

Cucumber Report

Jan 26, 2022, 11:16:33 AM

Start : Jan 26, 11:15:16.076 AM

End : Jan 26, 11:16:31.124 AM

Duration : 1 m 15.048 s

Features

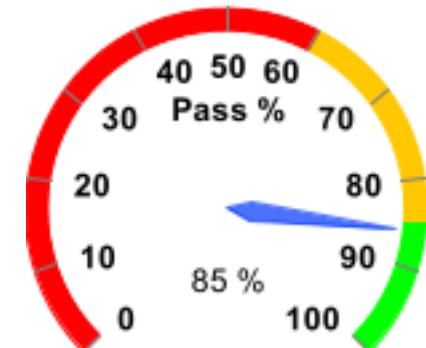
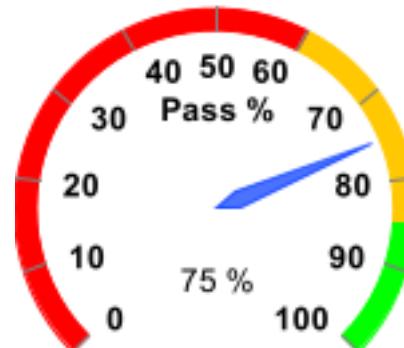
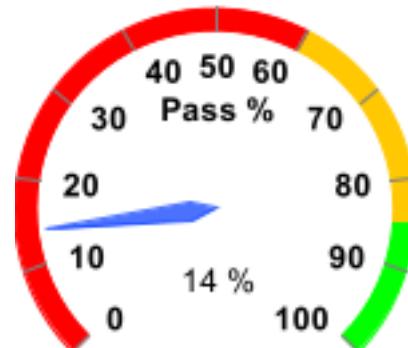
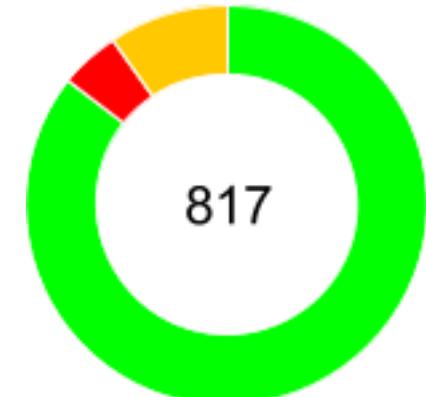
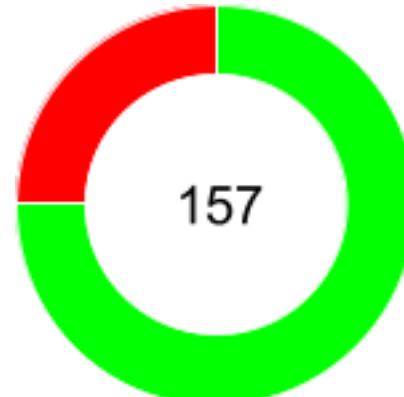
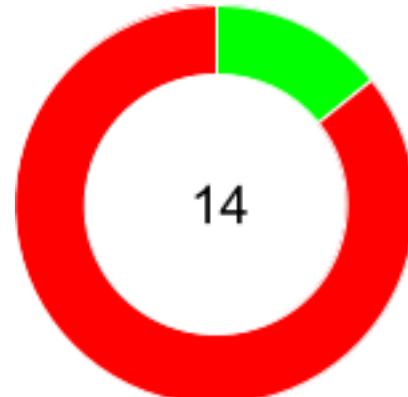
PASSED - 2
FAILED - 12
SKIPPED - 0

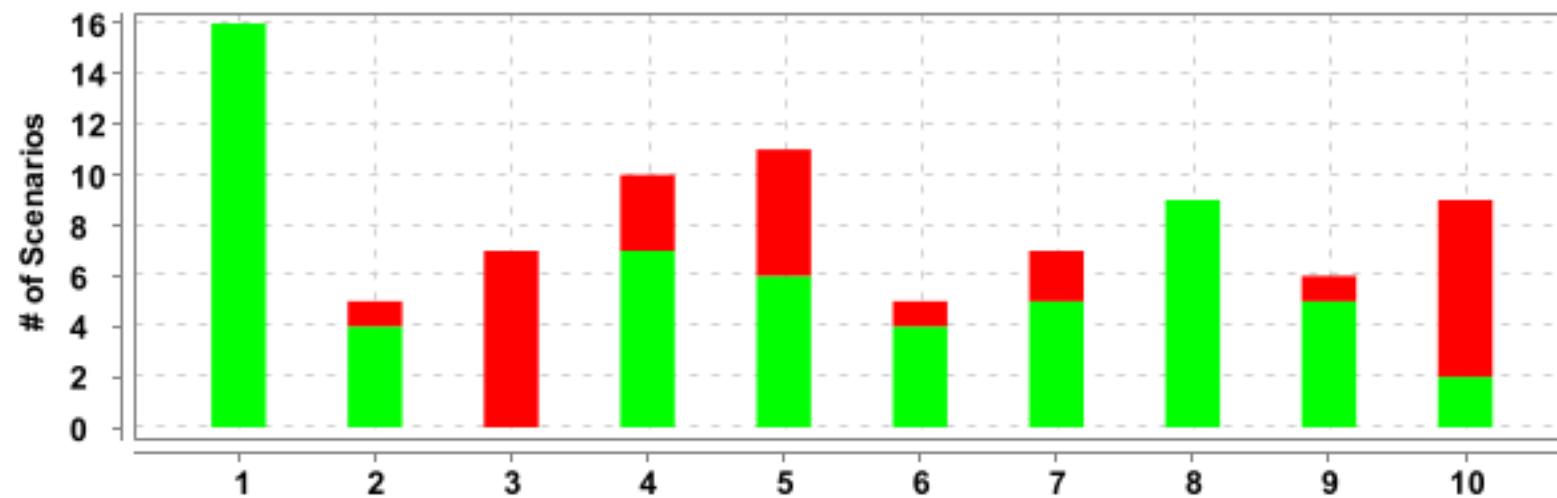
Scenarios

PASSED - 118
FAILED - 39
SKIPPED - 0

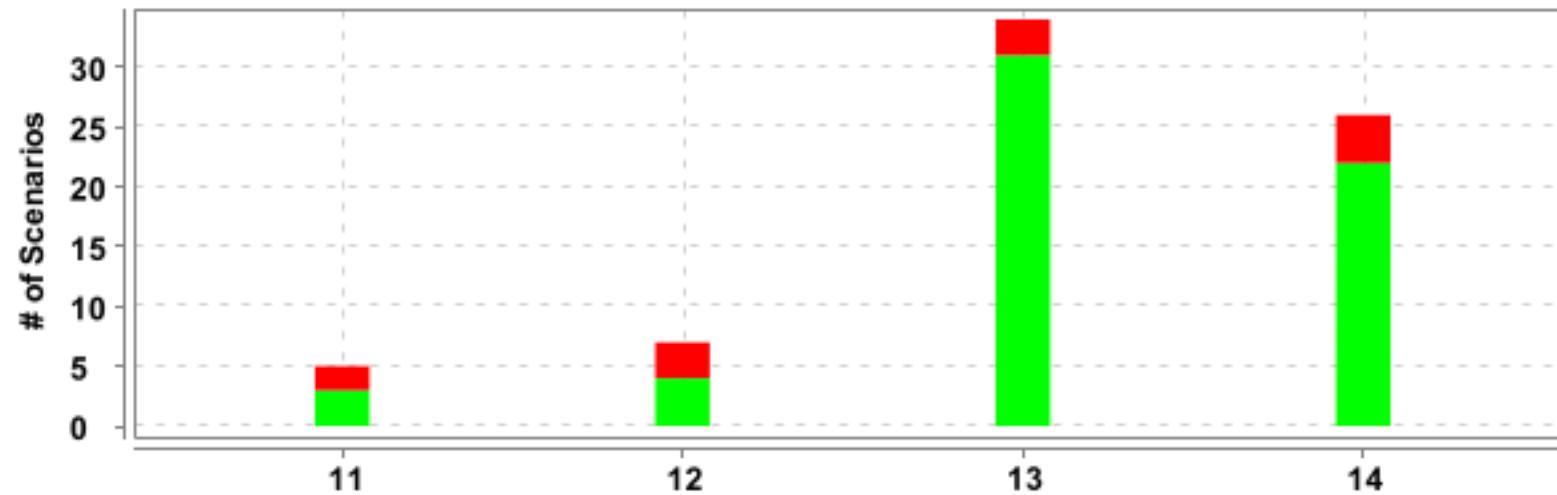
Steps

PASSED - 699
FAILED - 39
SKIPPED - 79

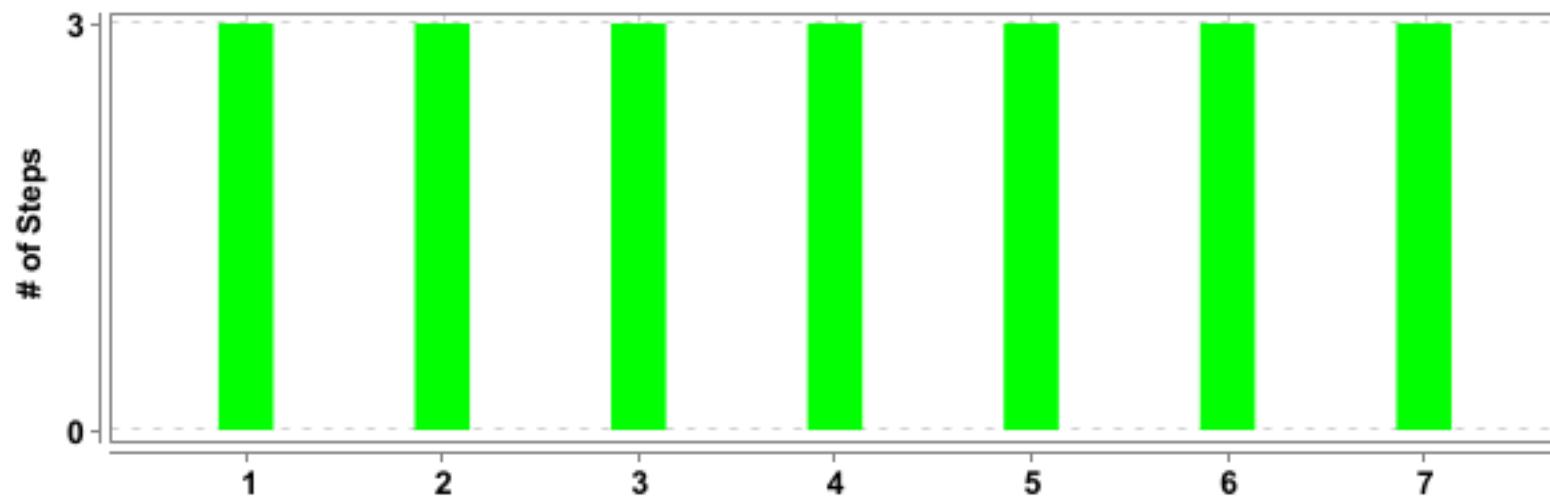




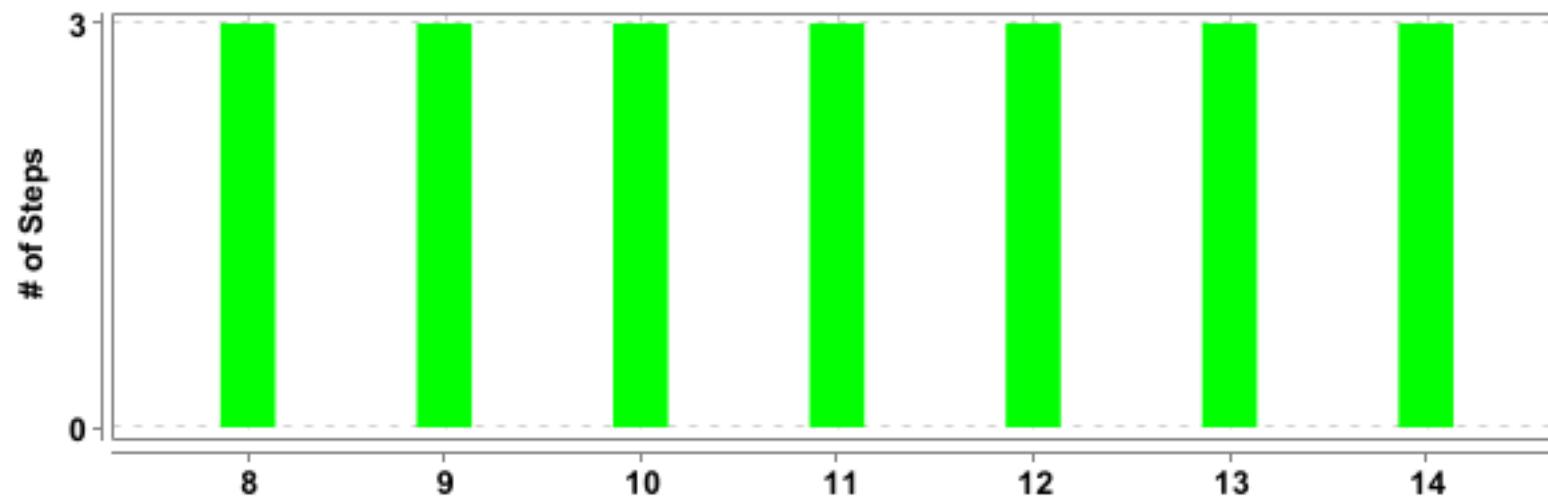
#	Feature Name	T	P	F	S	Duration
1	<u>Check the Authorization for LMS API</u>	16	16	0	0	35.280 s
2	<u>To check The DELETE method for Skills</u>	5	4	1	0	2.217 s
3	<u>To check The GET method for Skills</u>	7	0	7	0	1.307 s
4	<u>To check The POST method for Skills</u>	10	7	3	0	4.919 s
5	<u>To check The PUT method for Skills</u>	11	6	5	0	3.418 s
6	<u>Delete user skills details</u>	5	4	1	0	2.566 s
7	<u>To check The GET method for UserSkills</u>	7	5	2	0	2.455 s
8	<u>To check The POST method for UserSkills</u>	9	9	0	0	4.694 s
9	<u>To check The PUT method for UserSkills</u>	6	5	1	0	3.656 s
10	<u>To check The GET method for UserSkillsMap API</u>	9	2	7	0	1.364 s



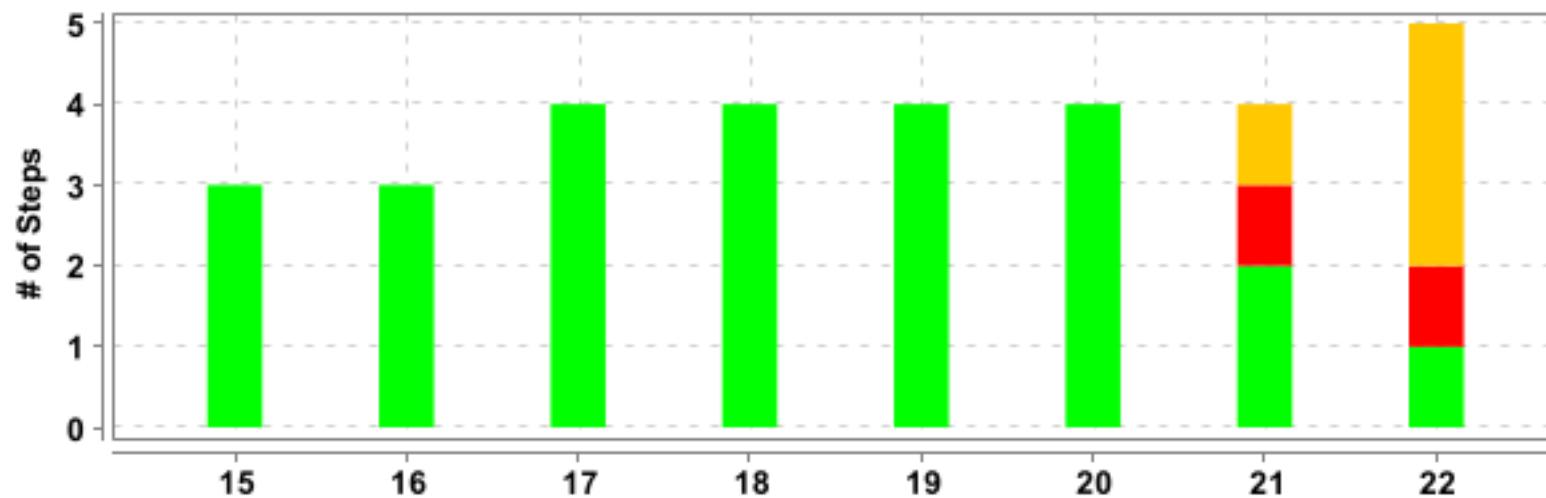
#	Feature Name	T	P	F	S	Duration
11	<u>Delete Users details</u>	5	3	2	0	1.017 s
12	<u>To check The GET method for Users</u>	7	4	3	0	3.156 s
13	<u>To check The POST method for Users</u>	34	31	3	0	19.973 s
14	<u>To check The PUT method for Users</u>	26	22	4	0	15.493 s



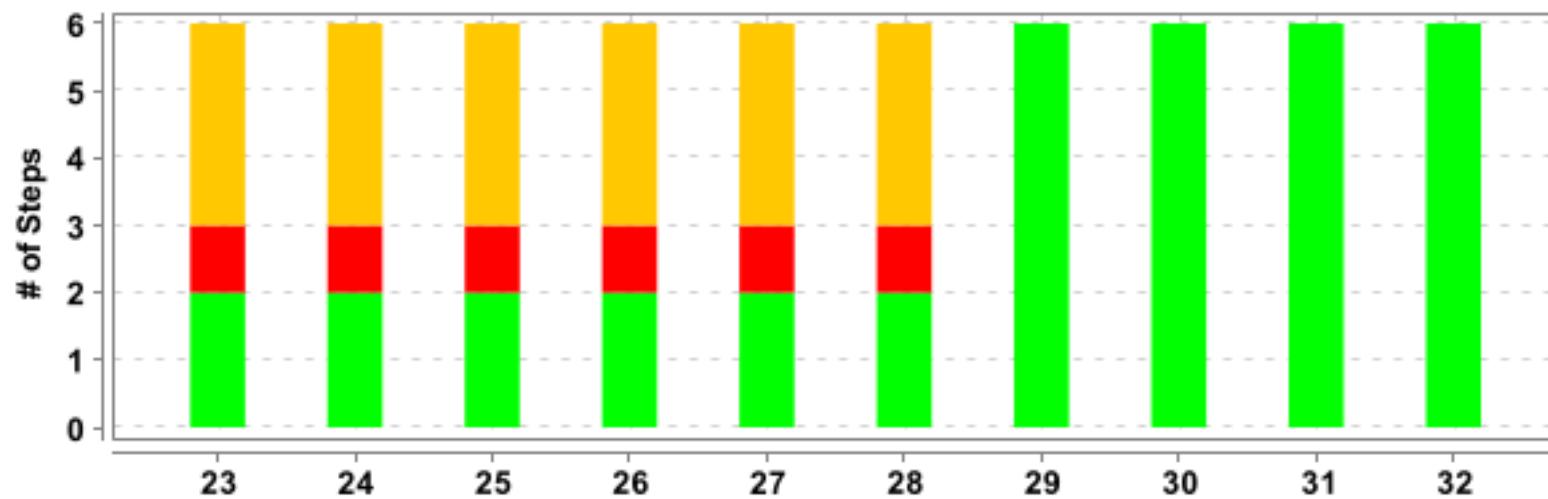
#	Feature Name	Scenario Name	T	P	F	S	Duration
1	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	4.083 s
2	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.315 s
3	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.334 s
4	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.813 s
5	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.204 s
6	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.144 s
7	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.156 s



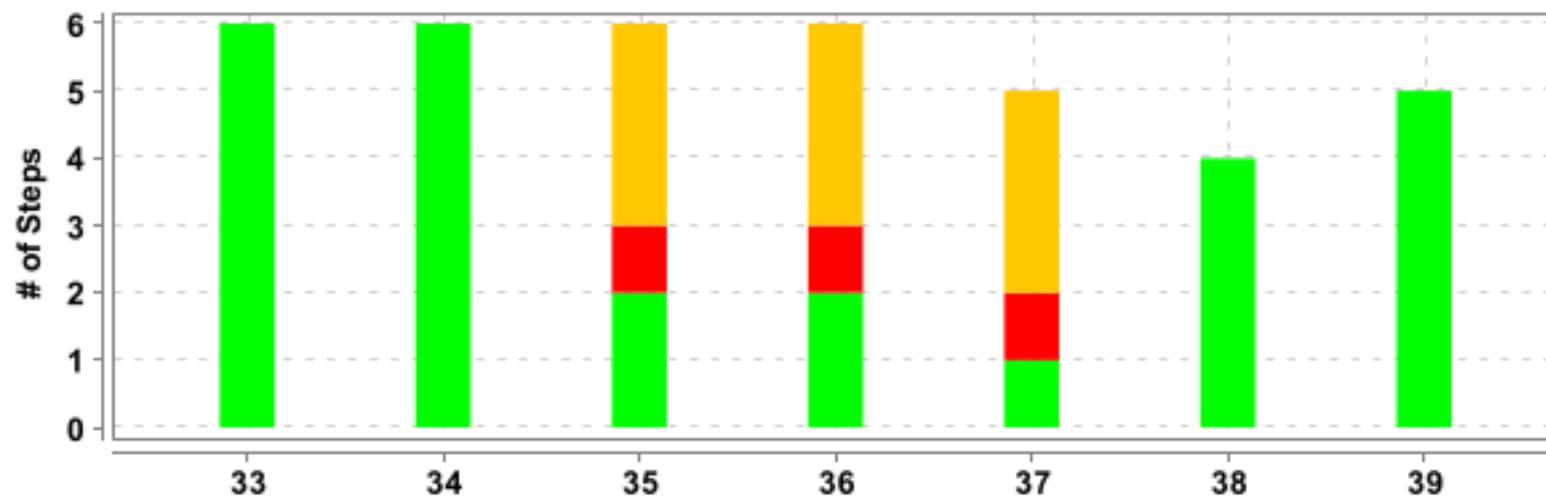
#	Feature Name	Scenario Name	T	P	F	S	Duration
8	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.163 s
9	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.089 s
10	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.110 s
11	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.103 s
12	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.148 s
13	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.164 s
14	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.196 s



