Cucumber Report

Start : Feb 10, 9:44:48.136 pm	End : Feb 10, 9:44:49.721 pm	Duration : 1.585 s
Features	Scenarios	Steps
PASSED - 1 FAILED - 0 SKIPPED - 0	PASSED - 3 FAILED - 0 SKIPPED - 0	PASSED - 25 FAILED - 0 SKIPPED - 0
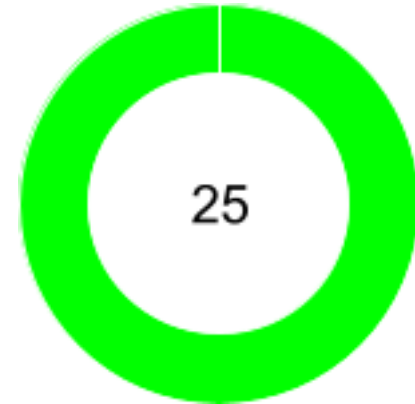
Feature		Scenario				Step			
Name	**Duration**	**T**	**P**	**F**	**S**	**T**	**P**	**F**	**S**
This feature will test/verify basic functionalities of REST API [GET, POST, PUT & DELETE] using RESTAssured library.	*1.585 s*	*3*	*3*	*0*	*0*	*25*	*25*	*0*	*0*

TAG	Scenario				Feature			
Name	T	P	F	S	T	P	F	S
@get-method	2	2	0	0	1	1	0	0
@api-test1	2	2	0	0	1	1	0	0
@api-test2	1	1	0	0	1	1	0	0
@post-method	1	1	0	0	1	1	0	0

#	Feature Name	T	P	F	S	Duration
1	*This feature will test/verify basic functionalities of REST API [GET, POST, PUT & DELETE] using RESTAssured library.*	3	3	0	0	1.585 s
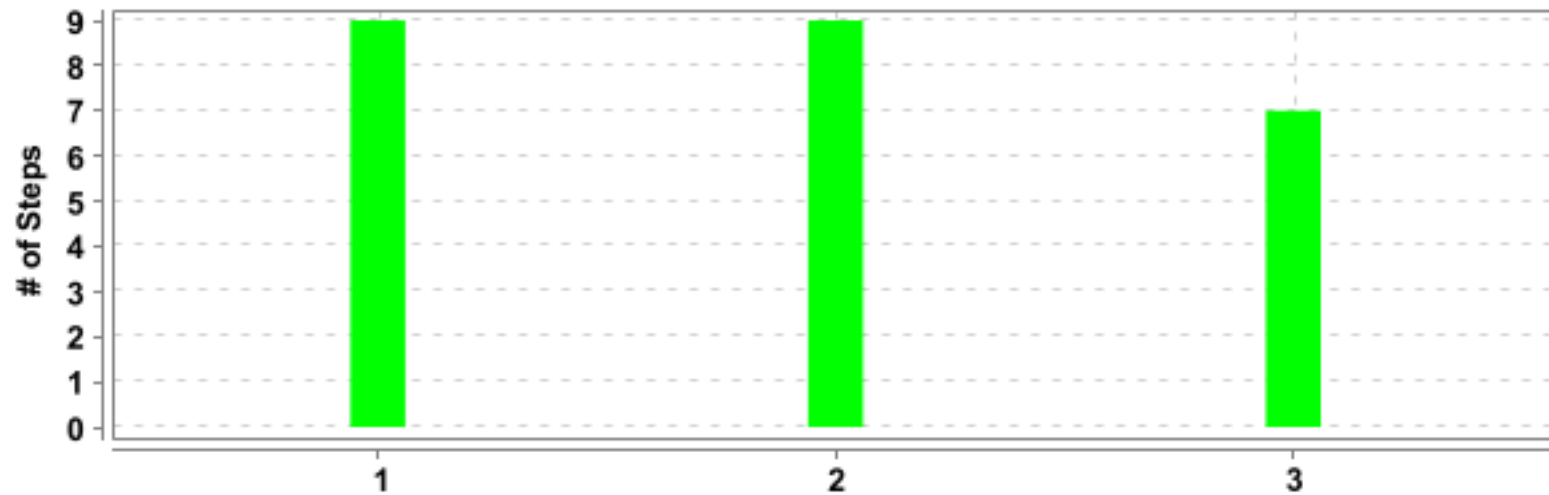
#	Feature Name	Scenario Name	T	P	F	S	Duration
1	*This feature will test/verify basic functionalities of REST API [GET, **	*GET Method Test - verify active users & stocks data using GET method*	9	9	0	0	1.396 s
2		*GET Method Test - verify active users & stocks data using GET method*	9	9	0	0	0.063 s
3		*POST Method Test - create new user using POST method*	7	7	0	0	0.067 s

This feature will test/verify basic functionalities of REST API [GET, POST, PUT & DELETE] using RESTAssured library.

