is submitted with a POST operation, ID is getting created.

Steps to reproduce:

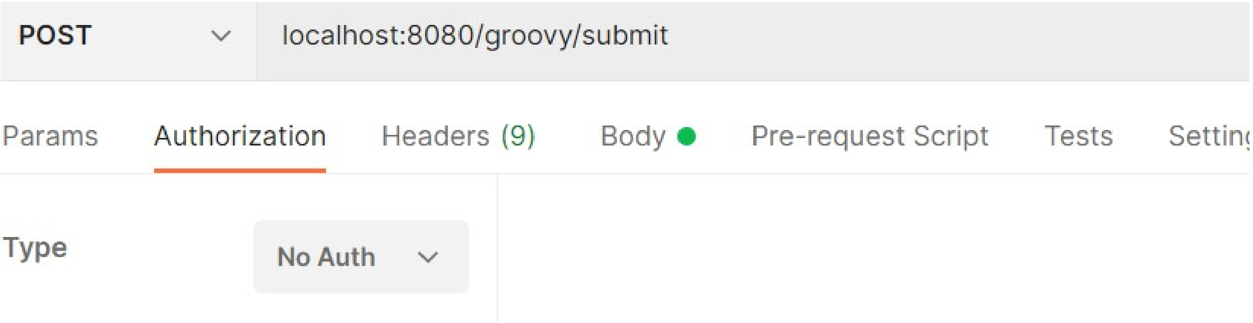
1. Make sure the server is up and running. Use postman or any other API client to post the request.
2. Do a POST operation for the URI localhost:8080/groovy/submit with payload as {"code":"10+3"} and use basic auth user_1/pass_1
3. Now select No Auth in header and do a POST operation for the payload {"code":"10+10"}.

Expected Result:

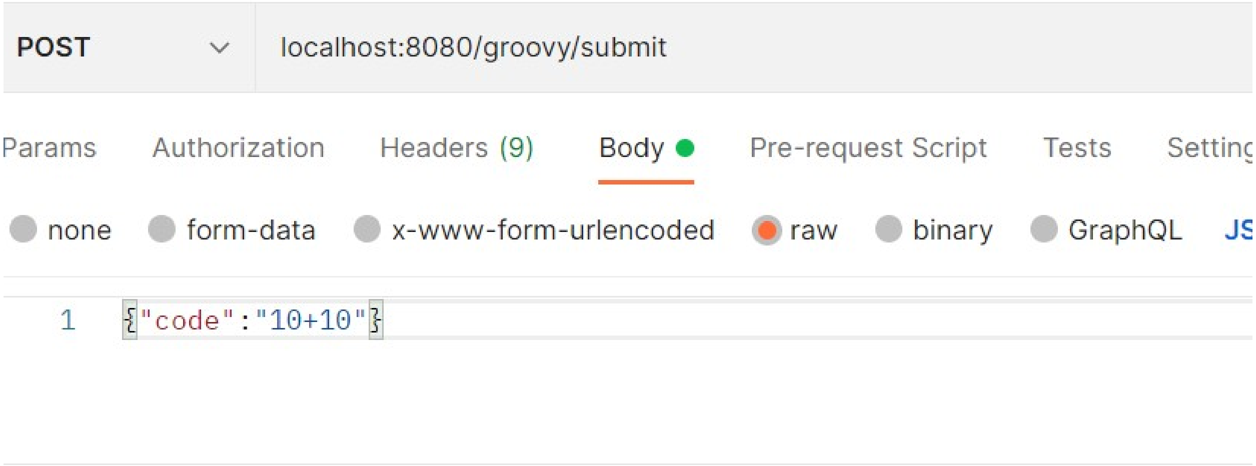
Should be with 401 unauthorized status

Actual Result:

200 OK, ID gets created.


POST localhost:8080/groovy/submit

Type: No Auth


POST localhost:8080/groovy/submit

○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JS

```
{"code":"10+10"}
```

3. Post operation fails when code has groovy class definition with main method and user defined method.

Priority: High

Description:

When the payload has a groovy class definition with main method and a user defined method then the POST operation throws 500, internal server error.