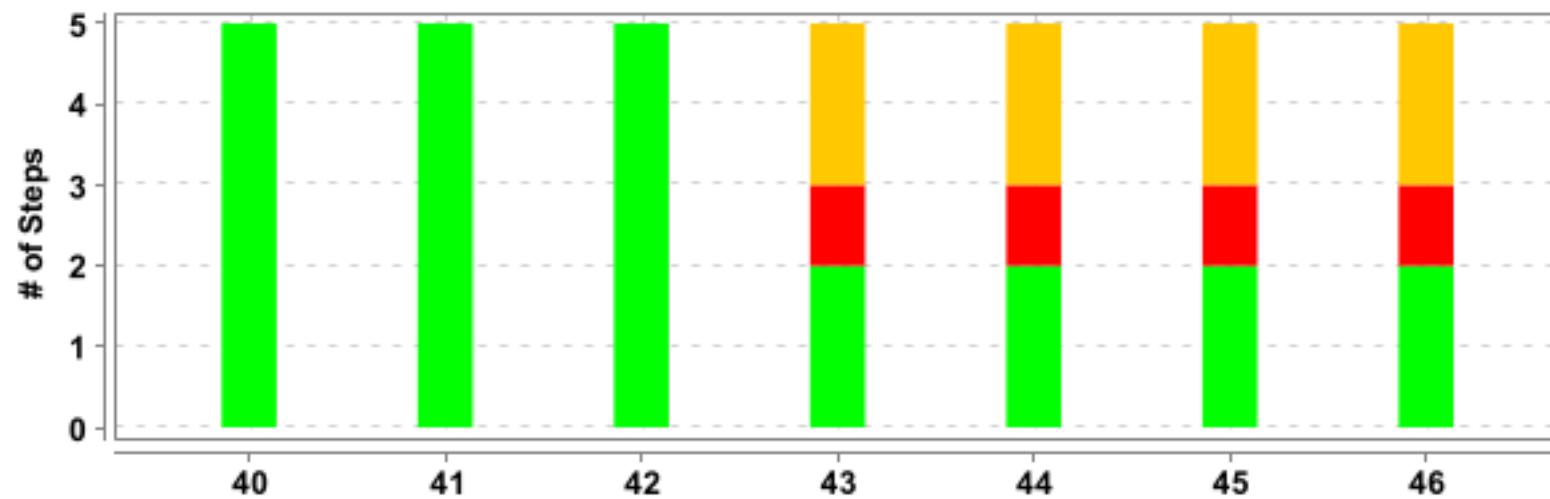
#	Feature Name	Scenario Name	T	P	F	S	Duration
15	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.246 s
16	<u>Check the Authorization for LMS API</u>	<u>Check the Authorization with with different set of username & password</u>	3	3	0	0	0.295 s
17	<u>To check The DELETE method for Skills</u>	<u>Validate DELETE Skills when user logged with UserName with Password</u>	4	4	0	0	0.638 s
18	<u>To check The DELETE method for Skills</u>	<u>Validate DELETE Skills when user logged with UserName with Password</u>	4	4	0	0	0.334 s
19	<u>To check The DELETE method for Skills</u>	<u>Validate DELETE Skills when user logged with UserName with Password</u>	4	4	0	0	0.413 s
20	<u>To check The DELETE method for Skills</u>	<u>Validate DELETE Skills when user logged with UserName with Password</u>	4	4	0	0	0.378 s
21	<u>To check The DELETE method for Skills</u>	<u>Validate DELETE Skills when user logged with UserName with Password</u>	4	2	1	1	0.225 s
22	<u>To check The GET method for Skills</u>	<u>Verify GET all Skills for Skill Master</u>	5	1	1	3	0.029 s



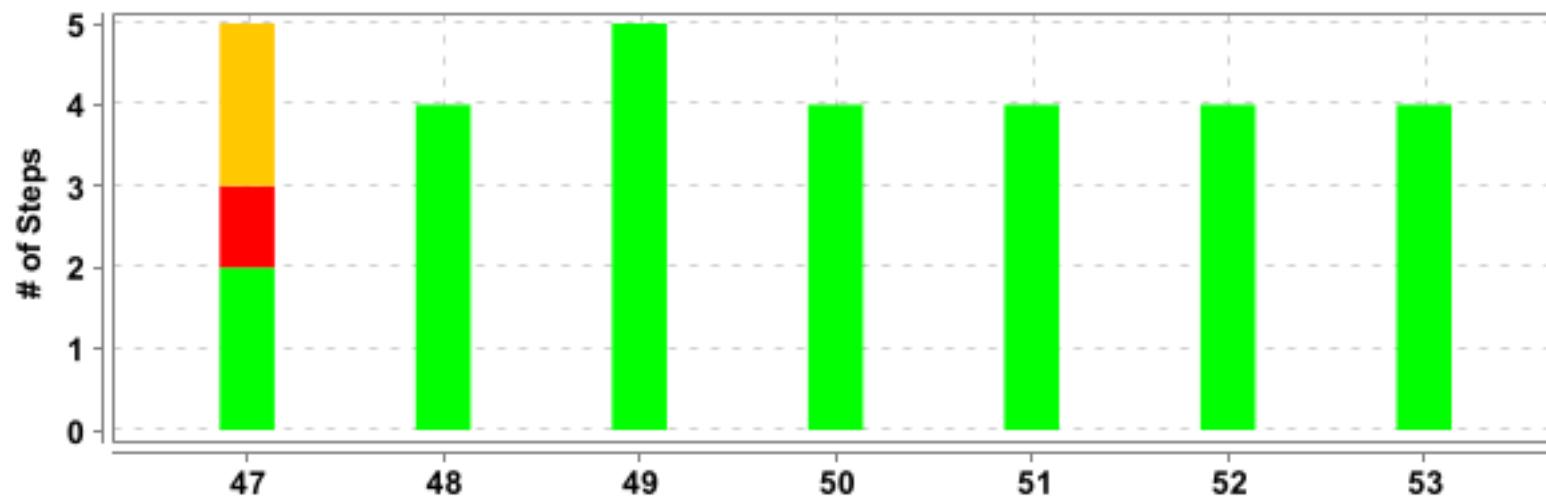
#	Feature Name	Scenario Name	T	P	F	S	Duration
23	<u>To check The GET method for Skills</u>	<u>Validate Skills Api with specific ID</u>	6	2	1	3	0.119 s
24	<u>To check The GET method for Skills</u>	<u>Validate Skills Api with specific ID</u>	6	2	1	3	0.123 s
25	<u>To check The GET method for Skills</u>	<u>Validate Skills Api with specific ID</u>	6	2	1	3	0.157 s
26	<u>To check The GET method for Skills</u>	<u>Validate Skills Api with specific ID</u>	6	2	1	3	0.154 s
27	<u>To check The GET method for Skills</u>	<u>Validate Skills Api with specific ID</u>	6	2	1	3	0.182 s
28	<u>To check The GET method for Skills</u>	<u>Validate Skills Api with specific ID</u>	6	2	1	3	0.160 s
29	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	6	6	0	0	1.905 s
30	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	6	6	0	0	0.300 s
31	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	6	6	0	0	0.463 s
32	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	6	6	0	0	0.442 s



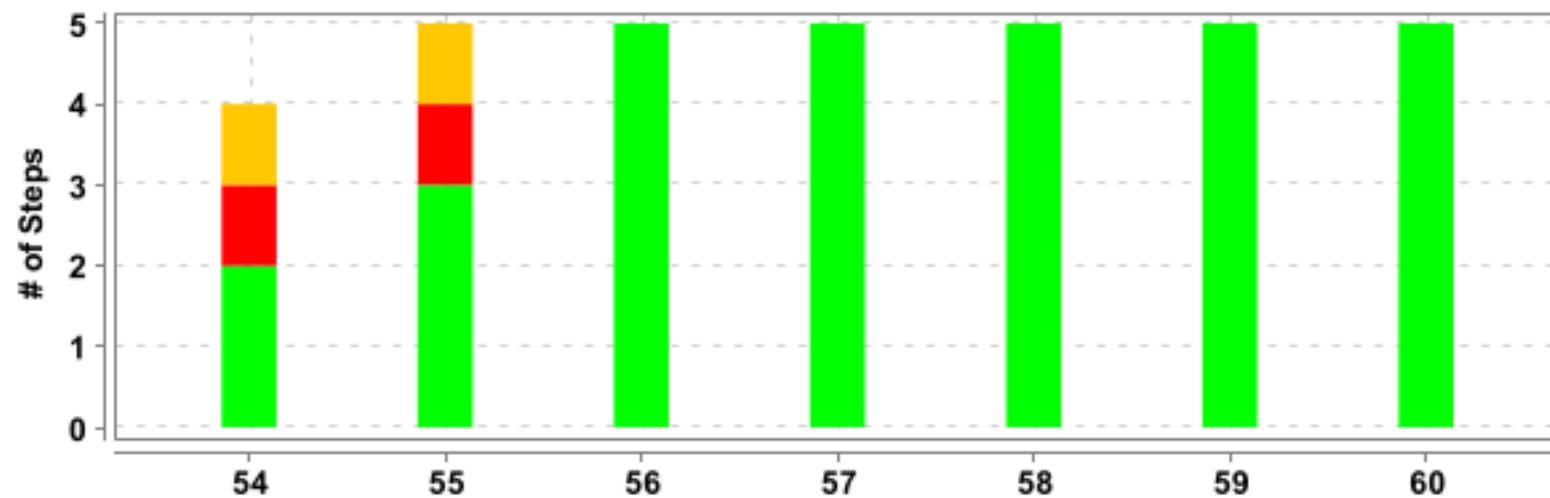
#	Feature Name	Scenario Name	T	P	F	S	Duration
33	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	6	6	0	0	0.346 s
34	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	6	6	0	0	0.284 s
35	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	6	2	1	3	0.200 s
36	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	6	2	1	3	0.187 s
37	<u>To check The POST method for Skills</u>	<u>Verify POST for a Skill when user logged with UserName with Password</u>	5	1	1	3	0.180 s
38	<u>To check The POST method for Skills</u>	<u>Verify if POST fails for improper format when user logged with UserName with Password</u>	4	4	0	0	0.289 s
39	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	5	0	0	0.818 s



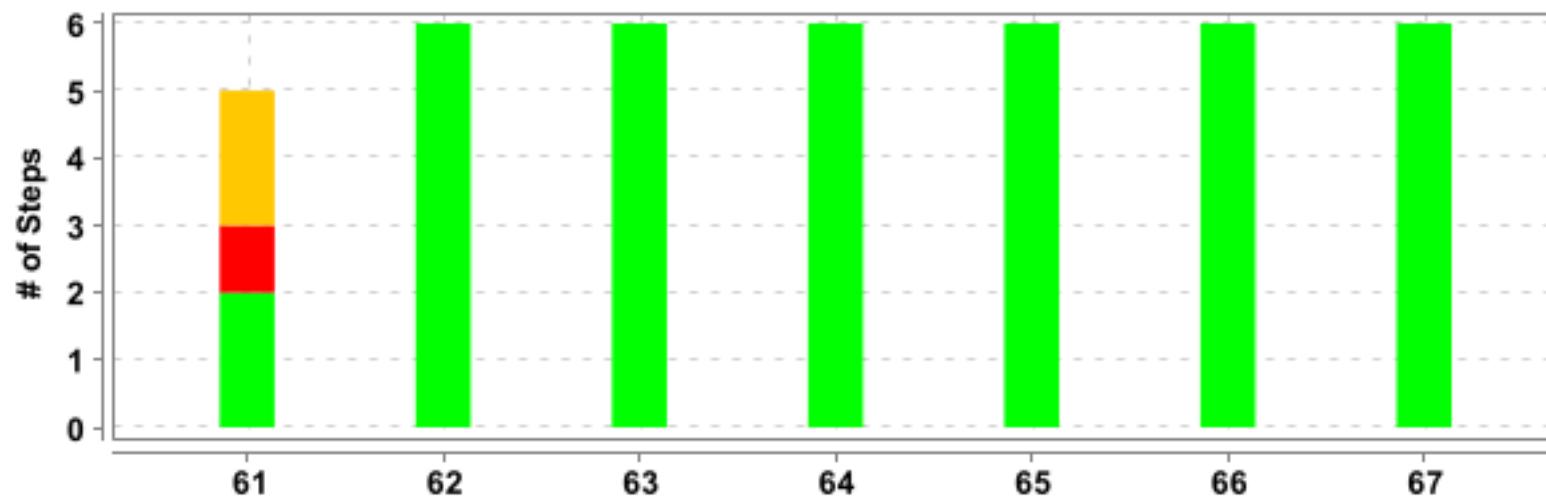
#	Feature Name	Scenario Name	T	P	F	S	Duration
40	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	5	0	0	0.265 s
41	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	5	0	0	0.245 s
42	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	5	0	0	0.259 s
43	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	2	1	2	0.237 s
44	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	2	1	2	0.239 s
45	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	2	1	2	0.238 s
46	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	2	1	2	0.216 s



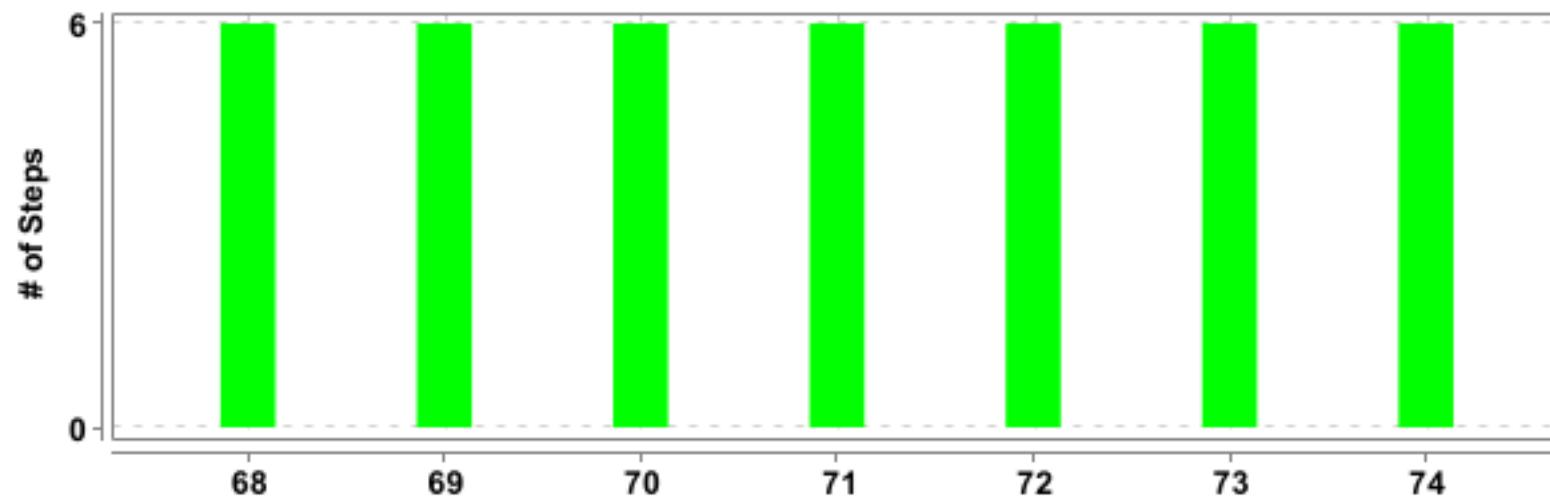
#	Feature Name	Scenario Name	T	P	F	S	Duration
47	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	5	2	1	2	0.188 s
48	<u>To check The PUT method for Skills</u>	<u>Verify PUT for a Skill when user logged with UserName with Password</u>	4	4	0	0	0.180 s
49	<u>To check The PUT method for Skills</u>	<u>Verify PUT fails for a Skill with improper format when user logged with UserName with Password</u>	5	5	0	0	0.233 s
50	<u>Delete user skills details</u>	<u>Validate the delete method for UserSkills</u>	4	4	0	0	0.543 s
51	<u>Delete user skills details</u>	<u>Validate the delete method for UserSkills</u>	4	4	0	0	0.784 s
52	<u>Delete user skills details</u>	<u>Validate the delete method for UserSkills</u>	4	4	0	0	0.535 s
53	<u>Delete user skills details</u>	<u>Validate the delete method for UserSkills</u>	4	4	0	0	0.473 s



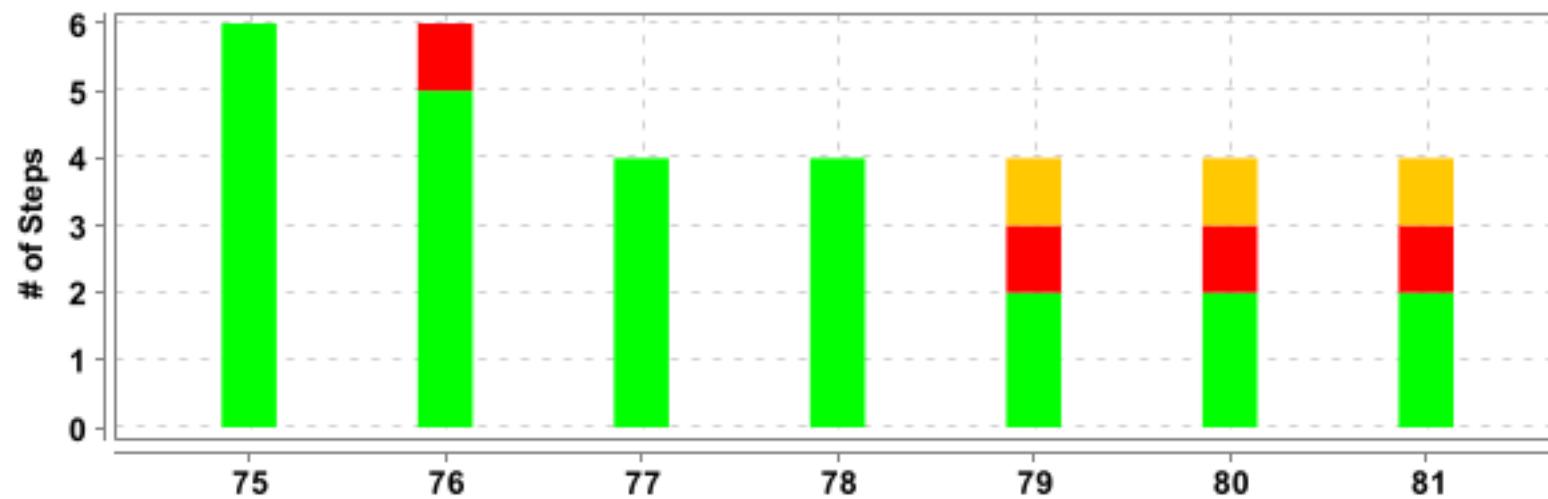
#	Feature Name	Scenario Name	T	P	F	S	Duration
54	<u>Delete user skills details</u>	<u>Validate the delete method for UserSkills</u>	4	2	1	1	0.117 s
55	<u>To check The GET method for UserSkills</u>	<u>Validate GET all UserSkills</u>	5	3	1	1	0.070 s
56	<u>To check The GET method for UserSkills</u>	<u>Validate UserSkills Api with specific ID</u>	5	5	0	0	0.448 s
57	<u>To check The GET method for UserSkills</u>	<u>Validate UserSkills Api with specific ID</u>	5	5	0	0	0.406 s
58	<u>To check The GET method for UserSkills</u>	<u>Validate UserSkills Api with specific ID</u>	5	5	0	0	0.446 s
59	<u>To check The GET method for UserSkills</u>	<u>Validate UserSkills Api with specific ID</u>	5	5	0	0	0.439 s
60	<u>To check The GET method for UserSkills</u>	<u>Validate UserSkills Api with specific ID</u>	5	5	0	0	0.407 s



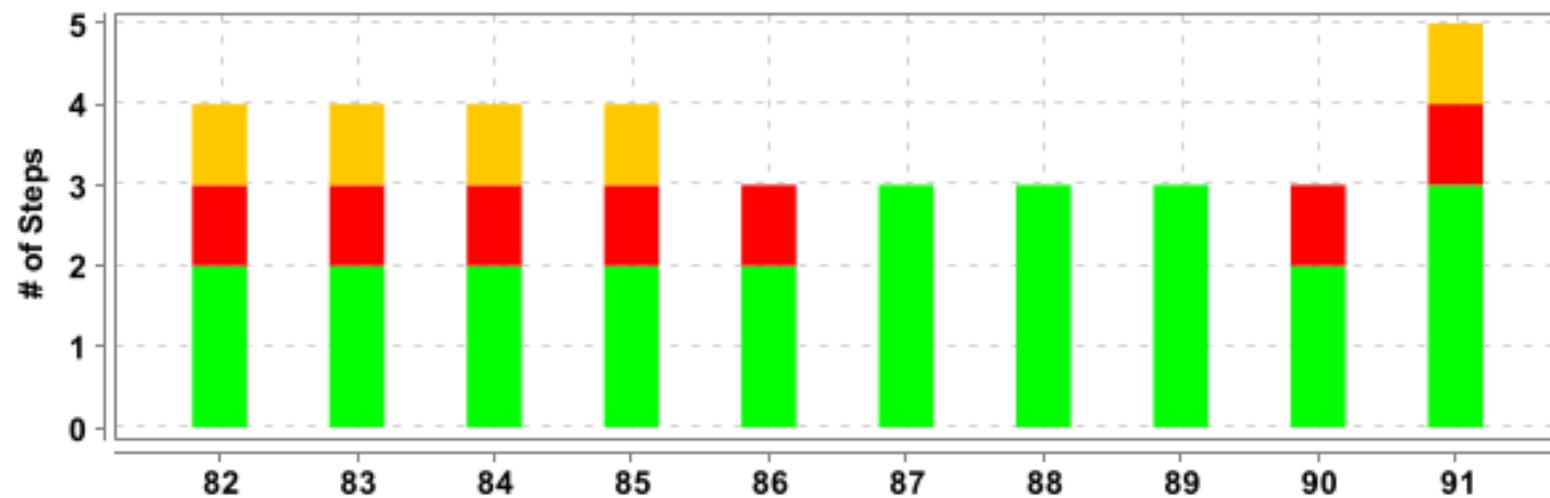
#	Feature Name	Scenario Name	T	P	F	S	Duration
61	<u>To check The GET method for UserSkills</u>	Validate UserSkills Api with specific ID	5	2	1	2	0.086 s
62	<u>To check The POST method for UserSkills</u>	Validate POST all UserSkills	6	6	0	0	0.647 s
63	<u>To check The POST method for UserSkills</u>	Validate POST all UserSkills	6	6	0	0	0.471 s
64	<u>To check The POST method for UserSkills</u>	Validate POST all UserSkills	6	6	0	0	0.444 s
65	<u>To check The POST method for UserSkills</u>	Validate POST all UserSkills	6	6	0	0	0.559 s
66	<u>To check The POST method for UserSkills</u>	Validate POST all UserSkills	6	6	0	0	0.452 s
67	<u>To check The POST method for UserSkills</u>	Validate POST all UserSkills	6	6	0	0	0.465 s



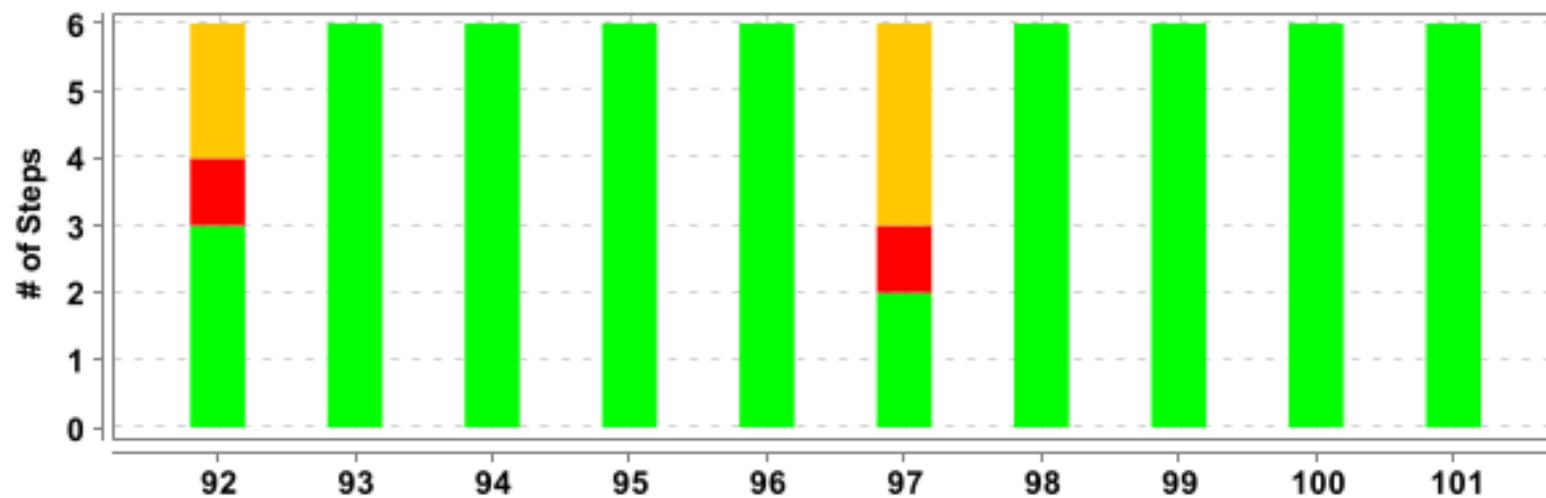
#	Feature Name	Scenario Name	T	P	F	S	Duration
68	<u>To check The POST method for UserSkills</u>	<u>Validate POST all UserSkills</u>	6	6	0	0	0.535 s
69	<u>To check The POST method for UserSkills</u>	<u>Validate POST all UserSkills</u>	6	6	0	0	0.460 s
70	<u>To check The POST method for UserSkills</u>	<u>Validate POST all UserSkills</u>	6	6	0	0	0.475 s
71	<u>To check The PUT method for UserSkills</u>	<u>Validate PUT method for UserSkills</u>	6	6	0	0	0.584 s
72	<u>To check The PUT method for UserSkills</u>	<u>Validate PUT method for UserSkills</u>	6	6	0	0	0.521 s
73	<u>To check The PUT method for UserSkills</u>	<u>Validate PUT method for UserSkills</u>	6	6	0	0	0.472 s
74	<u>To check The PUT method for UserSkills</u>	<u>Validate PUT method for UserSkills</u>	6	6	0	0	0.528 s