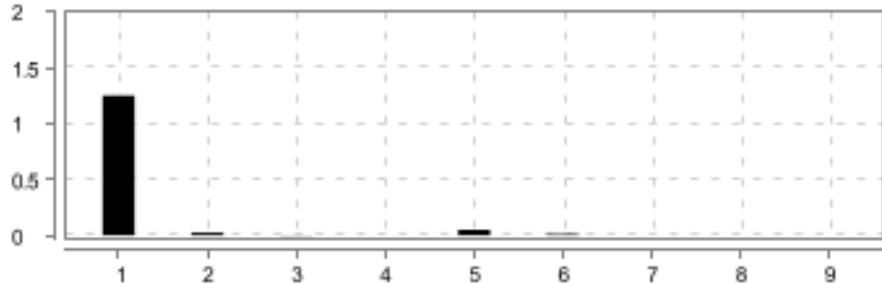| PASSED	DURATION - 1.585 s				
/ 9:44:48.136 pm // 9:44:49.721 pm /					

Scenarios
Total - 3
Pass - 3
Fail - 0
Skip - 0

3

Steps
Total - 25
Pass - 25
Fail - 0
Skip - 0

25

GET Method Test - verify active users & stocks data using GET method

PASSED	DURATION - 1.396 s
/ 9:44:48.143 pm // 9:44:49.539 pm /	
This feature will test/verify basic functionalities of REST API [GET, POST, PUT & DELETE] using RESTAssu *	
@get-method @api-test1	

Steps
Total - 9
Pass - 9
Fail - 0
Skip - 0

9

#	Step / Hook Details	Status	Duration
1	Given API server is up & running on "http://localhost:3000"	PASSED	1.252 s
2	When user access the endpoint "/users"	PASSED	0.024 s
3	Then user should get status as "200"	PASSED	0.003 s
4	And user should receive data with header content-type as "application/json"	PASSED	0.001 s
5	And user should get active user "Oneil Pinto"	PASSED	0.051 s
6	When user access the endpoint "/stocks"	PASSED	0.013 s
7	Then user should get status as "200"	PASSED	0.001 s
8	And user should receive data with header content-type as "application/json"	PASSED	0.001 s
9	Then user should get stock as "TCS"	PASSED	0.001 s

GET Method Test - verify active users & stocks data using GET method

PASSED	DURATION - 0.063 s

/ 9:44:49.573 pm // 9:44:49.636 pm /

*This feature will test/verify basic functionalities of REST API [GET, POST, PUT & DELETE] using RESTAssu ***

@get-method @api-test1

Steps

Total - 9

Pass - 9

Fail - 0

Skip - 0

9

#	Step / Hook Details	Status	Duration
1	Given API server is up & running on "http://localhost:3000"	PASSED	0.014 s
2	When user access the endpoint "/users"	PASSED	0.012 s
3	Then user should get status as "200"	PASSED	0.001 s
4	And user should receive data with header content-type as "application/json"	PASSED	0.001 s
5	And user should get active user "Amol Jadhav"	PASSED	0.004 s
6	When user access the endpoint "/stocks"	PASSED	0.012 s
7	Then user should get status as "200"	PASSED	0.000 s
8	And user should receive data with header content-type as "application/json"	PASSED	0.002 s
9	Then user should get stock as "HDFC"	PASSED	0.001 s

POST Method Test - create new user using POST method

PASSED	DURATION - 0.067 s

/ 9:44:49.654 pm // 9:44:49.721 pm /

*This feature will test/verify basic functionalities of REST API [GET, POST, PUT & DELETE] using RESTAssu ***

@api-test2 @post-method

Steps

Total - 7

Pass - 7

Fail - 0

Skip - 0

7

#	Step / Hook Details	Status	Duration
1	Given API server is up & running on "http://localhost:3000"	PASSED	0.017 s
2	When user access the endpoint "/users"	PASSED	0.012 s
3	Then user should get status as "200"	PASSED	0.000 s
4	And user should receive data with header content-type as "application/json"	PASSED	0.000 s

#	Step / Hook Details	Status	Duration
5	When user create and reads the payload "payload_new_user.txt"	PASSED	0.001 s
6	And attach it to the request body and post	PASSED	0.028 s
7	Then user should get status as "201"	PASSED	0.000 s