Steps to reproduce:

1. Make sure the server is up and running. Use postman or any other API client to post the request.
2. Do a POST operation for the URI localhost:8080/groovy/submit with payload as

```
{
  "code":"class Example {static void main(String[] args) {Example.test();}static String test(){return 'test';}}"
}
```

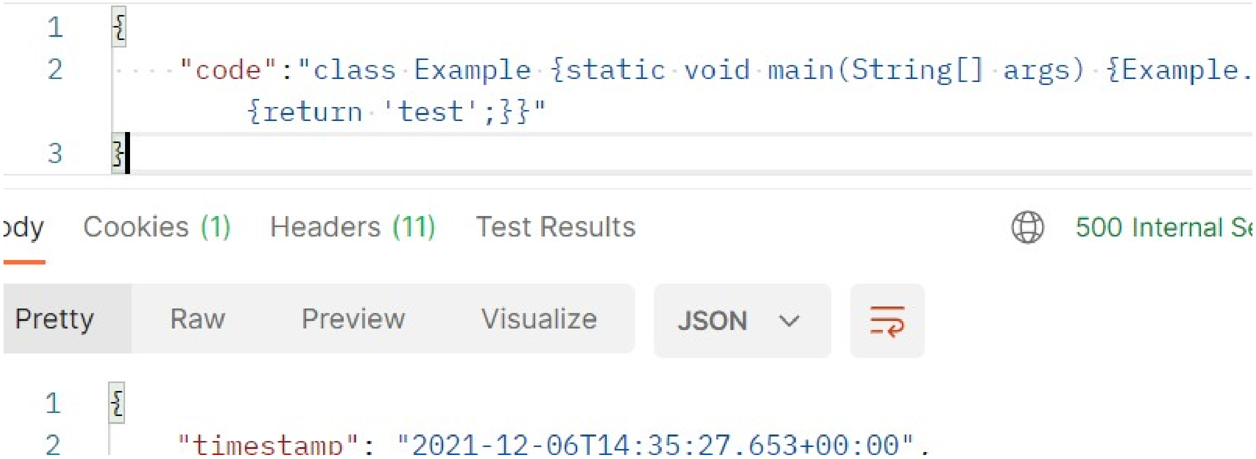
and use basic auth user_1/pass_1

Expected Result:

Should be 200 status

Actual Result:

500, internal server error.



```
{
    "code":"class Example {static void main(String[] args) {Example.
        {return 'test';}}"
}
```

```
{
    "timestamp": "2021-12-06T14:35:27.653+00:00",
```

4. Get operation fails when code has groovy class definition with only a static user defined method.

Priority: High

Description:

When the payload has a groovy class definition with only static method then the POST operation provides 200 status. But the GET operation throws Exception.

Steps to reproduce:

1. Make sure the server is up and running. Use postman or any other API client to post the request.
2. Do a POST operation for the URI localhost:8080/groovy/submit with payload as
   ```
   {
      "code":"class Example {static String test(){return 'test';}}"
   }
   ```
   and use basic auth user_1/pass_1
3. Copy the ID obtained from previous step and do a GET operation using basic auth user_1/pass_1

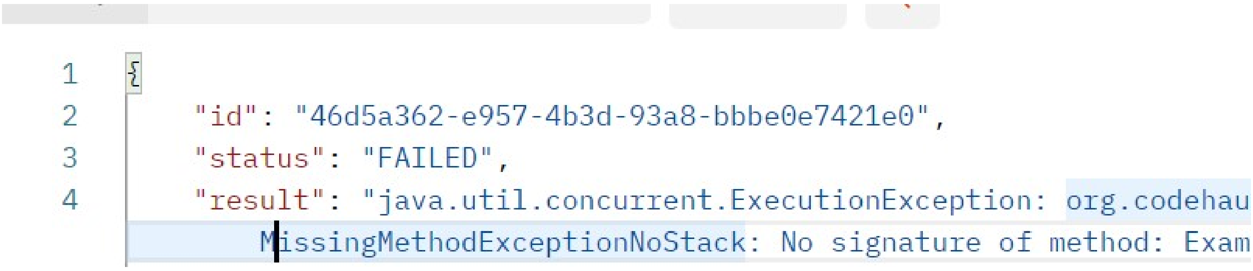
localhost:8080/groovy/status?id=4b8ef2b3-1f03-4401-aa86-cbb0d72a00e4

Expected Result:

Should be 200 status with status as 'COMPLETED' and should have value in result.