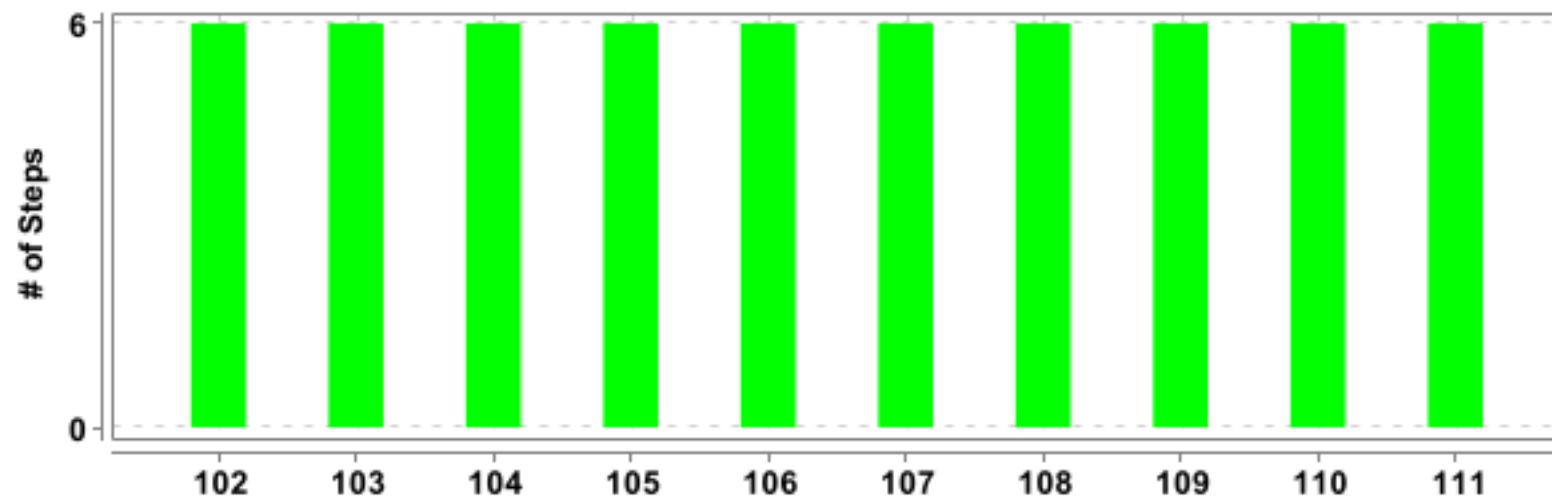
#	Feature Name	Scenario Name	T	P	F	S	Duration
75	<u>To check The PUT method for UserSkills</u>	<u>Validate PUT method for UserSkills</u>	6	6	0	0	0.495 s
76	<u>To check The PUT method for UserSkills</u>	<u>Validate creation time & modification time for PUT method for UserSkills</u>	6	5	1	0	0.953 s
77	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET all User-to-skill mappings</u>	4	4	0	0	0.171 s
78	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET one particular User-to-skill mappings with specific user ID</u>	4	4	0	0	0.175 s
79	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET one particular User-to-skill mappings with specific user ID</u>	4	2	1	1	0.128 s
80	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET one particular User-to-skill mappings with specific user ID</u>	4	2	1	1	0.095 s
81	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET one particular User-to-skill mappings with specific user ID</u>	4	2	1	1	0.115 s



#	Feature Name	Scenario Name	T	P	F	S	Duration
82	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET one particular Skill-to-users mappings with specific skill ID</u>	4	2	1	1	0.113 s
83	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET one particular Skill-to-users mappings with specific skill ID</u>	4	2	1	1	0.088 s
84	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET one particular Skill-to-users mappings with specific skill ID</u>	4	2	1	1	0.104 s
85	<u>To check The GET method for UserSkillsMap API</u>	<u>Validate GET one particular Skill-to-users mappings with specific skill ID</u>	4	2	1	1	0.104 s
86	<u>Delete Users details</u>	<u>Validate the delete method for Users</u>	3	2	1	0	0.226 s
87	<u>Delete Users details</u>	<u>Validate the delete method for Users</u>	3	3	0	0	0.189 s
88	<u>Delete Users details</u>	<u>Validate the delete method for Users</u>	3	3	0	0	0.165 s
89	<u>Delete Users details</u>	<u>Validate the delete method for Users</u>	3	3	0	0	0.177 s
90	<u>Delete Users details</u>	<u>Validate the delete method for Users</u>	3	2	1	0	0.168 s
91	<u>To check The GET method for Users</u>	<u>Validate GET all Users</u>	5	3	1	1	0.033 s

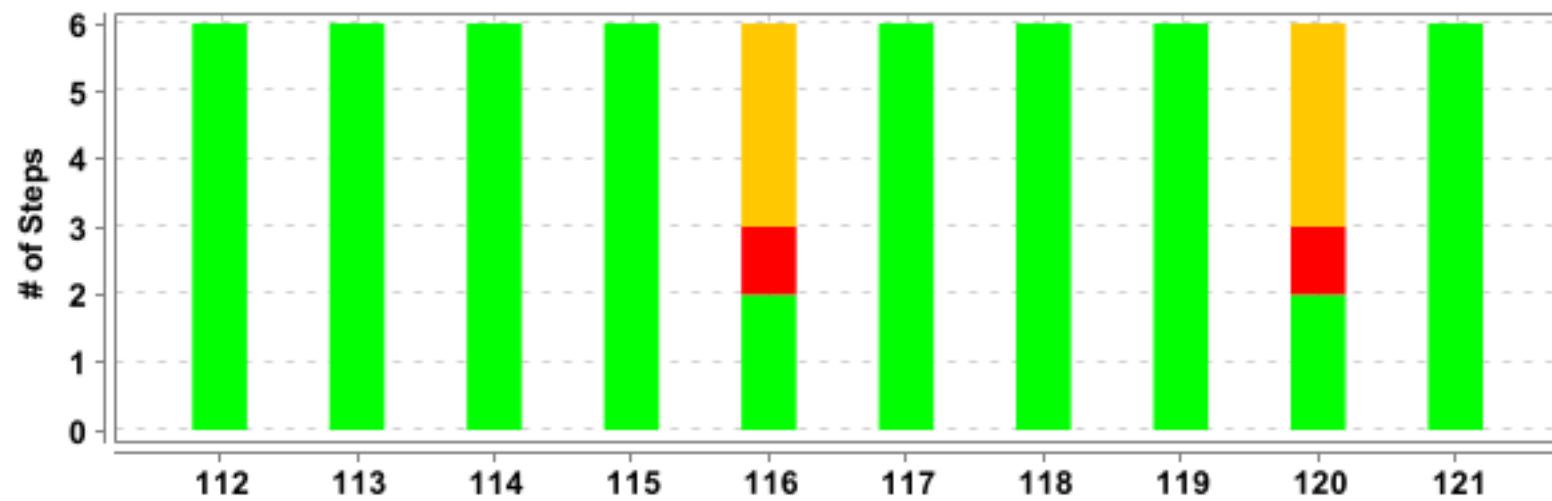


#	Feature Name	Scenario Name	T	P	F	S	Duration
92	<u>To check The GET method for Users</u>	<u>Validate Users Api with specific ID</u>	6	3	1	2	0.310 s
93	<u>To check The GET method for Users</u>	<u>Validate Users Api with specific ID</u>	6	6	0	0	0.751 s
94	<u>To check The GET method for Users</u>	<u>Validate Users Api with specific ID</u>	6	6	0	0	0.545 s
95	<u>To check The GET method for Users</u>	<u>Validate Users Api with specific ID</u>	6	6	0	0	0.633 s
96	<u>To check The GET method for Users</u>	<u>Validate Users Api with specific ID</u>	6	6	0	0	0.540 s
97	<u>To check The GET method for Users</u>	<u>Validate Users Api with specific ID</u>	6	2	1	3	0.154 s
98	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.773 s
99	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.548 s
10-0	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.635 s
10-1	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.586 s



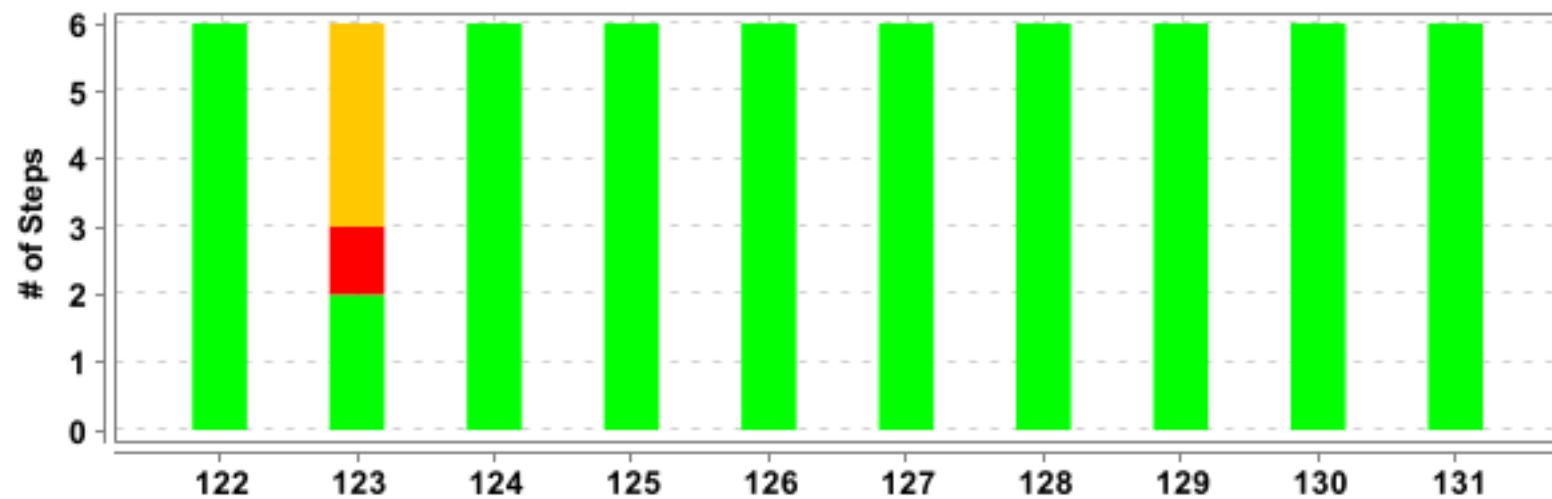
#	Feature Name	Scenario Name	T	P	F	S	Duration
10-2	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.558 s
10-3	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.597 s
10-4	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.525 s
10-5	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.587 s
10-6	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.631 s
10-7	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.555 s
10-8	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.625 s
10-9	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.598 s
11-0	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.802 s

#	Feature Name	Scenario Name	<i>T</i>	<i>P</i>	<i>F</i>	<i>S</i>	Duration
11-1	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.582 s



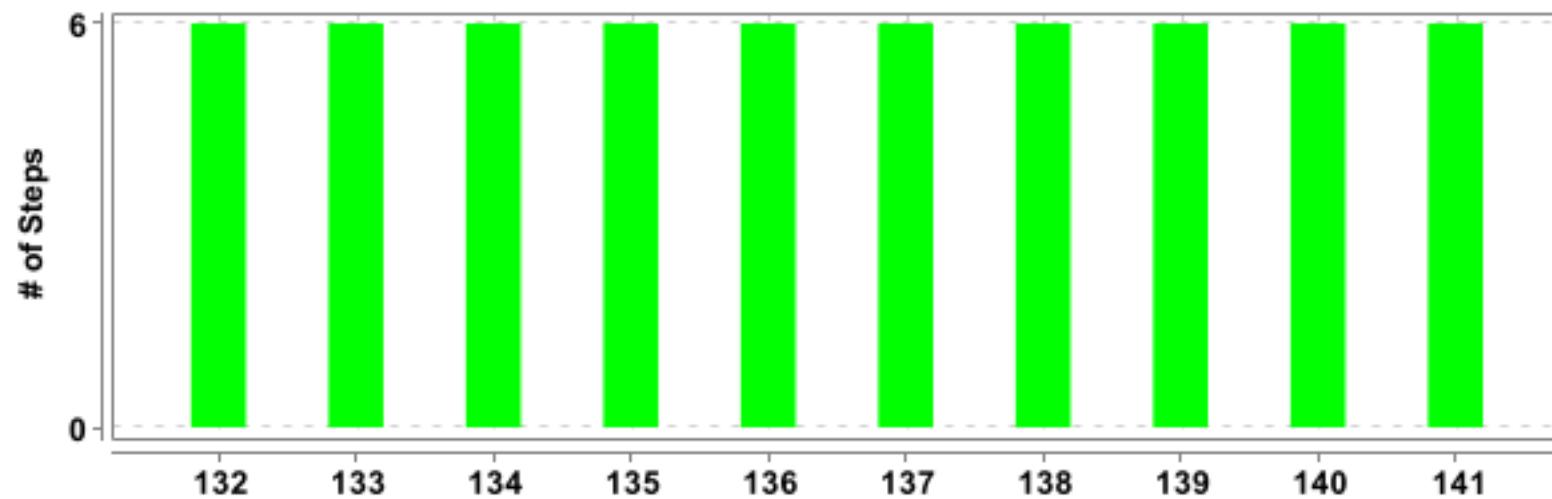
#	Feature Name	Scenario Name	T	P	F	S	Duration
11-2	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.617 s
11-3	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.878 s
11-4	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.580 s
11-5	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.526 s
11-6	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	2	1	3	0.138 s
11-7	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.611 s
11-8	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.527 s
11-9	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.547 s
12-0	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	2	1	3	0.155 s

#	Feature Name	Scenario Name	<i>T</i>	<i>P</i>	<i>F</i>	<i>S</i>	Duration
12-1	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.565 s



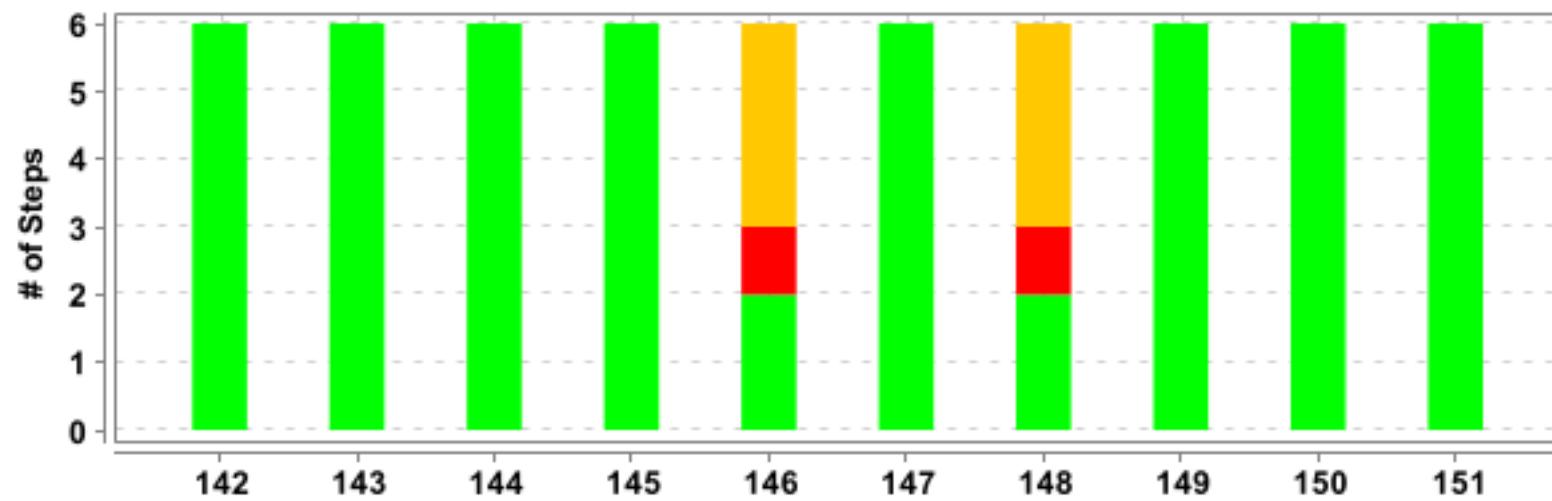
#	Feature Name	Scenario Name	T	P	F	S	Duration
12-2	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.519 s
12-3	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	2	1	3	0.148 s
12-4	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.622 s
12-5	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.464 s
12-6	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.538 s
12-7	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.720 s
12-8	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.678 s
12-9	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.597 s
13-0	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.548 s

#	Feature Name	Scenario Name	<i>T</i>	<i>P</i>	<i>F</i>	<i>S</i>	Duration
13-1	<u>To check The POST method for Users</u>	<u>Validate POST Users</u>	6	6	0	0	0.569 s



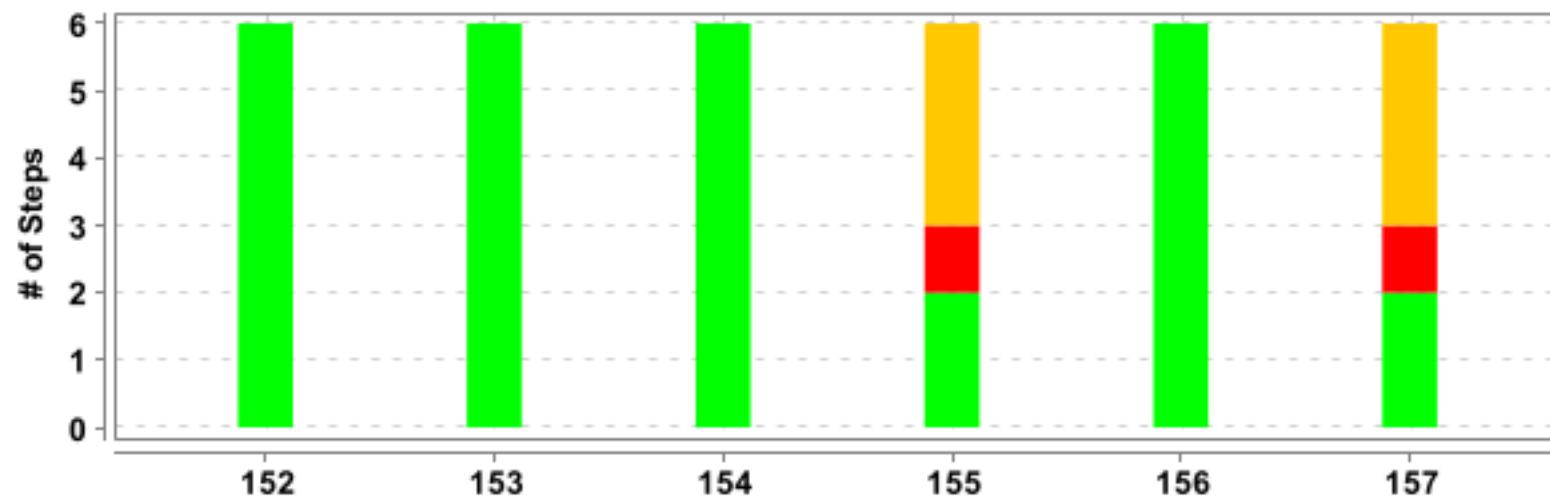
#	Feature Name	Scenario Name	T	P	F	S	Duration
13-2	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.639 s
13-3	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.589 s
13-4	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.814 s
13-5	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.797 s
13-6	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.672 s
13-7	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.597 s
13-8	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.646 s
13-9	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.577 s
14-0	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.700 s

#	Feature Name	Scenario Name	<i>T</i>	<i>P</i>	<i>F</i>	<i>S</i>	Duration
14-1	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.553 s



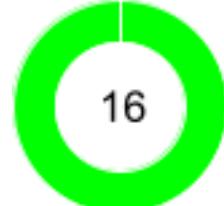
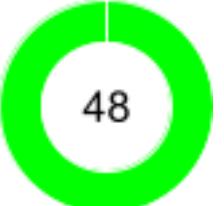
#	Feature Name	Scenario Name	T	P	F	S	Duration
14-2	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.560 s
14-3	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.609 s
14-4	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.631 s
14-5	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.640 s
14-6	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	2	1	3	0.204 s
14-7	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.549 s
14-8	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	2	1	3	0.268 s
14-9	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.623 s
15-0	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.540 s

#	Feature Name	Scenario Name	<i>T</i>	<i>P</i>	<i>F</i>	<i>S</i>	Duration
15-1	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.599 s



#	Feature Name	Scenario Name	T	P	F	S	Duration
15-2	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.558 s
15-3	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.777 s
15-4	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.675 s
15-5	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	2	1	3	0.193 s
15-6	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	6	0	0	0.579 s
15-7	<u>To check The PUT method for Users</u>	<u>Validate PUT method for Users</u>	6	2	1	3	0.327 s

(F)- Check the Authorization for LMS API

PASSED	DURATION - 35.280 s	Scenarios Total - 16 Pass - 16 Fail - 0 Skip - 0		Steps Total - 48 Pass - 48 Fail - 0 Skip - 0	
/ 11:15:16.076 AM // 11:15:51.356 AM /					

(S)- Check the Authorization with with different set of username & password

PASSED	DURATION - 4.083 s	Step / Hook Details	Steps Total - 3 Pass - 3 Fail - 0 Skip - 0	
/ 11:15:16.089 AM // 11:15:20.172 AM /				
Check the Authorization for LMS API				

#	Step / Hook Details	Status	Duration
1	Given Skills User with username & password from "Skills_Authorization" and 0 is on Endpoint: url/Skills	PASSED	2.492 s
2	When skills User sends GET request	PASSED	1.354 s
3	Then Skills User Checks for StatusCode StatusCode and StatusMessage from "Skills_Authorization" sheet and 0 row	PASSED	0.203 s

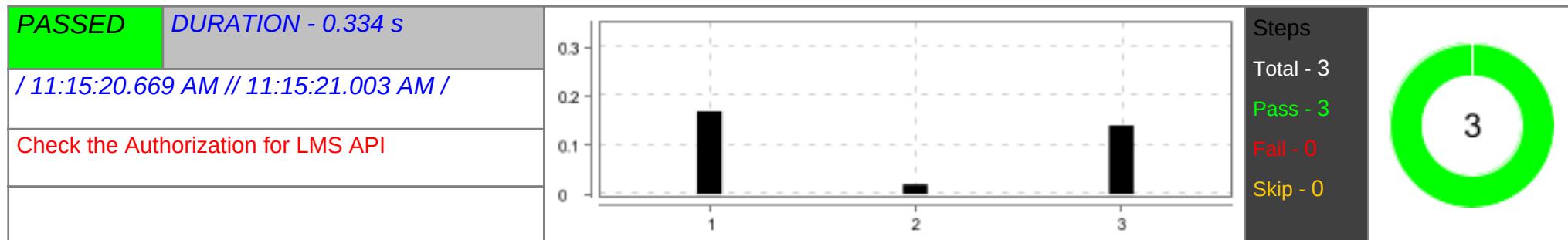
(S)- Check the Authorization with with different set of username & password

PASSED	DURATION - 0.315 s	Step / Hook Details	Steps Total - 3 Pass - 3 Fail - 0 Skip - 0	
/ 11:15:20.308 AM // 11:15:20.623 AM /				
Check the Authorization for LMS API				

#	Step / Hook Details	Status	Duration
1	Given Skills User with username & password from "Skills_Authorization" and 1 is on Endpoint: url/Skills	PASSED	0.170 s

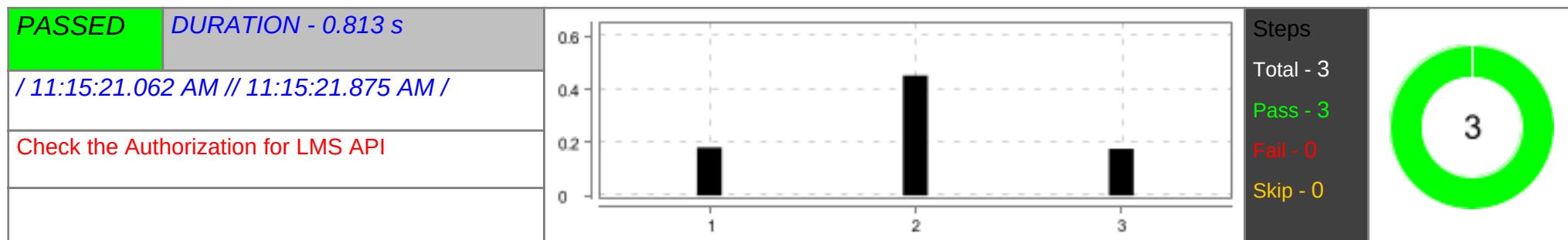
#	Step / Hook Details	Status	Duration
2	When skills User sends GET request	PASSED	0.022 s
3	Then Skills User Checks for StatusCode StatusCode and StatusMessage from "Skills_Authorization" sheet and 1 row	PASSED	0.118 s

(S)- Check the Authorization with with different set of username & password



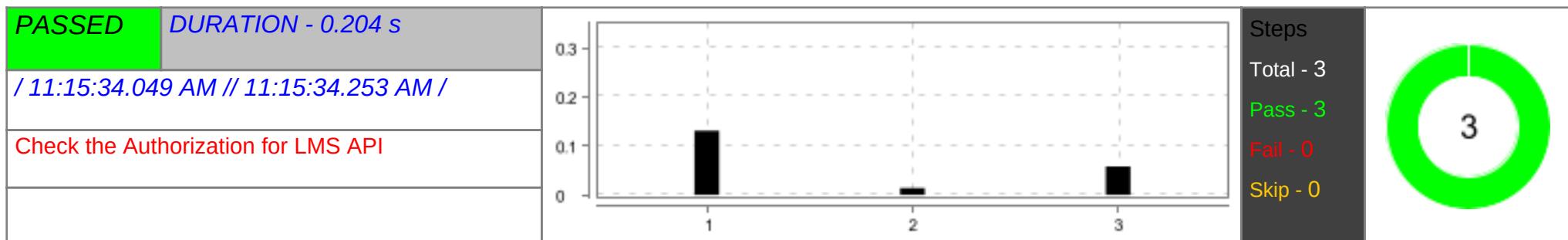
#	Step / Hook Details	Status	Duration
1	Given Skills User with username & password from "Skills_Authorization" and 2 is on Endpoint: url/Skills	PASSED	0.169 s
2	When skills User sends GET request	PASSED	0.020 s
3	Then Skills User Checks for StatusCode StatusCode and StatusMessage from "Skills_Authorization" sheet and 2 row	PASSED	0.140 s

(S)- Check the Authorization with with different set of username & password



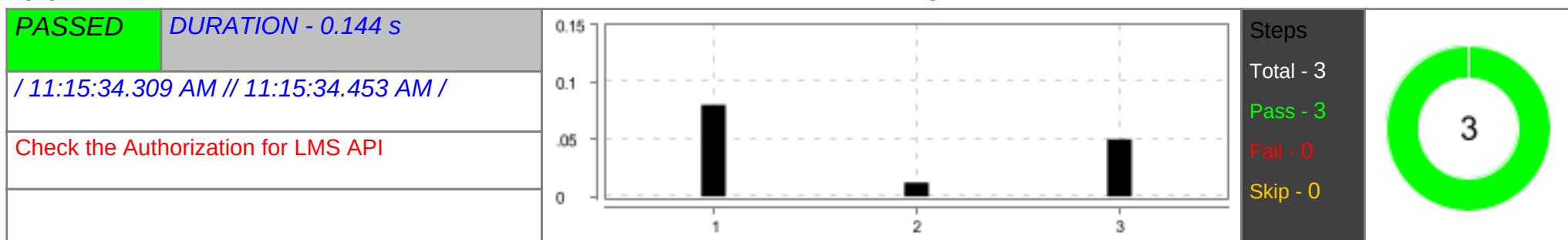
#	Step / Hook Details	Status	Duration
1	Given Skills User with username & password from "Skills_Authorization" and 3 is on Endpoint: url/Skills	PASSED	0.181 s
2	When skills User sends GET request	PASSED	0.453 s
3	Then Skills User Checks for StatusCode StatusCode and StatusMessage from "Skills_Authorization" sheet and 3 row	PASSED	0.176 s

(S)- Check the Authorization with with different set of username & password



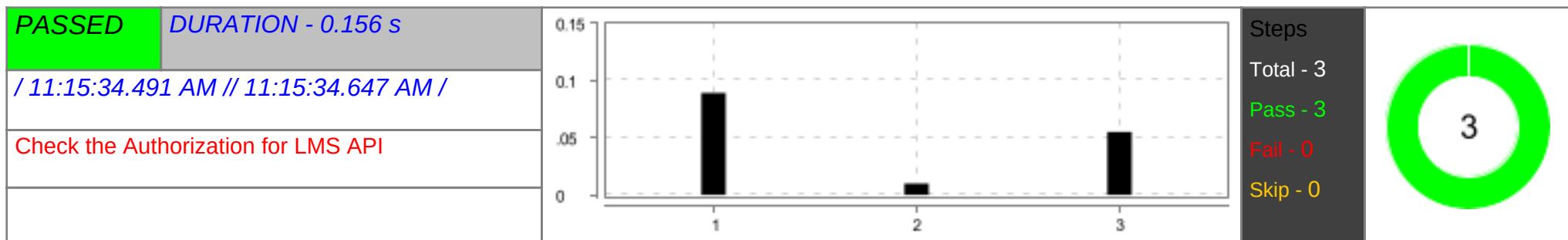
#	Step / Hook Details	Status	Duration
1	Given userSkill User with username & password from "UserSkills_Authorization" and 0 is on Endpoint: url/UserSkills	PASSED	0.131 s
2	When userSkills User sends GET request	PASSED	0.014 s
3	Then userSkills User Checks for StatusCode StatusMessage from "UserSkills_Authorization" sheet and 0 row	PASSED	0.058 s

(S)- Check the Authorization with with different set of username & password



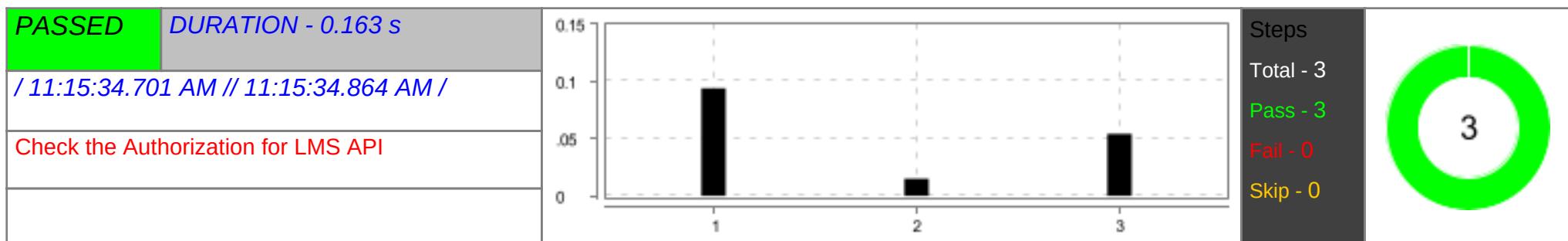
#	Step / Hook Details	Status	Duration
1	Given userSkill User with username & password from "UserSkills_Authorization" and 1 is on Endpoint: url/UserSkills	PASSED	0.080 s
2	When userSkills User sends GET request	PASSED	0.012 s
3	Then userSkills User Checks for StatusCode StatusMessage from "UserSkills_Authorization" sheet and 1 row	PASSED	0.050 s

(S)- Check the Authorization with with different set of username & password



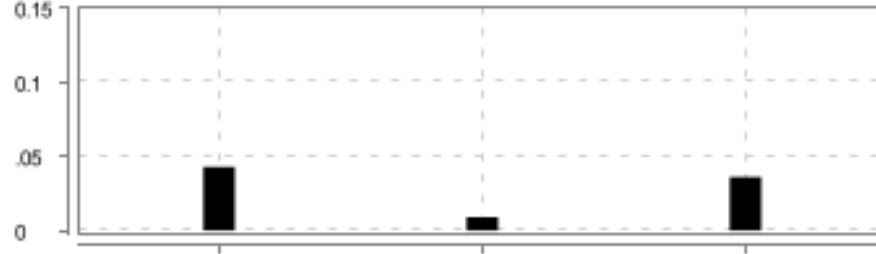
#	Step / Hook Details	Status	Duration
1	Given userSkill User with username & password from "UserSkills_Authorization" and 2 is on Endpoint: url/UserSkills	PASSED	0.089 s
2	When userSkills User sends GET request	PASSED	0.010 s
3	Then userSkills User Checks for StatusCode StatusMessage from "UserSkills_Authorization" sheet and 2 row	PASSED	0.055 s

(S)- Check the Authorization with with different set of username & password



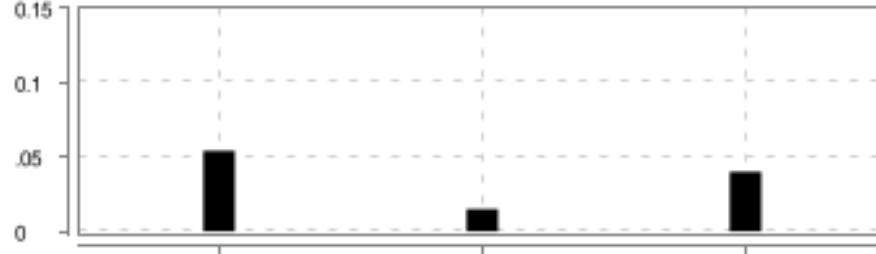
#	Step / Hook Details	Status	Duration
1	Given userSkill User with username & password from "UserSkills_Authorization" and 3 is on Endpoint: url/UserSkills	PASSED	0.094 s
2	When userSkills User sends GET request	PASSED	0.015 s
3	Then userSkills User Checks for StatusCode StatusMessage from "UserSkills_Authorization" sheet and 3 row	PASSED	0.054 s

(S)- Check the Authorization with with different set of username & password

PASSED	DURATION - 0.089 s		Steps Total - 3 Pass - 3 Fail - 0 Skip - 0	
/ 11:15:48.390 AM // 11:15:48.479 AM /				
Check the Authorization for LMS API				

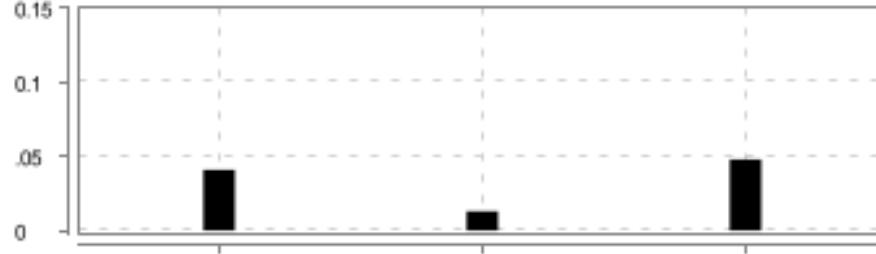
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User with username & password from "UserSkillsMap_Authorization" and 0 is on Endpoint: url/UserSkillsMap"	PASSED	0.043 s
2	When UserSkillsMap User sends GET request	PASSED	0.009 s
3	Then UserSkillsMap User Checks for StatusCode StatusCode and StatusMessage from "UserSkillsMap_Authorization" sheet and 0 row	PASSED	0.036 s

(S)- Check the Authorization with with different set of username & password

PASSED	DURATION - 0.110 s		Steps Total - 3 Pass - 3 Fail - 0 Skip - 0	
/ 11:15:48.508 AM // 11:15:48.618 AM /				
Check the Authorization for LMS API				

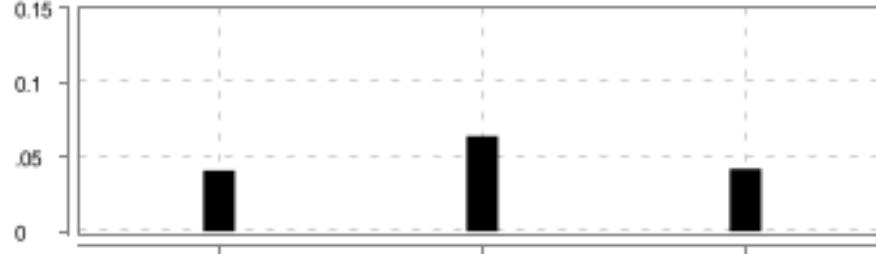
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User with username & password from "UserSkillsMap_Authorization" and 1 is on Endpoint: url/UserSkillsMap"	PASSED	0.054 s
2	When UserSkillsMap User sends GET request	PASSED	0.015 s
3	Then UserSkillsMap User Checks for StatusCode StatusCode and StatusMessage from "UserSkillsMap_Authorization" sheet and 1 row	PASSED	0.040 s

(S)- Check the Authorization with with different set of username & password

PASSED	DURATION - 0.103 s		Steps Total - 3 Pass - 3 Fail - 0 Skip - 0	
/ 11:15:48.650 AM // 11:15:48.753 AM /				
Check the Authorization for LMS API				

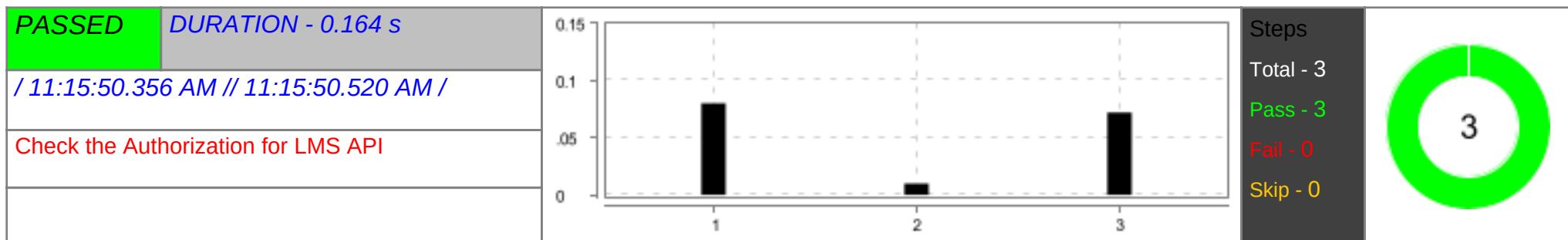
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User with username & password from "UserSkillsMap_Authorization" and 2 is on Endpoint: url/UserSkillsMap"	PASSED	0.041 s
2	When UserSkillsMap User sends GET request	PASSED	0.013 s
3	Then UserSkillsMap User Checks for StatusCode StatusCode and StatusMessage from "UserSkillsMap_Authorization" sheet and 2 row	PASSED	0.048 s

(S)- Check the Authorization with with different set of username & password

PASSED	DURATION - 0.148 s		Steps Total - 3 Pass - 3 Fail - 0 Skip - 0	
/ 11:15:48.790 AM // 11:15:48.938 AM /				
Check the Authorization for LMS API				

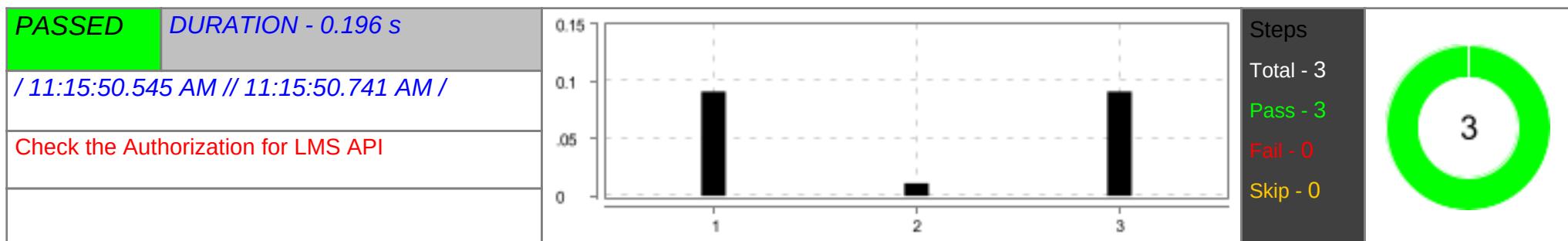
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User with username & password from "UserSkillsMap_Authorization" and 3 is on Endpoint: url/UserSkillsMap"	PASSED	0.041 s
2	When UserSkillsMap User sends GET request	PASSED	0.064 s
3	Then UserSkillsMap User Checks for StatusCode StatusCode and StatusMessage from "UserSkillsMap_Authorization" sheet and 3 row	PASSED	0.042 s

(S)- Check the Authorization with with different set of username & password



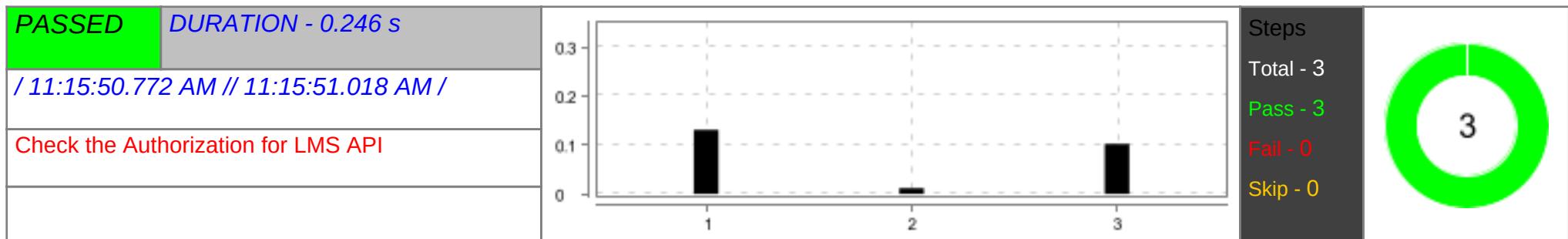
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users username & password from "Users_Authorization" and 0	PASSED	0.080 s
2	When UsersAPI User sends GET request	PASSED	0.010 s
3	Then UsersAPI User Checks for StatusCode StatusMessage from "Users_Authorization" sheet and 0 row	PASSED	0.072 s

(S)- Check the Authorization with with different set of username & password



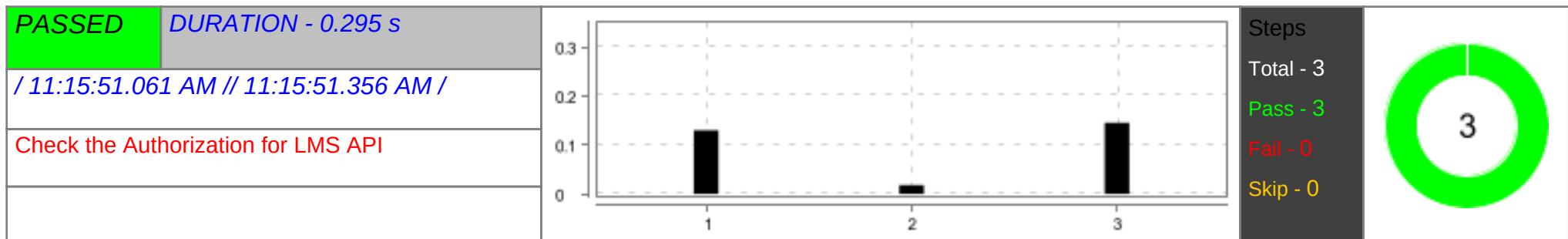
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users username & password from "Users_Authorization" and 1	PASSED	0.091 s
2	When UsersAPI User sends GET request	PASSED	0.011 s
3	Then UsersAPI User Checks for StatusCode StatusMessage from "Users_Authorization" sheet and 1 row	PASSED	0.091 s

(S)- Check the Authorization with with different set of username & password



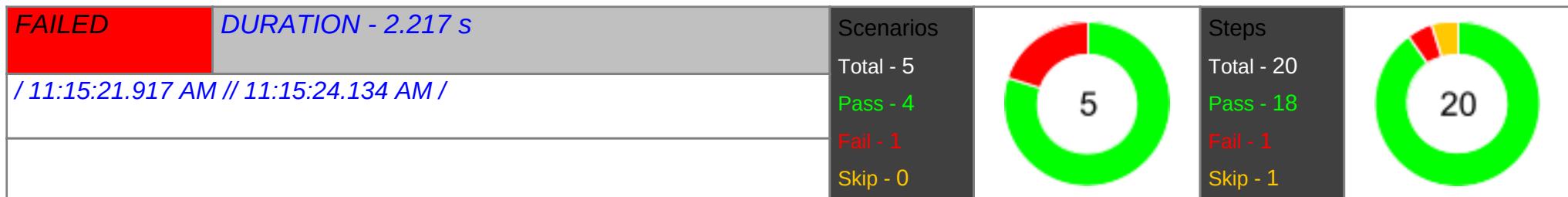
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users username & password from "Users_Authorization" and 2	PASSED	0.131 s
2	When UsersAPI User sends GET request	PASSED	0.012 s
3	Then UsersAPI User Checks for StatusCode StatusMessage from "Users_Authorization" sheet and 2 row	PASSED	0.102 s

(S)- Check the Authorization with with different set of username & password

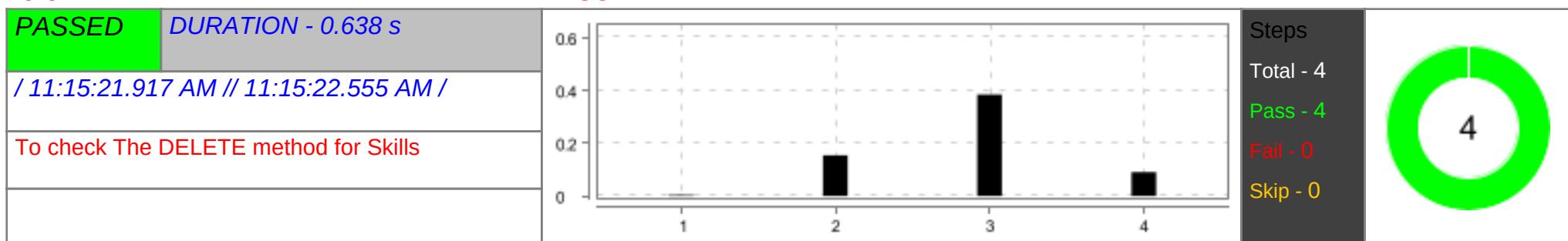


#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users username & password from "Users_Authorization" and 3	PASSED	0.130 s
2	When UsersAPI User sends GET request	PASSED	0.018 s
3	Then UsersAPI User Checks for StatusCode StatusMessage from "Users_Authorization" sheet and 3 row	PASSED	0.145 s