Actual Result:

```
"status": "FAILED",
   "result": "java.util.concurrent.ExecutionException: org.codehaus.groovy.runtime.metaclass.MissingMethodExceptionNoStack: No signature of method: Example.main() is applicable for argument types: ([Ljava.lang.String;) values: [[]]\nPossible solutions: wait(), wait(long), find(), any(), wait(long, int), find(groovy.lang.Closure)"
```



```
{
    "id": "46d5a362-e957-4b3d-93a8-bbbe0e7421e0",
    "status": "FAILED",
    "result": "java.util.concurrent.ExecutionException: org.codehau
        MissingMethodExceptionNoStack: No signature of method: Exam
```

5. User operations are not protected. Can be accessed by other users.

Priority: High

Description:

When user_1 does a POST operation and user_2 is trying to do a GET operation with the id obtained from user_1 then user_2 is able to get the details.

Steps to reproduce:

1. Make sure the server is up and running. Use postman or any other API client to post the request.
2. Do a POST operation for the URI localhost:8080/groovy/submit with payload as
   ```
   {
     "code":"200+10"
   }
   ```
   and use basic auth user_1/pass_1
3. Copy the ID obtained from previous step and do a GET operation using basic auth user_2/pass_2
   localhost:8080/groovy/status?id=4b8ef2b3-1f03-4401-aa86-cbb0d72a00e4

Expected Result:

There should be a proper message or the content should not be accessable.

Actual Result:

```
200 OK
"status": "COMPLETED",
  "result": "210"
```

6. When multiple users are using the service endpoints fails with internal server error

Priority: High

Description:

When multiple users are trying to use the API service with different payloads in parallel then the response shows as internal server error. More than two users are not getting queued.

Steps to reproduce:

1. Make sure the server is up and running.
2. Do a POST operation for the URI localhost:8080/groovy/submit with different basic payloads and use different basic auth users user_1/pass_1, user_2/pass_2, user_3/pass_3, user_4/pass_4, user_5/pass_5

Expected Result:

The execution should result in valid response.

Actual Result:

500, internal server error.

7. Execution of multiple endpoints with same user in parallel failed.

Priority: High

Description:

When a user tries to use automation script to execute multiple payloads in parallel then the service fails with internal server error.

Steps to reproduce:

1. Make sure the server is up and running.
2. Use the automation test suite with configuring 5 threads and features run in parallel

Expected Result:

The execution should result in valid response.

Actual Result:

500, internal server error.

8. GET operation for payload with string content gives MissingProperty Exception

Priority: Medium

Description:

When a user tries to do a GET operation for a successful POST operation with string as a payload then MissingProperty Exception occurs.

Steps to reproduce:

1. Make sure the server is up and running. Use basic auth credentials user_1/pass_1
2. Do a POST operation on localhost:8080/groovy/submit with payload as
   ```
   {
     "code":"abc"
   }
   ```
3. Copy the obtained id from the above response and do a GET operation.