(F)- To check The DELETE method for Skills

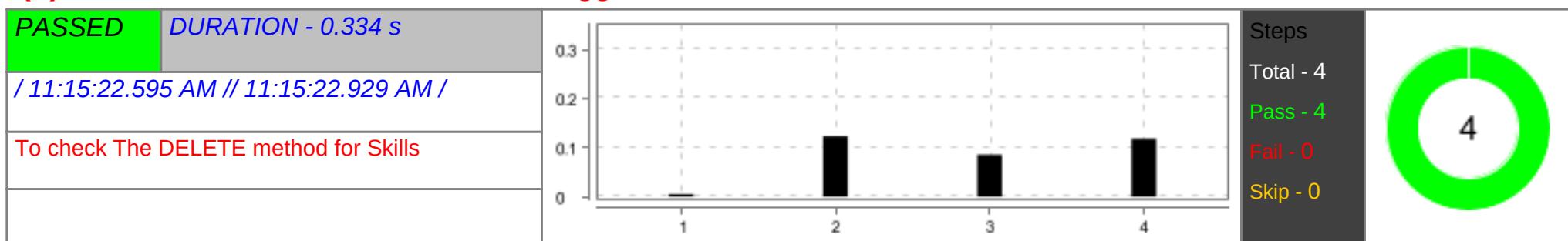


(S)- Validate DELETE Skills when user logged with UserName with Password



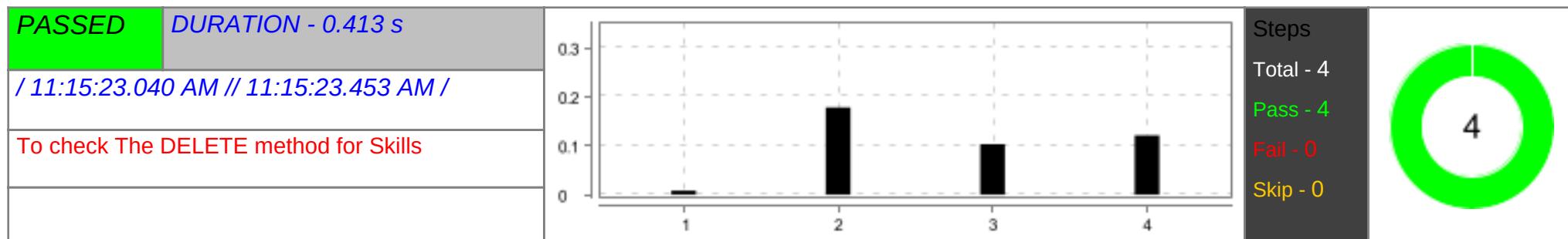
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.004 s
2	When skills User sends DELETE skill id ON DELETE Method from "Skills_POST" and 0	PASSED	0.154 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_DELETE" sheet and 0 row	PASSED	0.385 s
4	And skills User checks the Database to validate deletion from "Skills_POST" sheet and 0 row	PASSED	0.091 s

(S)- Validate DELETE Skills when user logged with UserName with Password



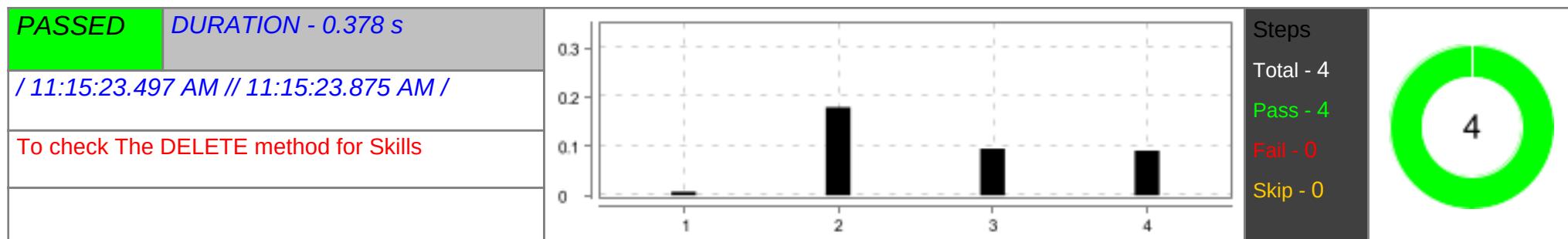
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.005 s
2	When skills User sends DELETE skill id ON DELETE Method from "Skills_DELETE" and 1	PASSED	0.123 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_DELETE" sheet and 1 row	PASSED	0.085 s
4	And skills User checks the Database to validate deletion from "Skills_DELETE" sheet and 1 row	PASSED	0.118 s

(S)- Validate DELETE Skills when user logged with UserName with Password



#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.008 s
2	When skills User sends DELETE skill id ON DELETE Method from "Skills_DELETE" and 2	PASSED	0.178 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_DELETE" sheet and 2 row	PASSED	0.103 s
4	And skills User checks the Database to validate deletion from "Skills_DELETE" sheet and 2 row	PASSED	0.121 s

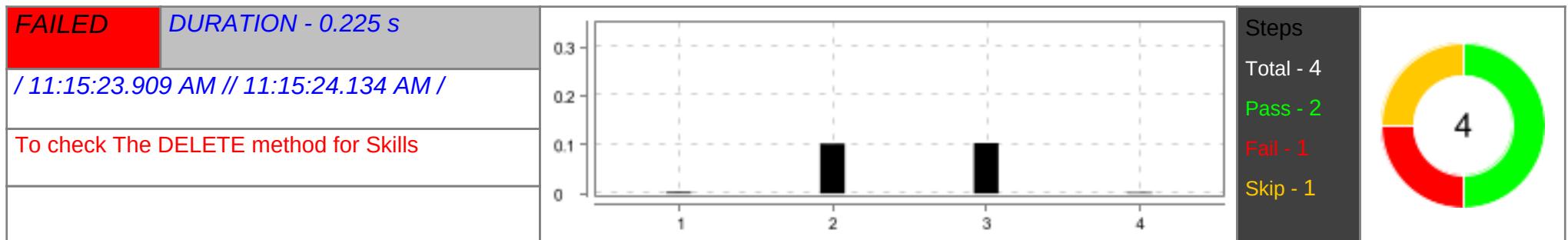
(S)- Validate DELETE Skills when user logged with UserName with Password



#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.007 s
2	When skills User sends DELETE skill id ON DELETE Method from "Skills_DELETE" and 3	PASSED	0.179 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_DELETE" sheet and 3 row	PASSED	0.095 s

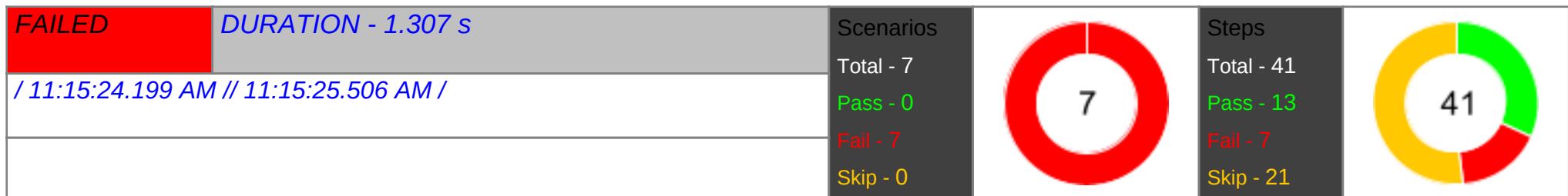
#	Step / Hook Details	Status	Duration
4	And skills User checks the Database to validate deletion from "Skills_DELETE" sheet and 3 row	PASSED	0.091 s

(S)- Validate DELETE Skills when user logged with UserName with Password

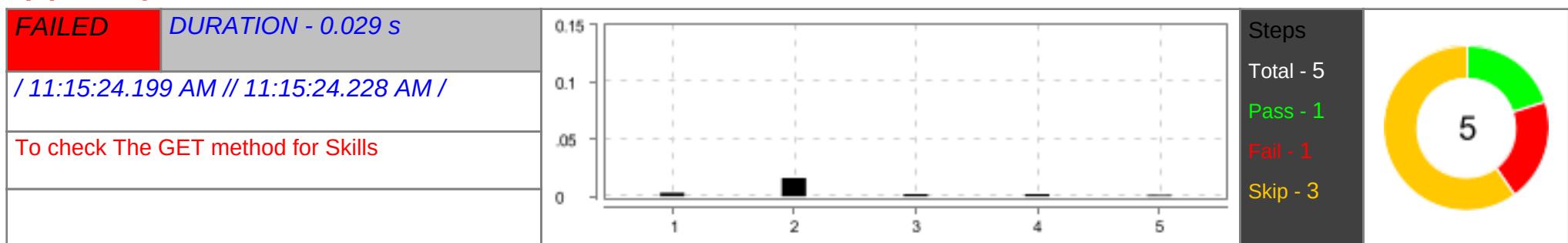


#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends DELETE skill id ON DELETE Method from "Skills_DELETE" and 4	PASSED	0.102 s
3	<p>Then skills User validates the StatusCode and StatusMessage from "Skills_DELETE" sheet and 4 row</p> <pre>com.fasterxml.jackson.core.JsonParseException: Unexpected character ('<' (code 60)): expected a valid value (JSON String, Number, Array, Object or token 'null', 'true' or 'false') at [Source: (String)"<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <title>404 Not Found</title> <h1>Not Found</h1> <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p> "; line: 1, column: 2] at com.fasterxml.jackson.core.JsonParser._constructError(JsonParser.java:2391) at com.fasterxml.jackson.core.base.ParserMinimalBase._reportError(ParserMinimalBase.java:735) at com.fasterxml.jackson.core.base.ParserMinimalBase._reportUnexpectedChar(ParserMinimalBase.java:659) at com.fasterxml.jackson.core.json.ReaderBasedJsonParser._handleOddValue(ReaderBasedJsonParser.java:2005) at com.fasterxml.jackson.core.json.ReaderBasedJsonParser.nextToken(ReaderBasedJsonParser.java:802) at com.fasterxml.jackson.databind.ObjectMapper._initForReading(ObjectMapper.java:4762) at com.fasterxml.jackson.databind.ObjectMapper._readMapAndClose(ObjectMapper.java:4668) at com.fasterxml.jackson.databind.ObjectMapper.readValue(ObjectMapper.java:3630) at com.fasterxml.jackson.databind.ObjectMapper.readValue(ObjectMapper.java:3598) at step_definition.SkillMaster.extractResponse(SkillMaster.java:305) at step_definition.SkillMaster.user_validates_the_statuscode_and_statusmessage_from(SkillMaster.java:291) at ?.skills User validates the StatusCode and StatusMessage from "Skills_DELETE" sheet and 4 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_DELETE.feature:7)</pre> <p>* Not displayable characters are replaced by '?'.</p>	FAILED	0.103 s
4	And skills User checks the Database to validate deletion from "Skills_DELETE" sheet and 4 row	SKIPPED	0.002 s

(F)- To check The GET method for Skills



(S)- Verify GET all Skills for Skill Master



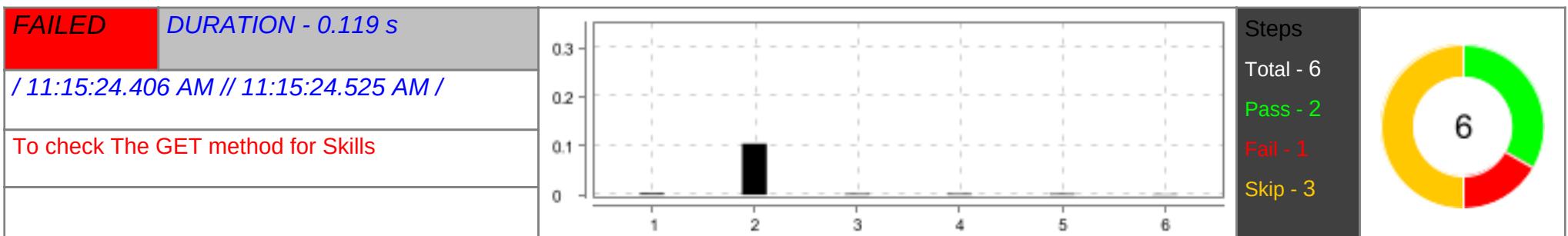
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends GET request <pre>java.net.URISyntaxException: Illegal character in path at index 29: http://127.0.0.1:5000/Skills/ at java.base/java.net.URI\$Parser.fail(URI.java:2913) at java.base/java.net.URI\$Parser.checkChars(URI.java:3084) at java.base/java.net.URI\$Parser.parseHierarchical(URI.java:3166) at java.base/java.net.URI\$Parser.parse(URI.java:3114) at java.base/java.net.URI.<init>(URI.java:600) at java.base/jdk.internal.reflect.GeneratedConstructorAccessor50.newInstance(Unknown Source) at java.base/jdk.internal.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45) at java.base/java.lang.reflect.Constructor.newInstance(Constructor.java:490) at org.codehaus.groovy.reflection.CachedConstructor.invoke(CachedConstructor.java:83) at org.codehaus.groovy.runtime.callsite.ConstructorSite\$ConstructorSiteNoUnwrapNoCoerce.callConstructor(ConstructorSite.java:105) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callConstructor(AbstractCallSite.java:250) at io.restassured.internal.RequestSpecificationImpl.getTargetPath(RequestSpecificationImpl.groovy:1452) at jdk.internal.reflect.GeneratedMethodAccessor111.invoke(Unknown Source) at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.base/java.lang.reflect.Method.invoke(Method.java:566) at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:98)</pre>	FAILED	0.016 s

#	Step / Hook Details	Status	Duration
	<pre> at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:325) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1225) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1034) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:822) at io.restassured.internal.RequestSpecificationImpl.invokeMethod(RequestSpecificationImpl.groovy) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.call(PogoInterceptableSite.java:47) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.callCurrent(PogoInterceptableSite.java:57) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:169) at io.restassured.internal.RequestSpecificationImpl.getPathParamPlaceholders(RequestSpecificationImpl.groovy:2063) at jdk.internal.reflect.GeneratedMethodAccessor176.invoke(Unknown Source) at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.base/java.lang.reflect.Method.invoke(Method.java:566) at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:98) at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:325) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1225) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1034) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:822) at io.restassured.internal.RequestSpecificationImpl.invokeMethod(RequestSpecificationImpl.groovy) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.call(PogoInterceptableSite.java:47) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.callCurrent(PogoInterceptableSite.java:57) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:161) at io.restassured.internal.RequestSpecificationImpl.getRedundantNamedPathParamParams(RequestSpecificationImpl.groovy:2329) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.base/java.lang.reflect.Method.invoke(Method.java:566) at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:98) at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:325) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1225) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1034) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:822) at io.restassured.internal.RequestSpecificationImpl.invokeMethod(RequestSpecificationImpl.groovy) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.call(PogoInterceptableSite.java:47) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.callCurrent(PogoInterceptableSite.java:57) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:161) at io.restassured.internal.RequestSpecificationImpl.assertCorrectNumberOfPathParamParams(RequestSpecificationImpl.groovy:1300) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.base/java.lang.reflect.Method.invoke(Method.java:566) at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:98) at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:325) </pre>		

#	Step / Hook Details	Status	Duration
	<pre> at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1225) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1034) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:822) at io.restassured.internal.RequestSpecificationImpl.invokeMethod(RequestSpecificationImpl.groovy) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.call(PogoInterceptableSite.java:47) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.callCurrent(PogoInterceptableSite.java:57) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:161) at io.restassured.internal.RequestSpecificationImpl.sendRequest(RequestSpecificationImpl.groovy:1228) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.base/java.lang.reflect.Method.invoke(Method.java:566) at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:98) at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:325) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1225) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1034) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:822) at io.restassured.internal.RequestSpecificationImpl.invokeMethod(RequestSpecificationImpl.groovy) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.call(PogoInterceptableSite.java:47) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.call(AbstractCallSite.java:152) at io.restassured.internal.filter.SendRequestFilter.filter(SendRequestFilter.groovy:30) at io.restassured.filter.Filter\$filter\$0.call(Unknown Source) at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCall(CallSiteArray.java:47) at io.restassured.filter.Filter\$filter.call(Unknown Source) at io.restassured.internal.filter.FilterContextImpl.next(FilterContextImpl.groovy:72) at io.restassured.filter.time.TimingFilter.filter(TimingFilter.java:56) at io.restassured.filter.Filter\$filter.call(Unknown Source) at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCall(CallSiteArray.java:47) at io.restassured.filter.Filter\$filter\$0.call(Unknown Source) at io.restassured.internal.filter.FilterContextImpl.next(FilterContextImpl.groovy:72) at io.restassured.filter.FilterContext\$next.call(Unknown Source) at io.restassured.internal.RequestSpecificationImpl.applyPathParamsAndSendRequest(RequestSpecificationImpl.groovy:1749) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.base/java.lang.reflect.Method.invoke(Method.java:566) at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:98) at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:325) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1225) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1034) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:822) at io.restassured.internal.RequestSpecificationImpl.invokeMethod(RequestSpecificationImpl.groovy) </pre>		

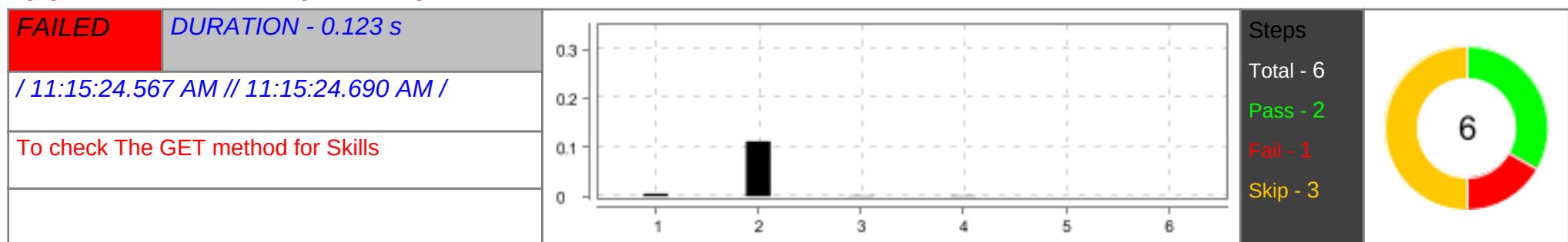
#	Step / Hook Details	Status	Duration
	at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.call(PogoInterceptableSite.java:47) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.callCurrent(PogoInterceptableSite.java:57) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:185) at io.restassured.internal.RequestSpecificationImpl.applyPathParamsAndSendRequest(RequestSpecificationImpl.groovy:1755) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.base/java.lang.reflect.Method.invoke(Method.java:566) at org.codehaus.groovy.reflection.CachedMethod.invoke(CachedMethod.java:98) at groovy.lang.MetaMethod.doMethodInvoke(MetaMethod.java:325) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1225) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:1034) at groovy.lang.MetaClassImpl.invokeMethod(MetaClassImpl.java:822) at io.restassured.internal.RequestSpecificationImpl.invokeMethod(RequestSpecificationImpl.groovy) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.call(PogoInterceptableSite.java:47) at org.codehaus.groovy.runtime.callsite.PogoInterceptableSite.callCurrent(PogoInterceptableSite.java:57) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:185) at io.restassured.internal.RequestSpecificationImpl.get(RequestSpecificationImpl.groovy:171) at io.restassured.internal.RequestSpecificationImpl.get(RequestSpecificationImpl.groovy) at util.Send_Request_For_Method.Sent_request(Send_Request_For_Method.java:237) at step_definition.SkillMaster.user_sends_get_request_on_skills(SkillMaster.java:75) at ?.skills User sends GET request(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/ SkillMaster_GET.feature:6) * Not displayable characters are replaced by '?'.		
3	Then skills User validates StatusCode	SKIPPED	0.002 s
4	And skills JSON schema is valid	SKIPPED	0.002 s
5	And check the Database for skills	SKIPPED	0.001 s

(S)- Validate Skills Api with specific ID



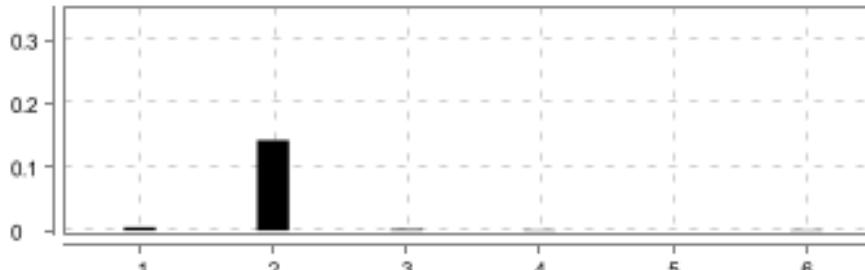
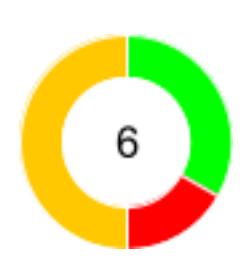
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When User sends GET request on skill id from "Skills_GET" and 0	PASSED	0.104 s
3	Then User validates the StatusCode and StatusMessage from "Skills_GET" sheet and 0 row Step undefined	FAILED	0.002 s
4	And JSON schema is valid for "GET" in Skills	SKIPPED	0.002 s
5	And User should receive a particular skill from "Skills_GET" and 0	SKIPPED	0.002 s
6	And check the Database for validation	SKIPPED	0.001 s

(S)- Validate Skills Api with specific ID



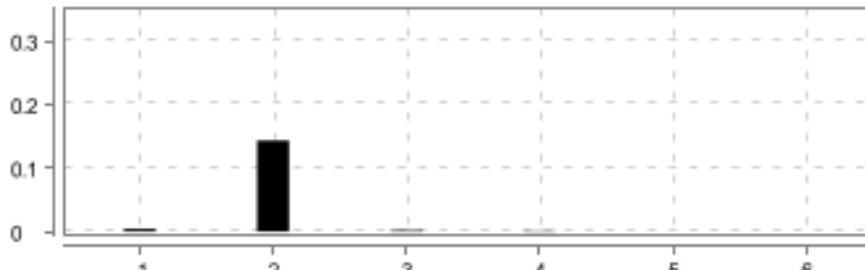
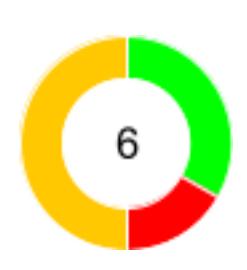
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.005 s
2	When User sends GET request on skill id from "Skills_GET" and 1	PASSED	0.112 s
3	Then User validates the StatusCode and StatusMessage from "Skills_GET" sheet and 1 row Step undefined	FAILED	0.001 s
4	And JSON schema is valid for "GET" in Skills	SKIPPED	0.001 s
5	And User should receive a particular skill from "Skills_GET" and 1	SKIPPED	0.000 s
6	And check the Database for validation	SKIPPED	0.000 s

(S)- Validate Skills Api with specific ID

FAILED	DURATION - 0.157 s		Steps Total - 6 Pass - 2 Fail - 1 Skip - 3	
/ 11:15:24.736 AM // 11:15:24.893 AM /				
To check The GET method for Skills				

#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.005 s
2	When User sends GET request on skill id from "Skills_GET" and 2	PASSED	0.142 s
3	Then User validates the StatusCode and StatusMessage from "Skills_GET" sheet and 2 row Step undefined	FAILED	0.002 s
4	And JSON schema is valid for "GET" in Skills	SKIPPED	0.001 s
5	And User should receive a particular skill from "Skills_GET" and 2	SKIPPED	0.000 s
6	And check the Database for validation	SKIPPED	0.001 s

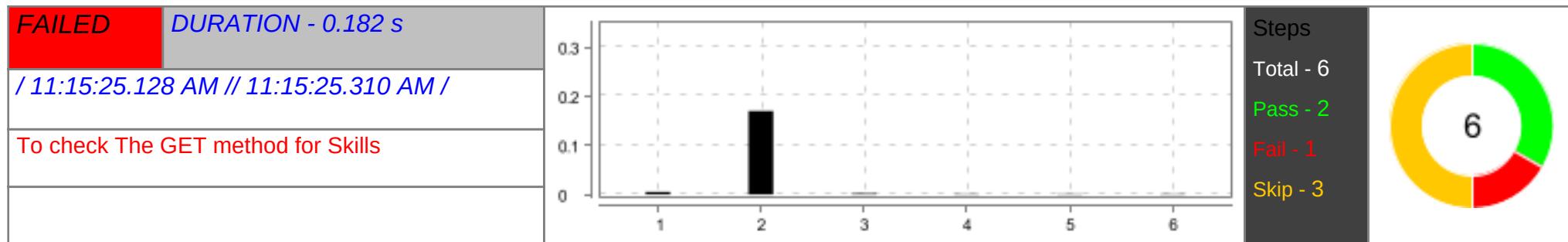
(S)- Validate Skills Api with specific ID

FAILED	DURATION - 0.154 s		Steps Total - 6 Pass - 2 Fail - 1 Skip - 3	
/ 11:15:24.931 AM // 11:15:25.085 AM /				
To check The GET method for Skills				

#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.004 s
2	When User sends GET request on skill id from "Skills_GET" and 3	PASSED	0.143 s
3	Then User validates the StatusCode and StatusMessage from "Skills_GET" sheet and 3 row Step undefined	FAILED	0.002 s
4	And JSON schema is valid for "GET" in Skills	SKIPPED	0.001 s

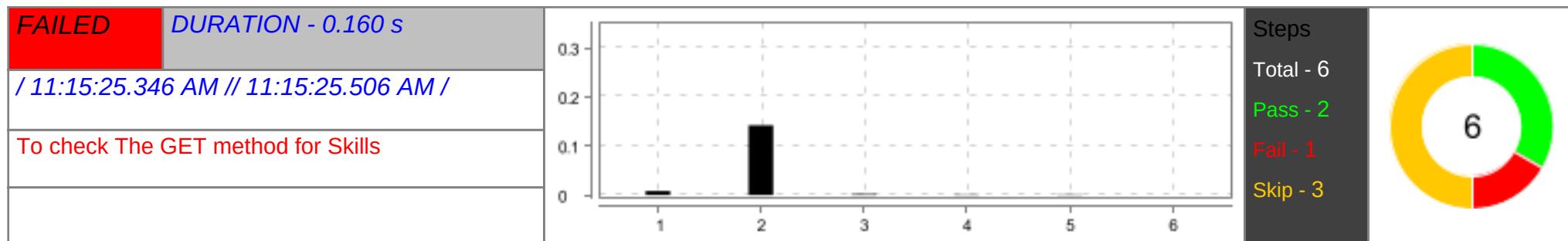
#	Step / Hook Details	Status	Duration
5	And User should receive a particular skill from "Skills_GET" and 3	SKIPPED	0.000 s
6	And check the Database for validation	SKIPPED	0.000 s

(S)- Validate Skills Api with specific ID



#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.005 s
2	When User sends GET request on skill id from "Skills_GET" and 4	PASSED	0.170 s
3	Then User validates the StatusCode and StatusMessage from "Skills_GET" sheet and 4 row Step undefined	FAILED	0.002 s
4	And JSON schema is valid for "GET" in Skills	SKIPPED	0.001 s
5	And User should receive a particular skill from "Skills_GET" and 4	SKIPPED	0.001 s
6	And check the Database for validation	SKIPPED	0.001 s

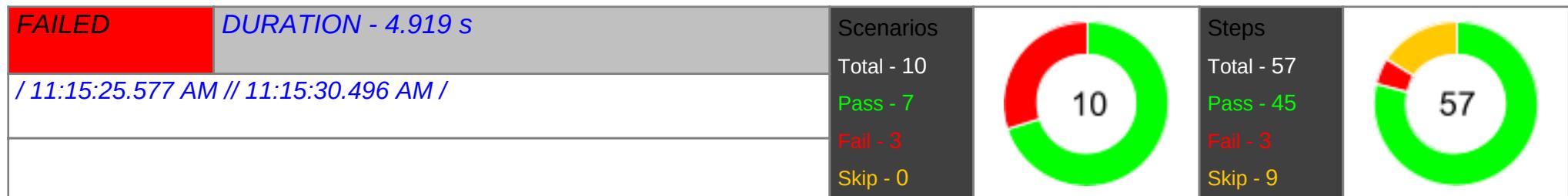
(S)- Validate Skills Api with specific ID



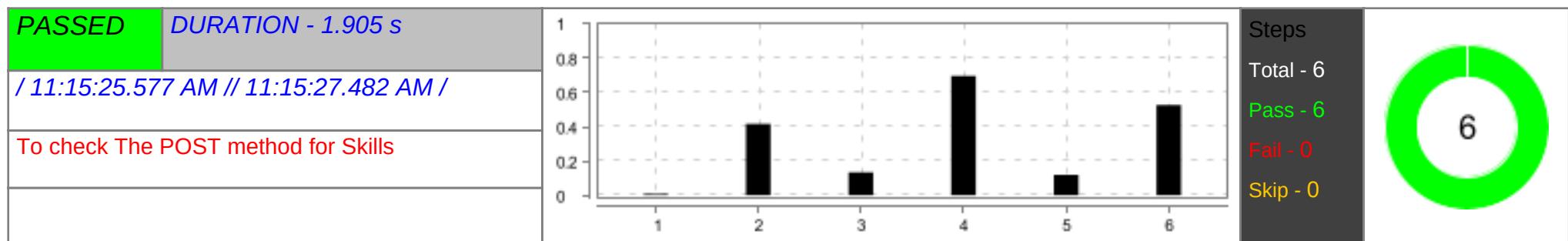
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.008 s
2	When User sends GET request on skill id from "Skills_GET" and 5	PASSED	0.142 s

#	Step / Hook Details	Status	Duration
3	Then User validates the StatusCode and StatusMessage from "Skills_GET" sheet and 5 row Step undefined	FAILED	0.002 s
4	And JSON schema is valid for "GET" in Skills	SKIPPED	0.001 s
5	And User should receive a particular skill from "Skills_GET" and 5	SKIPPED	0.001 s
6	And check the Database for validation	SKIPPED	0.000 s

(F)- To check The POST method for Skills

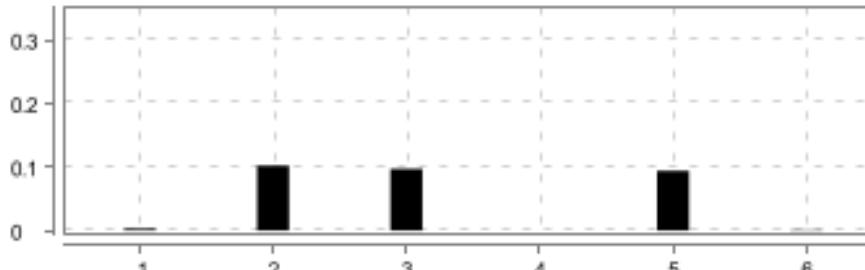


(S)- Verify POST for a Skill when user logged with UserName with Password



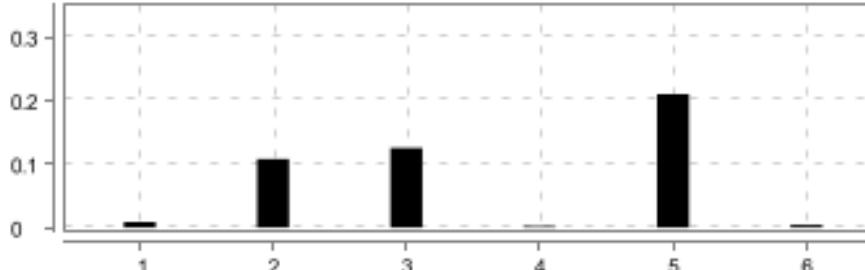
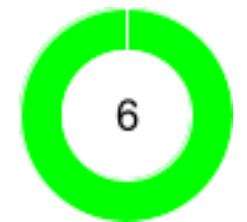
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.007 s
2	When skills User sends POST request body in skills from "Skills_POST" and 0 with valid JSON Schema	PASSED	0.417 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 0 row	PASSED	0.133 s
4	And JSON schema is valid for "POST" in Skills	PASSED	0.696 s
5	And skills User should receive the skill in JSON body from "Skills_POST" and 0	PASSED	0.119 s
6	And skills check the Database	PASSED	0.527 s

(S)- Verify POST for a Skill when user logged with UserName with Password

PASSED	DURATION - 0.300 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:27.519 AM // 11:15:27.819 AM /				
To check The POST method for Skills				

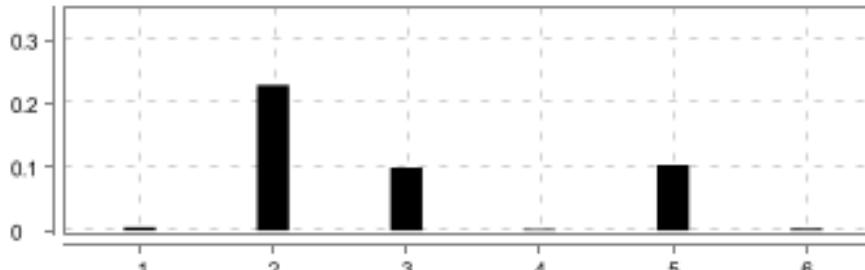
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends POST request body in skills from "Skills_POST" and 1 with valid JSON Schema	PASSED	0.102 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 1 row	PASSED	0.097 s
4	And JSON schema is valid for "POST" in Skills	PASSED	0.000 s
5	And skills User should receive the skill in JSON body from "Skills_POST" and 1	PASSED	0.094 s
6	And skills check the Database	PASSED	0.001 s

(S)- Verify POST for a Skill when user logged with UserName with Password

PASSED	DURATION - 0.463 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:27.859 AM // 11:15:28.322 AM /				
To check The POST method for Skills				

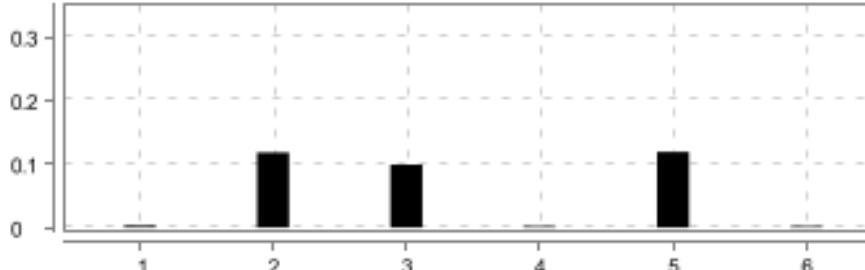
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.009 s
2	When skills User sends POST request body in skills from "Skills_POST" and 2 with valid JSON Schema	PASSED	0.108 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 2 row	PASSED	0.125 s
4	And JSON schema is valid for "POST" in Skills	PASSED	0.002 s
5	And skills User should receive the skill in JSON body from "Skills_POST" and 2	PASSED	0.210 s
6	And skills check the Database	PASSED	0.004 s

(S)- Verify POST for a Skill when user logged with UserName with Password

PASSED	DURATION - 0.442 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:28.361 AM // 11:15:28.803 AM /				
To check The POST method for Skills				

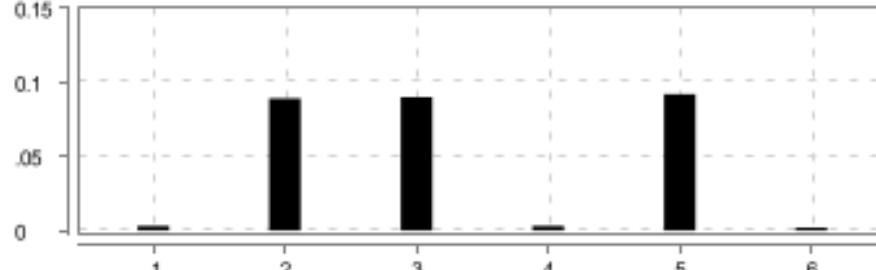
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.005 s
2	When skills User sends POST request body in skills from "Skills_POST" and 3 with valid JSON Schema	PASSED	0.229 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 3 row	PASSED	0.099 s
4	And JSON schema is valid for "POST" in Skills	PASSED	0.002 s
5	And skills User should receive the skill in JSON body from "Skills_POST" and 3	PASSED	0.103 s
6	And skills check the Database	PASSED	0.003 s

(S)- Verify POST for a Skill when user logged with UserName with Password

PASSED	DURATION - 0.346 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:28.844 AM // 11:15:29.190 AM /				
To check The POST method for Skills				

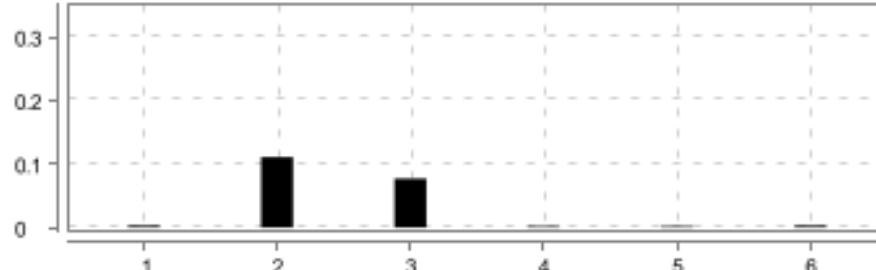
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends POST request body in skills from "Skills_POST" and 4 with valid JSON Schema	PASSED	0.118 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 4 row	PASSED	0.099 s
4	And JSON schema is valid for "POST" in Skills	PASSED	0.002 s
5	And skills User should receive the skill in JSON body from "Skills_POST" and 4	PASSED	0.119 s
6	And skills check the Database	PASSED	0.002 s

(S)- Verify POST for a Skill when user logged with UserName with Password

PASSED	DURATION - 0.284 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:29.217 AM // 11:15:29.501 AM /				
To check The POST method for Skills				

#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends POST request body in skills from "Skills_POST" and 5 with valid JSON Schema	PASSED	0.089 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 5 row	PASSED	0.090 s
4	And JSON schema is valid for "POST" in Skills	PASSED	0.003 s
5	And skills User should receive the skill in JSON body from "Skills_POST" and 5	PASSED	0.092 s
6	And skills check the Database	PASSED	0.002 s

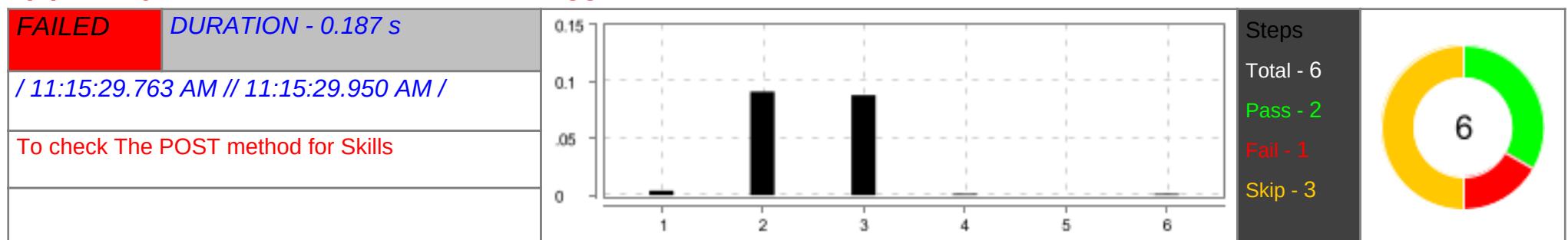
(S)- Verify POST for a Skill when user logged with UserName with Password

FAILED	DURATION - 0.200 s		Steps Total - 6 Pass - 2 Fail - 1 Skip - 3	
/ 11:15:29.527 AM // 11:15:29.727 AM /				
To check The POST method for Skills				

#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends POST request body in skills from "Skills_POST" and 6 with valid JSON Schema	PASSED	0.111 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 6 row java.lang.AssertionError: expected [201] but found [400] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137)	FAILED	0.077 s

#	Step / Hook Details	Status	Duration
	at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.SkillMaster.user_validates_the_statuscode_and_statusmessage_from(SkillMaster.java:289) at ?.skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 6 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_POST.feature:7) * Not displayable characters are replaced by '?'. 4 And JSON schema is valid for "POST" in Skills 5 And skills User should receive the skill in JSON body from "Skills_POST" and 6 6 And skills check the Database		
4	And JSON schema is valid for "POST" in Skills	SKIPPED	0.002 s
5	And skills User should receive the skill in JSON body from "Skills_POST" and 6	SKIPPED	0.002 s
6	And skills check the Database	SKIPPED	0.003 s

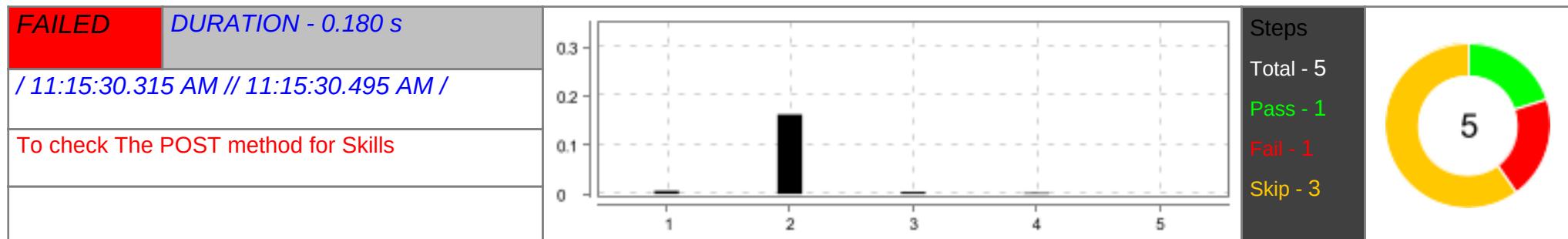
(S)- Verify POST for a Skill when user logged with UserName with Password



#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.004 s
2	When skills User sends POST request body in skills from "Skills_POST" and 7 with valid JSON Schema	PASSED	0.091 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 7 row java.lang.AssertionError: expected [Failed to create.Missing fields or Invalid Skill name] but found [Failed to create-Duplicate skill name] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:442) at step_definition.SkillMaster.user_validates_the_statuscode_and_statusmessage_from(SkillMaster.java:294) at ?.skills User validates the StatusCode and StatusMessage from "Skills_POST" sheet and 7 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_POST.feature:7) * Not displayable characters are replaced by '?'. 4 And JSON schema is valid for "POST" in Skills 5 And skills User should receive the skill in JSON body from "Skills_POST" and 7	FAILED	0.088 s
4	And JSON schema is valid for "POST" in Skills	SKIPPED	0.001 s
5	And skills User should receive the skill in JSON body from "Skills_POST" and 7	SKIPPED	0.000 s

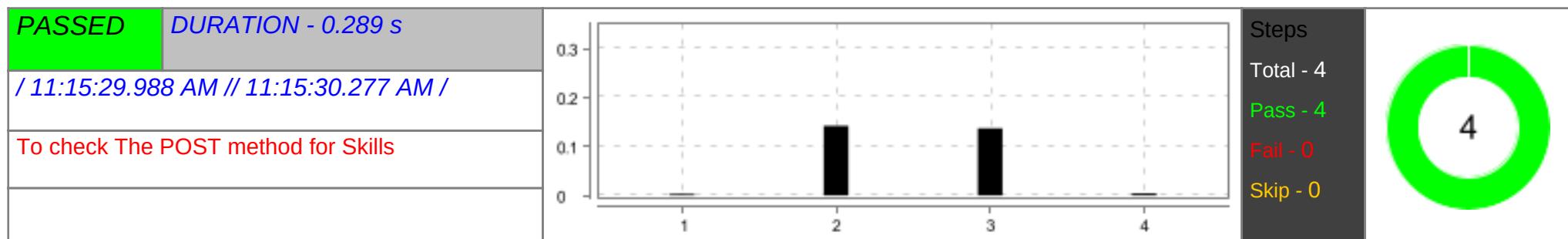
#	Step / Hook Details	Status	Duration
6	And skills check the Database	SKIPPED	0.001 s

(S)- Verify POST for a Skill when user logged with UserName with Password



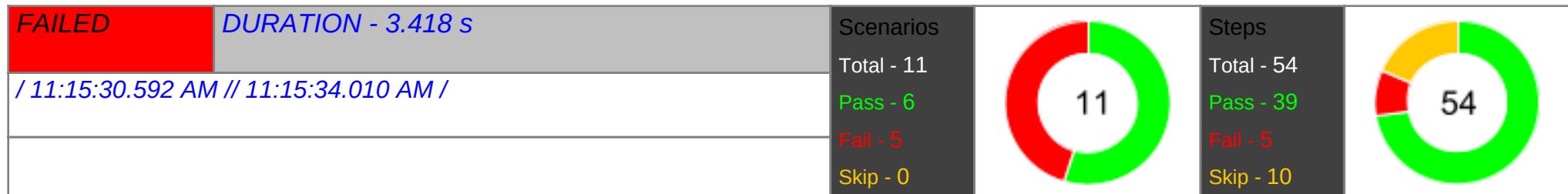
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.006 s
2	When skills User sends POST request body in skills from "Invalid_content_POST" and 0 with valid JSON Schema java.lang.NullPointerException at util.ExcelReader.readSheet(ExcelReader.java:58) at util.ExcelReader.getData(ExcelReader.java:28) at util.Send_Request_For_Method.Sent_request(Send_Request_For_Method.java:89) at step_definition.SkillMaster.user_sends_POST_request_body_from_and(SkillMaster.java:100) at ?.skills User sends POST request body in skills from "Invalid_content_POST" and 0 with valid JSON Schema(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_POST.feature:33) * Not displayable characters are replaced by '?'. 	FAILED	0.162 s
3	Then skills User validates the StatusCode and StatusMessage from "Invalid_content_POST" sheet and 0 row	SKIPPED	0.004 s
4	And skills User should receive the skill in JSON body from "Invalid_content_POST" and 0	SKIPPED	0.002 s
5	And skills check the Database	SKIPPED	0.000 s

(S)- Verify if POST fails for improper format when user logged with UserName with Password

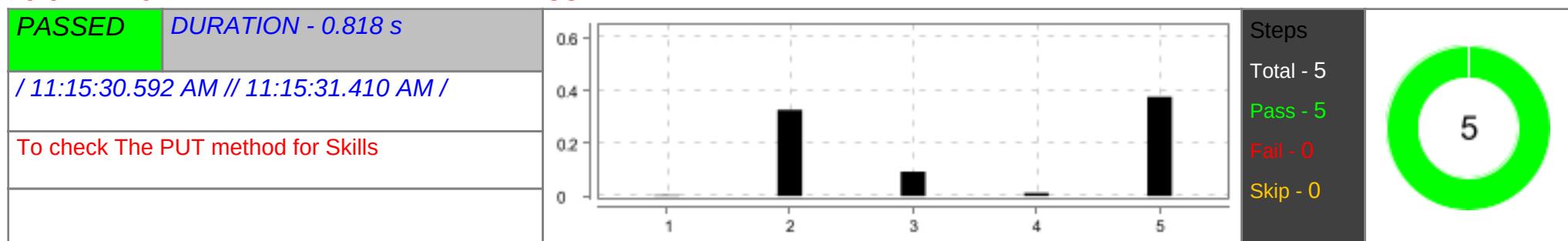


#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends POST request body in text format skills from "Text_POST" and 0	PASSED	0.142 s
3	Then skills User validates the StatusCode and StatusMessage from "Text_POST" sheet and 0 row	PASSED	0.137 s
4	And skills check the Database	PASSED	0.004 s

(F)- To check The PUT method for Skills

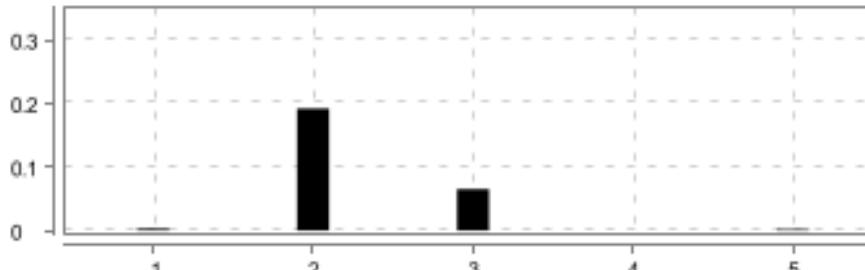


(S)- Verify PUT for a Skill when user logged with UserName with Password



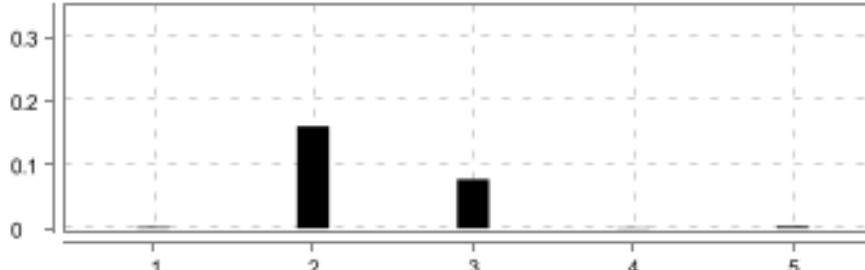
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 0 with valid JSON Schema	PASSED	0.327 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 0 row	PASSED	0.092 s
4	And JSON schema is valid for "PUT" in Skills	PASSED	0.011 s
5	And skills check the Database	PASSED	0.376 s