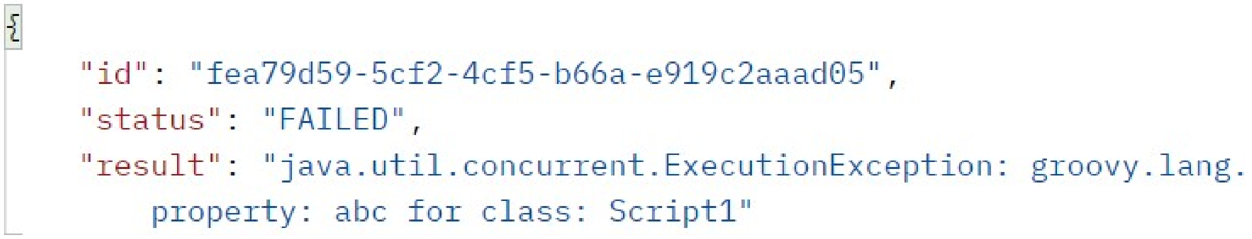
Expected Result:

The execution should result in valid response.

Actual Result:

500, internal server error.
"status": "FAILED",

"result": "java.util.concurrent.ExecutionException: groovy.lang.MissingPropertyException: No such property: abc for class: Script1"



```
{
    "id": "fea79d59-5cf2-4cf5-b66a-e919c2aaad05",
    "status": "FAILED",
    "result": "java.util.concurrent.ExecutionException: groovy.lang.
        property: abc for class: Script1"
```

9. Groovy code with while loop as a payload returns incorrect results

Priority: Low

Description:

When a user tries to test the while loop as a payload then the GET operation results in incorrect values.

Steps to reproduce:

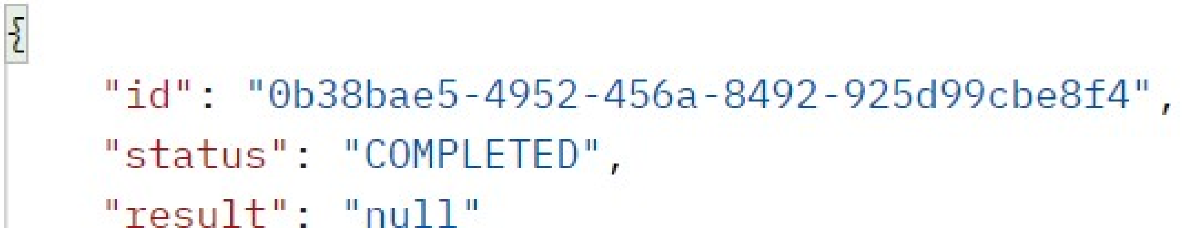
1. Make sure the server is up and running. Use basic auth credentials user_1/pass_1
2. Do a POST operation on localhost:8080/groovy/submit with payload as
   ```
   {
     "code":"int count = 0; while(count<10) {count++;  count;}"
   }
   ```
3. Copy the obtained id from the above response and do a GET operation.
4. Do a POST operation on localhost:8080/groovy/submit with payload as
   ```
   {
     "code":"int count = 0; while(count<10) {count++; return count;}"
   }
   ```
5. Copy the obtained id from the above response and do a GET operation.

Expected Result:

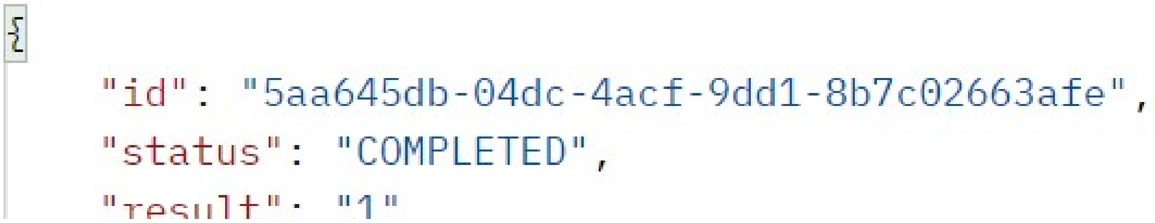
The execution should result in correct values.

Actual Result:

In the first case the result is displayed as ‘null’, where as in the second case the result displayer as ‘1’



```
{
    "id": "0b38bae5-4952-456a-8492-925d99cbe8f4",
    "status": "COMPLETED",
    "result": "null"
```



```
{
    "id": "5aa645db-04dc-4acf-9dd1-8b7c02663afe",
    "status": "COMPLETED",
    "result": "1"
```