(S)- Verify PUT for a Skill when user logged with UserName with Password

PASSED	DURATION - 0.265 s		Steps Total - 5 Pass - 5 Fail - 0 Skip - 0	
/ 11:15:31.435 AM // 11:15:31.700 AM /				
To check The PUT method for Skills				

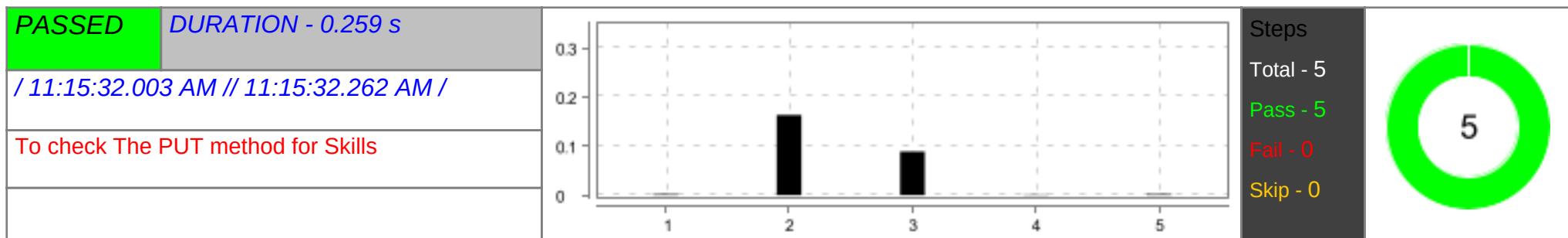
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 1 with valid JSON Schema	PASSED	0.192 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 1 row	PASSED	0.065 s
4	And JSON schema is valid for "PUT" in Skills	PASSED	0.000 s
5	And skills check the Database	PASSED	0.002 s

(S)- Verify PUT for a Skill when user logged with UserName with Password

PASSED	DURATION - 0.245 s		Steps Total - 5 Pass - 5 Fail - 0 Skip - 0	
/ 11:15:31.733 AM // 11:15:31.978 AM /				
To check The PUT method for Skills				

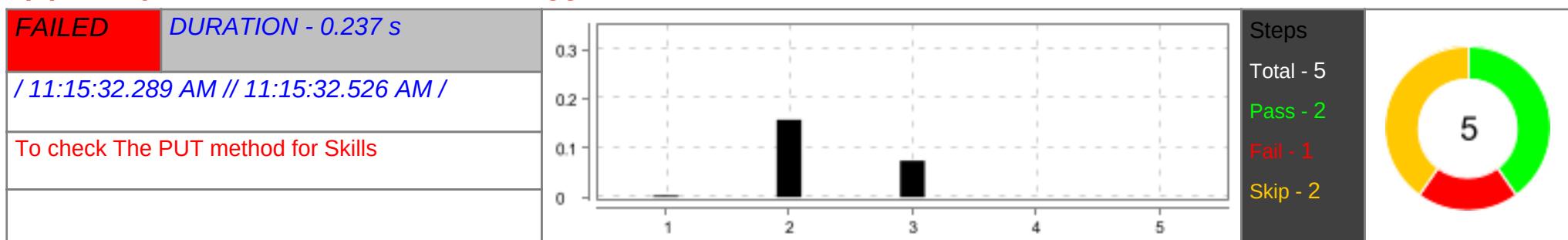
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.002 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 2 with valid JSON Schema	PASSED	0.160 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 2 row	PASSED	0.077 s
4	And JSON schema is valid for "PUT" in Skills	PASSED	0.001 s
5	And skills check the Database	PASSED	0.003 s

(S)- Verify PUT for a Skill when user logged with UserName with Password



#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.002 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 3 with valid JSON Schema	PASSED	0.163 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 3 row	PASSED	0.089 s
4	And JSON schema is valid for "PUT" in Skills	PASSED	0.001 s
5	And skills check the Database	PASSED	0.002 s

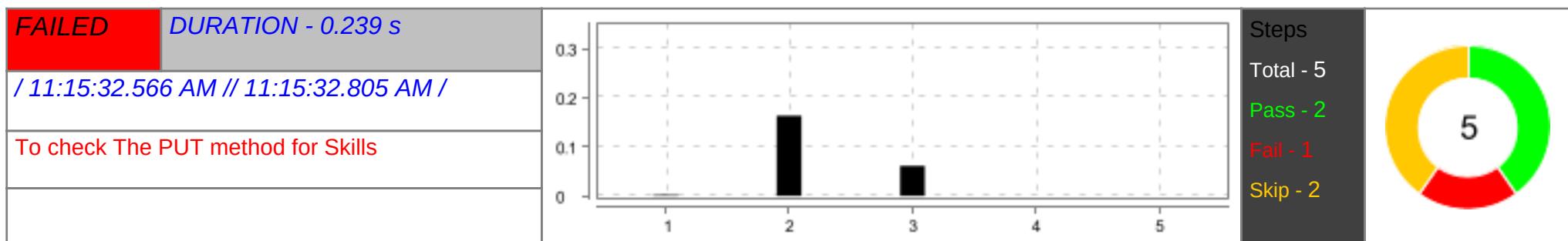
(S)- Verify PUT for a Skill when user logged with UserName with Password



#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 4 with valid JSON Schema	PASSED	0.157 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 4 row com.fasterxml.jackson.core.JsonParseException: Unexpected character ('<' (code 60)): expected a valid value (JSON String, Number, Array, Object or token 'null', 'true' or 'false') at [Source: (String)"<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN"> <title>404 Not Found</title> <h1>Not Found</h1> <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p> "; line: 1, column: 2] at com.fasterxml.jackson.core.JsonParser._constructError(JsonParser.java:2391) at com.fasterxml.jackson.core.base.ParserMinimalBase._reportError(ParserMinimalBase.java:735)	FAILED	0.074 s

#	Step / Hook Details	Status	Duration
	<pre> at com.fasterxml.jackson.core.base.ParserMinimalBase._reportUnexpectedChar(ParserMinimalBase.java:659) at com.fasterxml.jackson.core.json.ReaderBasedJsonParser._handleOddValue(ReaderBasedJsonParser.java:2005) at com.fasterxml.jackson.core.json.ReaderBasedJsonParser.nextToken(ReaderBasedJsonParser.java:802) at com.fasterxml.jackson.databind.ObjectMapper._initForReading(ObjectMapper.java:4762) at com.fasterxml.jackson.databind.ObjectMapper._readMapAndClose(ObjectMapper.java:4668) at com.fasterxml.jackson.databind.ObjectMapper.readValue(ObjectMapper.java:3630) at com.fasterxml.jackson.databind.ObjectMapper.readValue(ObjectMapper.java:3598) at step_definition.SkillMaster.extractResponse(SkillMaster.java:305) at step_definition.SkillMaster.user_validates_the_statuscode_and_statusmessage_from(SkillMaster.java:291) at ?.skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 4 row(file:///C:/@Ruchira/Workspace/ LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_PUT.feature:7) * Not displayable characters are replaced by '?'. </pre>		
4	And JSON schema is valid for "PUT" in Skills	SKIPPED	0.000 s
5	And skills check the Database	SKIPPED	0.000 s

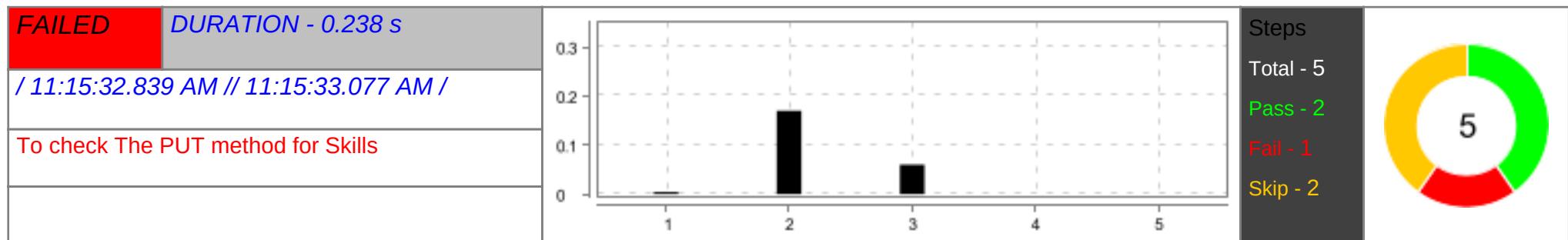
(S)- Verify PUT for a Skill when user logged with UserName with Password



#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.002 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 5 with valid JSON Schema	PASSED	0.163 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 5 row <pre> java.lang.AssertionError: expected [400] but found [201] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.SkillMaster.user_validates_the_statuscode_and_statusmessage_from(SkillMaster.java:289) at ?.skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 5 row(file:///C:/@Ruchira/Workspace/ </pre>	FAILED	0.061 s

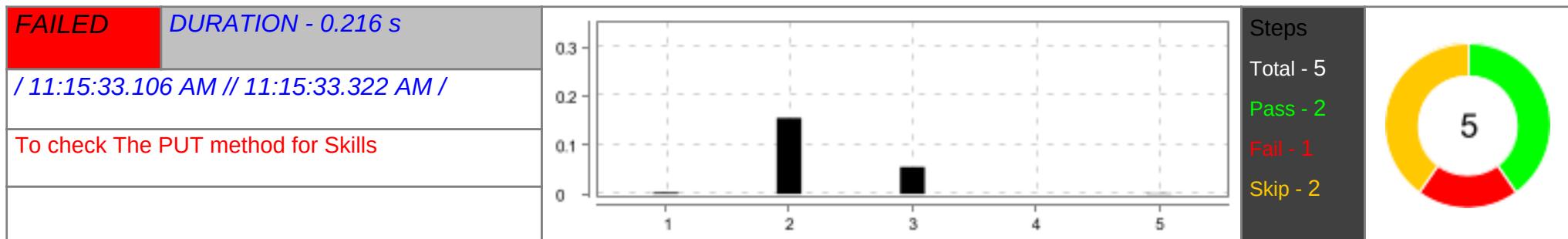
#	Step / Hook Details	Status	Duration
	LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_PUT.feature:7 * Not displayable characters are replaced by '?'.		
4	And JSON schema is valid for "PUT" in Skills	SKIPPED	0.000 s
5	And skills check the Database	SKIPPED	0.000 s

(S)- Verify PUT for a Skill when user logged with UserName with Password



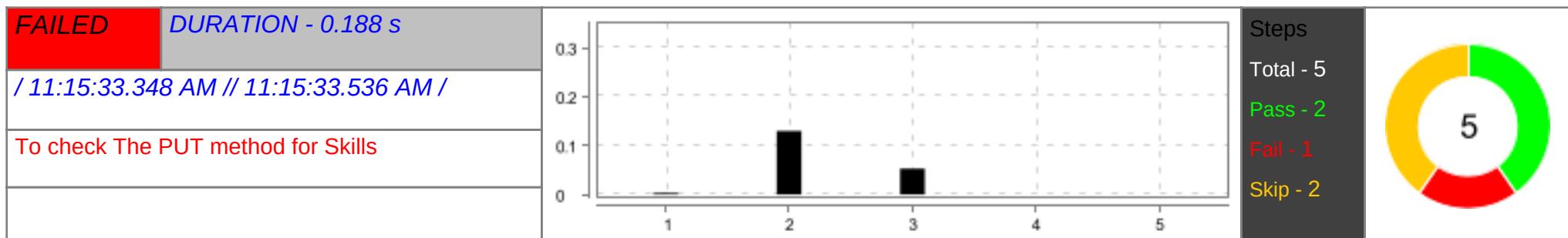
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.004 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 6 with valid JSON Schema	PASSED	0.170 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 6 row java.lang.AssertionError: expected [404] but found [400] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.SkillMaster.user_validates_the_statuscode_and_statusmessage_from(SkillMaster.java:289) at ?.skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 6 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_PUT.feature:7) * Not displayable characters are replaced by '?'.	FAILED	0.060 s
4	And JSON schema is valid for "PUT" in Skills	SKIPPED	0.000 s
5	And skills check the Database	SKIPPED	0.000 s

(S)- Verify PUT for a Skill when user logged with UserName with Password



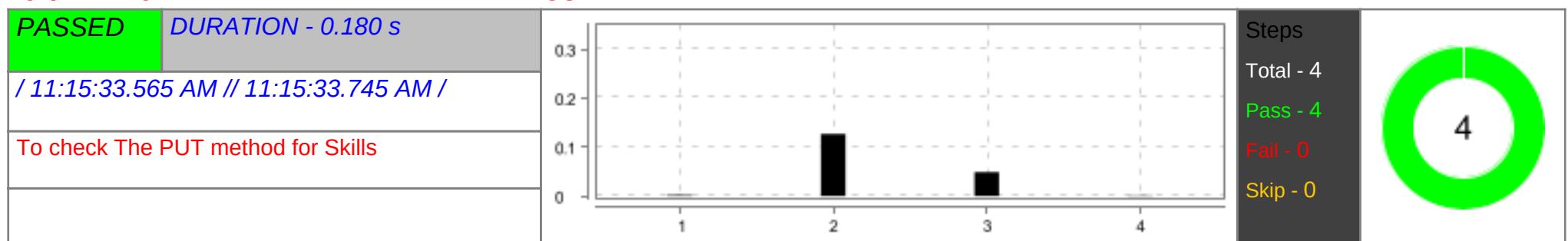
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 7 with valid JSON Schema	PASSED	0.155 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 7 row java.lang.AssertionError: expected [400] but found [201] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.SkillMaster.user_validates_the_statuscode_and_statusmessage_from(SkillMaster.java:289) at ?.skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 7 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_PUT.feature:7) * Not displayable characters are replaced by '?'.	FAILED	0.055 s
4	And JSON schema is valid for "PUT" in Skills	SKIPPED	0.000 s
5	And skills check the Database	SKIPPED	0.001 s

(S)- Verify PUT for a Skill when user logged with UserName with Password



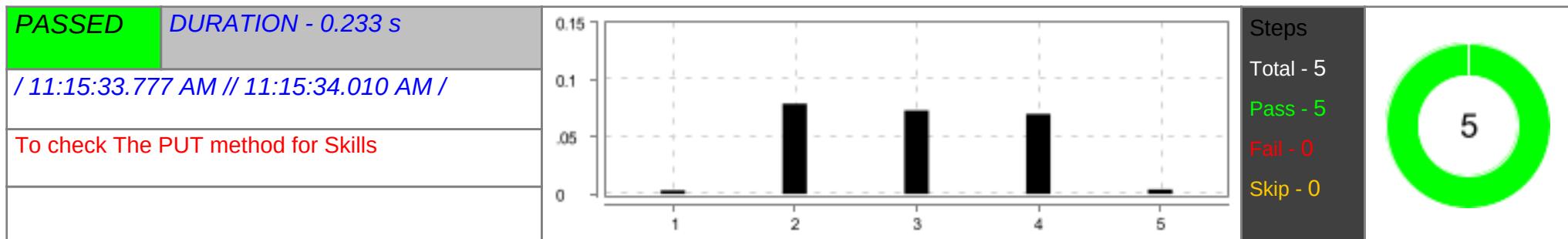
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends PUT request on id and request body in skills from "Skills_PUT" and 8 with valid JSON Schema	PASSED	0.130 s
3	Then skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 8 row java.lang.AssertionError: expected [400] but found [404] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.SkillMaster.user_validates_the_statuscode_and_statusmessage_from(SkillMaster.java:289) at ?.skills User validates the StatusCode and StatusMessage from "Skills_PUT" sheet and 8 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/skillMaster.feature/SkillMaster_PUT.feature:7) * Not displayable characters are replaced by '?'. And JSON schema is valid for "PUT" in Skills And skills check the Database	FAILED	0.053 s
4	And JSON schema is valid for "PUT" in Skills	SKIPPED	0.000 s
5	And skills check the Database	SKIPPED	0.000 s

(S)- Verify PUT for a Skill when user logged with UserName with Password



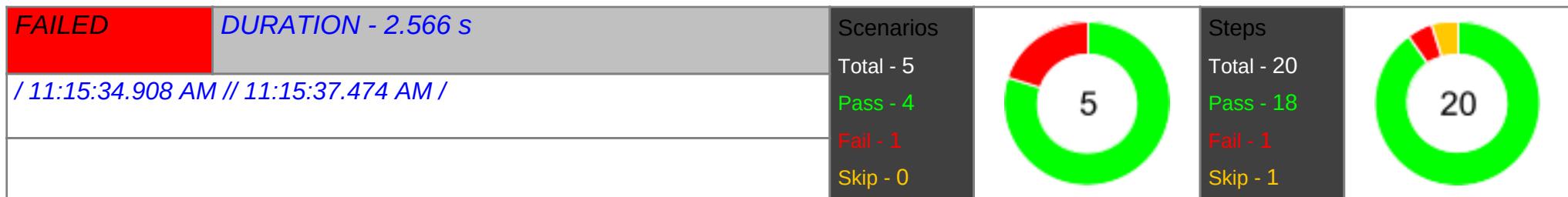
#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.002 s
2	When skills User sends PUT request on id and request body in skills from "Invalid_content_PUT" and 0 with valid JSON Schema	PASSED	0.127 s
3	Then skills User validates the StatusCode and StatusMessage from "Invalid_content_PUT" sheet and 0 row	PASSED	0.049 s
4	And skills check the Database	PASSED	0.001 s

(S)- Verify PUT fails for a Skill with improper format when user logged with UserName with Password

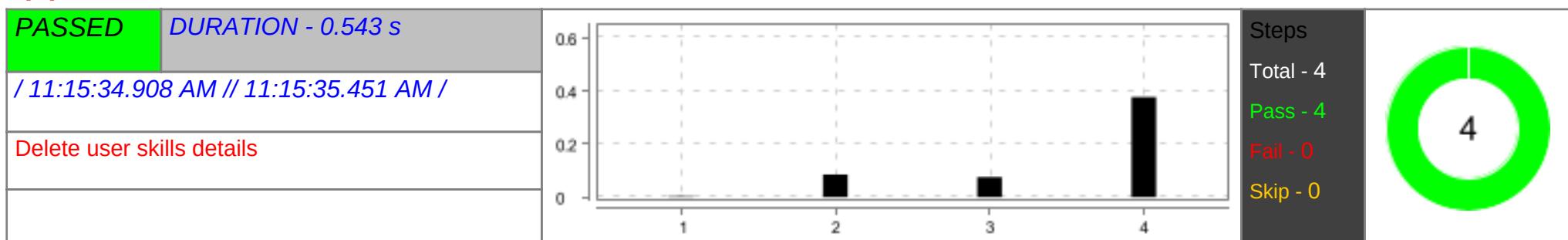


#	Step / Hook Details	Status	Duration
1	Given Skills User is on Endpoint: url/Skills with valid username and password	PASSED	0.003 s
2	When skills User sends PUT request on id and request body in text format skills from "Text_PUT" and 0	PASSED	0.079 s
3	Then skills User validates the StatusCode and StatusMessage from "Text_PUT" sheet and 0 row	PASSED	0.073 s
4	And skills User should receive the skill in JSON body from "Text_PUT" and 0	PASSED	0.070 s
5	And skills check the Database	PASSED	0.004 s

(F)- Delete user skills details

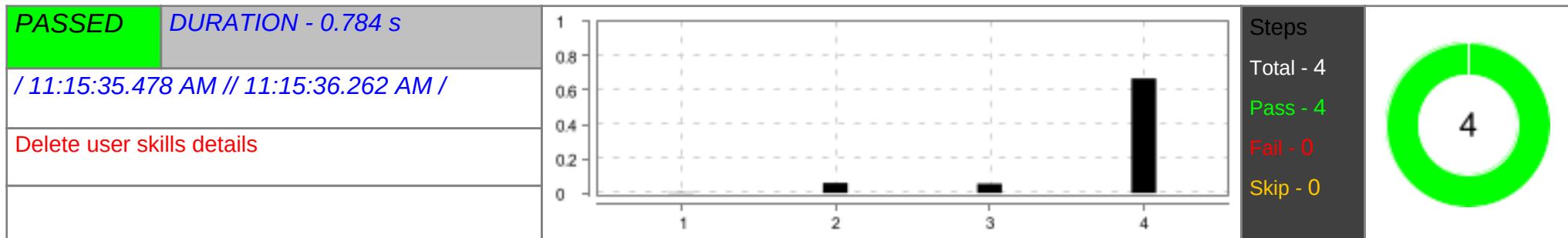


(S)- Validate the delete method for UserSkills



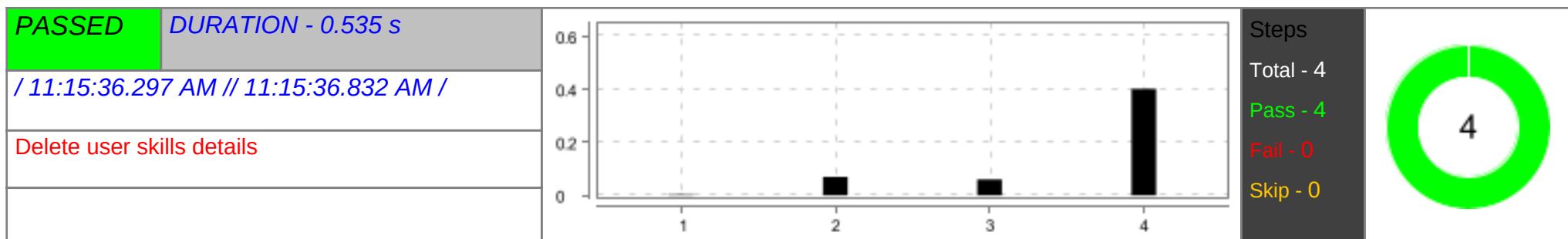
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends request id ON DELETE Method from "InputData_UserSkills_POST" and 0	PASSED	0.085 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_DELETE" sheet and 0 row	PASSED	0.075 s
4	And userSkills users check the Database to validate deletion from "InputData_UserSkills_POST" sheet and 0 row	PASSED	0.377 s

(S)- Validate the delete method for UserSkills



#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends request id ON DELETE Method from "InputData_UserSkills_DELETE" and 1	PASSED	0.058 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_DELETE" sheet and 1 row	PASSED	0.054 s
4	And userSkills users check the Database to validate deletion from "InputData_UserSkills_DELETE" sheet and 1 row	PASSED	0.668 s

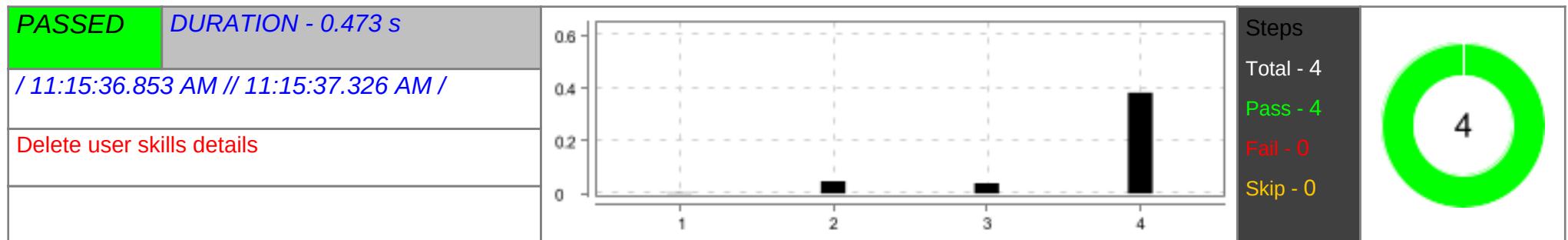
(S)- Validate the delete method for UserSkills



#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends request id ON DELETE Method from "InputData_UserSkills_DELETE" and 2	PASSED	0.070 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_DELETE" sheet and 2 row	PASSED	0.060 s

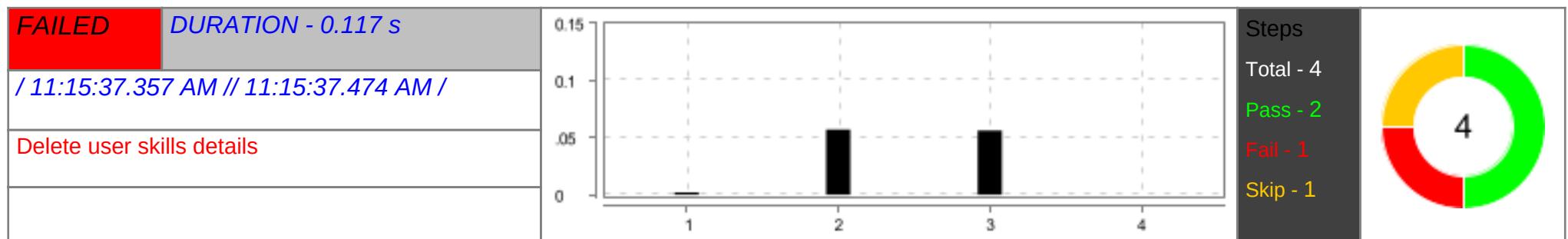
#	Step / Hook Details	Status	Duration
4	And userSkills users check the Database to validate deletion from "InputData_UserSkills_DELETE" sheet and 2 row	PASSED	0.402 s

(S)- Validate the delete method for UserSkills



#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends request id ON DELETE Method from "InputData_UserSkills_DELETE" and 3	PASSED	0.047 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_DELETE" sheet and 3 row	PASSED	0.039 s
4	And userSkills users check the Database to validate deletion from "InputData_UserSkills_DELETE" sheet and 3 row	PASSED	0.383 s

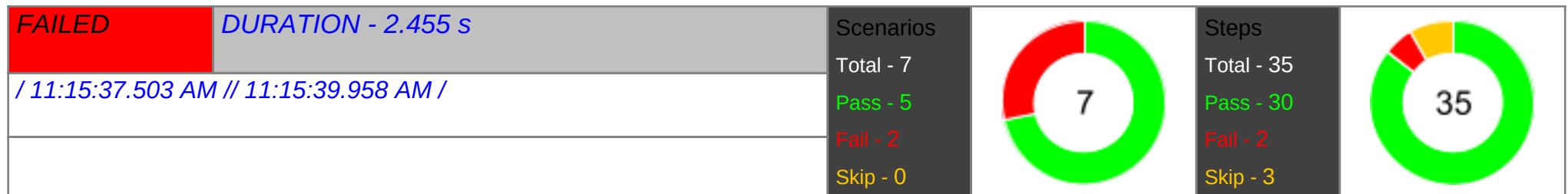
(S)- Validate the delete method for UserSkills



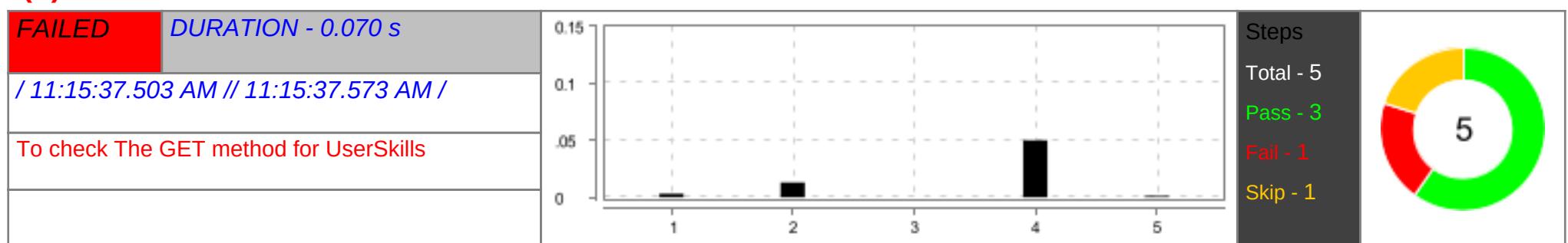
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends request id ON DELETE Method from "InputData_UserSkills_DELETE" and 4	PASSED	0.057 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_DELETE" sheet and 4 row java.lang.AssertionError: expected object to not be null at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.assertNotNull(Assert.java:686)	FAILED	0.056 s

#	Step / Hook Details	Status	Duration
	at org.testng.Assert.assertNotNull Assert.java:671 at step_definition.UserSkills.userSkills_user_receives_status_code_with UserSkills.java:258 at ?.userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_DELETE" sheet and 4 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkills.feature/UserSkills_DELETE.feature:7) * Not displayable characters are replaced by '?'.		
4	And userSkills users check the Database to validate deletion from "InputData_UserSkills_DELETE" sheet and 4 row	SKIPPED	0.000 s

(F)- To check The GET method for UserSkills



(S)- Validate GET all UserSkills

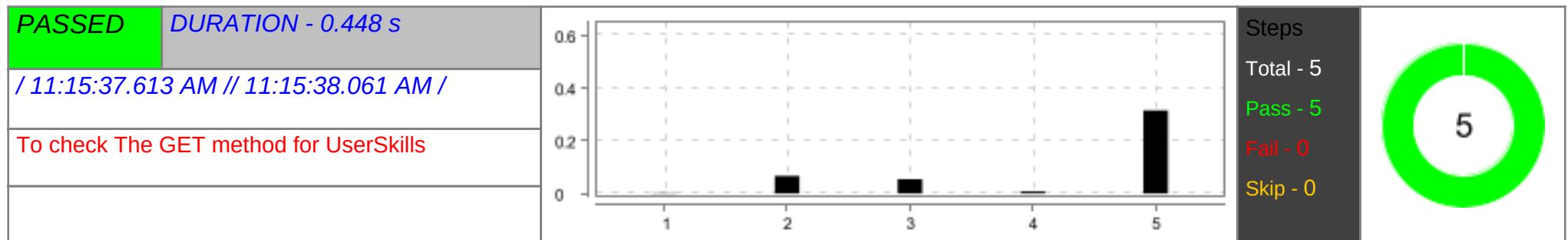


#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends GET request	PASSED	0.013 s
3	Then userSkills User validates StatusCode	PASSED	0.000 s
4	And userSkills JSON schema is valid	FAILED	0.050 s
	java.lang.AssertionError: 1 expectation failed. Response body doesn't match expectation. Expected: The content to match the given JSON schema. error: instance type (string) does not match any allowed primitive type (allowed: ["integer"]) level: "error" schema: {"loadingURI":"file:/C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/target/test-classes/JSON_Schemas/user_skill_schema_GET_all.json#","pointer":"/properties/UserSkillMapping/items/0/properties/Skill_id"} instance: {"pointer":"/UserSkillMapping/0/Skill_id"} domain: "validation" keyword: "type" found: "string" expected: ["integer"] error: instance type (string) does not match any allowed primitive type (allowed: ["integer"]) level: "error" schema: {"loadingURI":"file:/C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/target/test-classes/JSON_Schemas/user_skill_schema_GET_all.json#","pointer":"/properties/UserSkillMapping/items/0/properties/Skill_id"} instance: {"pointer":"/UserSkillMapping/0/Skill_id"} domain: "validation" keyword: "type" found: "string" expected: ["integer"]		

#	Step / Hook Details	Status	Duration
	<pre> JSON_Schemas/user_skill_schema_GET_all.json#,"pointer":"/properties/UserSkillMapping/items/0/properties/months_of_exp"} instance: {"pointer":"/UserSkillMapping/0/months_of_exp"} domain: "validation" keyword: "type" found: "string" expected: ["integer"] Actual: { "UserSkillMapping": [{ "user_skill_id": "US01", "user_id": "U01", "Skill_id": "2", "months_of_exp": "24" }, { "user_skill_id": "US02", "user_id": "U02", "Skill_id": "1", "months_of_exp": "36" }, { "user_skill_id": "US03", "user_id": "U03", "Skill_id": "2", "months_of_exp": "12" }, { "user_skill_id": "US05", "user_id": "U05", "Skill_id": "5", "months_of_exp": "12" }, { "user_skill_id": "US04", "user_id": "U04", "Skill_id": "6", "months_of_exp": "36" }, { "user_skill_id": "US06", "user_id": "U06", "Skill_id": "5", "months_of_exp": "12" }, { "user_skill_id": "US07", "user_id": "U07", "Skill_id": "2", "months_of_exp": "12" }, { "user_skill_id": "US09", "user_id": "U09", "Skill_id": "3", "months_of_exp": "36" }, { "user_skill_id": "US10", "user_id": "U10", "Skill_id": "2", "months_of_exp": "10" }, { "user_skill_id": "US12", "user_id": "U08", "Skill_id": "4", "months_of_exp": "12" }, { "user_skill_id": "US08", "user_id": "U08", "Skill_id": "1", "months_of_exp": "30" }] } at java.base/jdk.internal.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method) at java.base/jdk.internal.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62) at java.base/jdk.internal.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.java:45) at java.base/java.lang.reflect.Constructor.newInstance(Constructor.java:490) at org.codehaus.groovy.reflection.CachedConstructor.invoke(CachedConstructor.java:83) at org.codehaus.groovy.reflection.CachedConstructor.doConstructorInvoke(CachedConstructor.java:77) at org.codehaus.groovy.runtime.callsite.ConstructorSite\$ConstructorSiteNoUnwrap.callConstructor(ConstructorSite.java:84) at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCallConstructor(CallSiteArray.java:59) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callConstructor(AbstractCallSite.java:238) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callConstructor(AbstractCallSite.java:250) at io.restassured.internal.ResponseSpecificationImpl\$HamcrestAssertionClosure.validate(ResponseSpecificationImpl.groovy:494) at io.restassured.internal.ResponseSpecificationImpl\$HamcrestAssertionClosure\$validate\$1.call(Unknown Source) at io.restassured.internal.ResponseSpecificationImpl.validateResponseIfRequired(ResponseSpecificationImpl.groovy:656) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at java.base/jdk.internal.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.base/java.lang.reflect.Method.invoke(Method.java:566) at org.codehaus.groovy.runtime.callsite.PogoMetaMethodSite\$PogoCachedMethodSiteNoUnwrapNoCoerce.invoke(PogoMetaMethodSite.java:210) at org.codehaus.groovy.runtime.callsite.PogoMetaMethodSite.callCurrent(PogoMetaMethodSite.java:59) at org.codehaus.groovy.runtime.callsite.AbstractCallSite.callCurrent(AbstractCallSite.java:169) at io.restassured.internal.ResponseSpecificationImpl.content(ResponseSpecificationImpl.groovy:97) at io.restassured.specification.ResponseSpecification\$content\$0.callCurrent(Unknown Source) at org.codehaus.groovy.runtime.callsite.CallSiteArray.defaultCallCurrent(CallSiteArray.java:51) at io.restassured.specification.ResponseSpecification\$content\$0.callCurrent(Unknown Source) at io.restassured.internal.ResponseSpecificationImpl.body(ResponseSpecificationImpl.groovy:256) at io.restassured.internal.ValidatableResponseOptionsImpl.body(ValidatableResponseOptionsImpl.java:268) at util.JSON_Schema_Validation.cls_JSON_SchemaValidation(JSON_Schema_Validation.java:29) at step_definition.UserSkills.userSkills_json_schema_is_valid(UserSkills.java:193) at ?.userSkills JSON schema is valid(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkills.feature/ UserSkills_GET.feature:7) * Not displayable characters are replaced by '?'. </pre>		

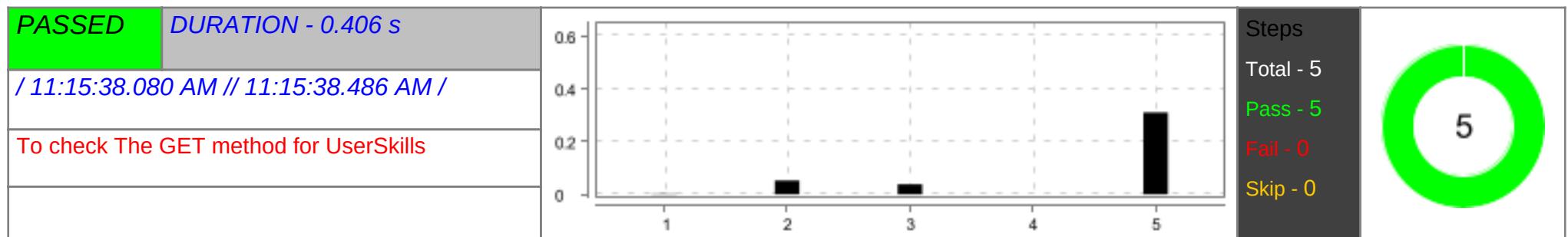
#	Step / Hook Details	Status	Duration
5	And check the Database for userSkills	SKIPPED	0.001 s

(S)- Validate UserSkills Api with specific ID



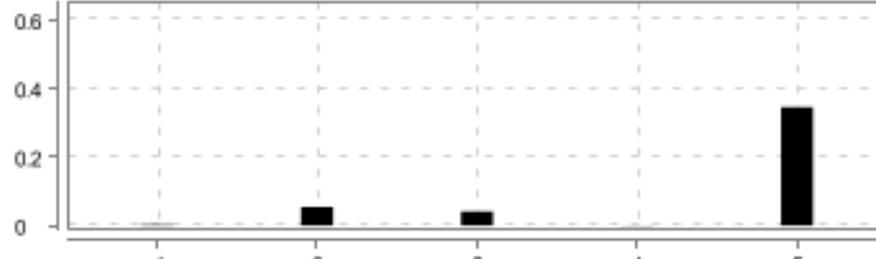
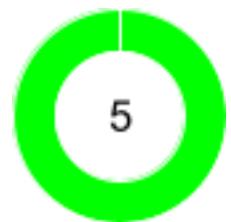
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends GET request on id from "InputData_UserSkills_GET" and 0	PASSED	0.067 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_GET" sheet and 0 row	PASSED	0.054 s
4	And userSkills JSON schema is valid for "GET"	PASSED	0.008 s
5	And check the Database	PASSED	0.315 s

(S)- Validate UserSkills Api with specific ID



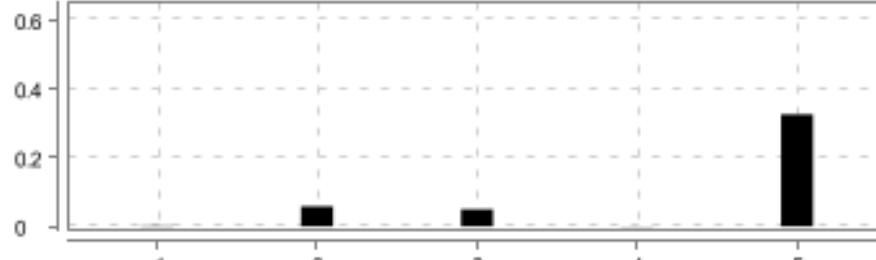
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends GET request on id from "InputData_UserSkills_GET" and 1	PASSED	0.052 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_GET" sheet and 1 row	PASSED	0.038 s
4	And userSkills JSON schema is valid for "GET"	PASSED	0.000 s
5	And check the Database	PASSED	0.311 s

(S)- Validate UserSkills Api with specific ID

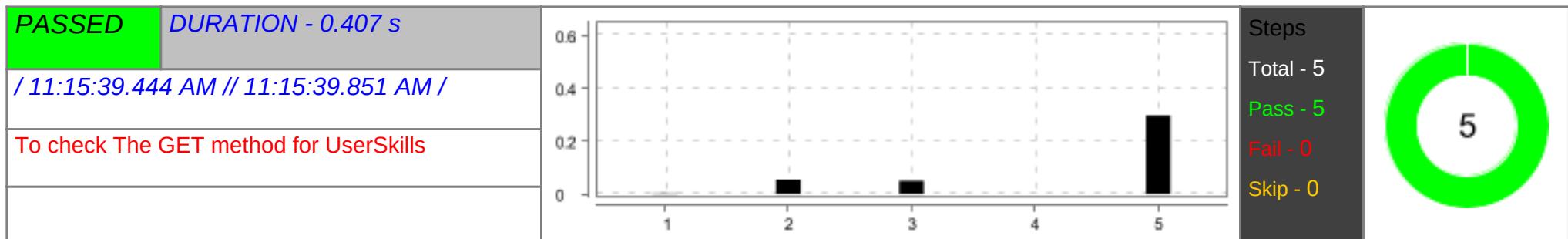
PASSED	DURATION - 0.446 s		Steps Total - 5 Pass - 5 Fail - 0 Skip - 0	
/ 11:15:38.507 AM // 11:15:38.953 AM /				
To check The GET method for UserSkills				

#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends GET request on id from "InputData_UserSkills_GET" and 2	PASSED	0.054 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_GET" sheet and 2 row	PASSED	0.041 s
4	And userSkills JSON schema is valid for "GET"	PASSED	0.001 s
5	And check the Database	PASSED	0.346 s

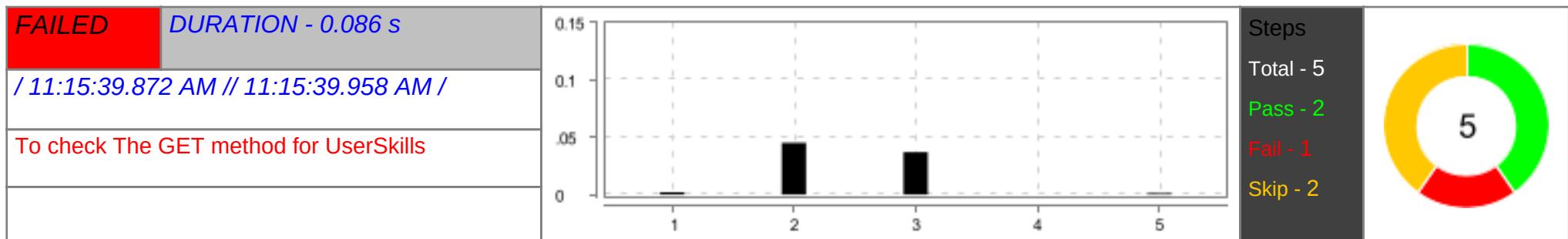
(S)- Validate UserSkills Api with specific ID

PASSED	DURATION - 0.439 s		Steps Total - 5 Pass - 5 Fail - 0 Skip - 0	
/ 11:15:38.985 AM // 11:15:39.424 AM /				
To check The GET method for UserSkills				

#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends GET request on id from "InputData_UserSkills_GET" and 3	PASSED	0.058 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_GET" sheet and 3 row	PASSED	0.050 s
4	And userSkills JSON schema is valid for "GET"	PASSED	0.001 s
5	And check the Database	PASSED	0.326 s

(S)- Validate UserSkills Api with specific ID

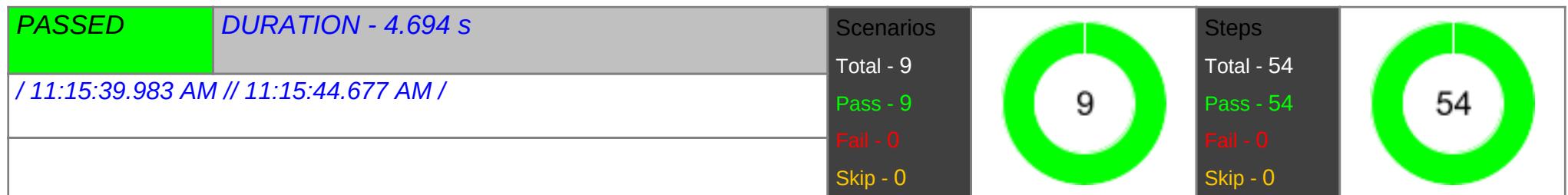
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends GET request on id from "InputData_UserSkills_GET" and 4	PASSED	0.054 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_GET" sheet and 4 row	PASSED	0.051 s
4	And userSkills JSON schema is valid for "GET"	PASSED	0.000 s
5	And check the Database	PASSED	0.297 s

(S)- Validate UserSkills Api with specific ID

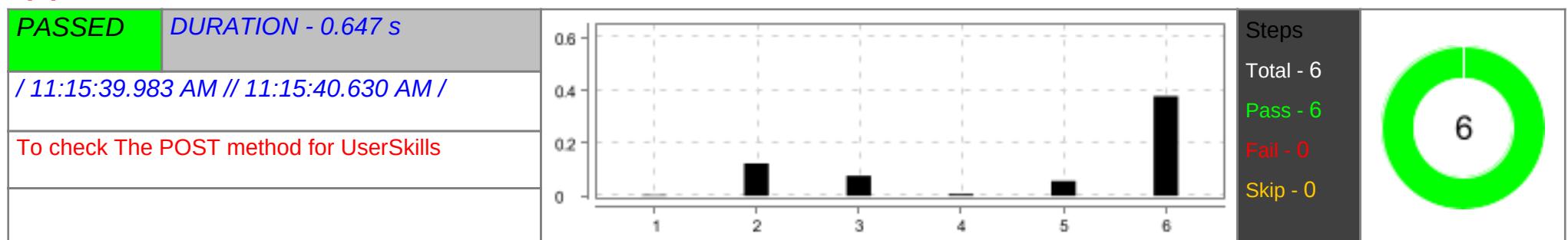
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends GET request on id from "InputData_UserSkills_GET" and 5	PASSED	0.045 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_GET" sheet and 5 row java.lang.AssertionError: expected object to not be null at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.assertNotNull(Assert.java:686) at org.testng.Assert.assertNotNull(Assert.java:671) at step_definition.UserSkills.userSkills_user_receives_status_code_with(UserSkills.java:258) at ?.userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_GET" sheet and 5 row(file:///C:/@Ruchira/)	FAILED	0.037 s

#	Step / Hook Details	Status	Duration
	Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkills.feature/UserSkills_GET.feature:12 * Not displayable characters are replaced by '?'.		
4	And userSkills JSON schema is valid for "GET"	SKIPPED	0.000 s
5	And check the Database	SKIPPED	0.001 s

(F)- To check The POST method for UserSkills

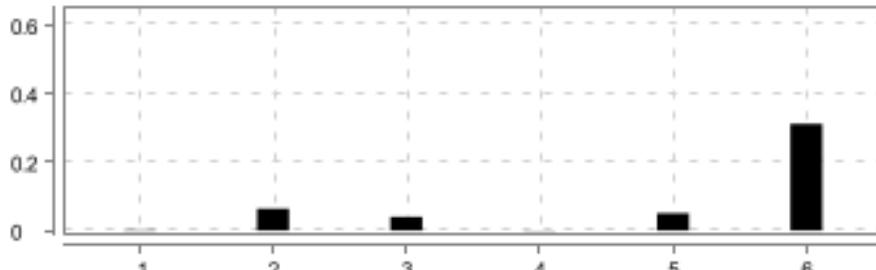


(S)- Validate POST all UserSkills



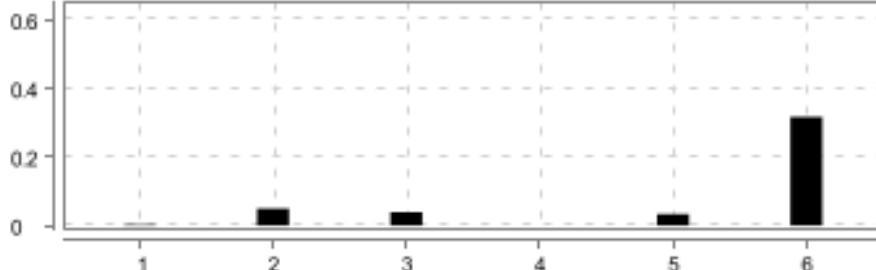
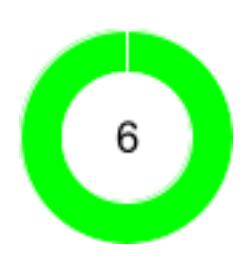
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 0 with valid JSON Schema	PASSED	0.123 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 0 row	PASSED	0.076 s
4	And userSkills JSON schema is valid for "POST"	PASSED	0.007 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id,months_of_exp from "InputData_UserSkills_POST" and 0	PASSED	0.056 s
6	And check the Database	PASSED	0.378 s

(S)- Validate POST all UserSkills

PASSED	DURATION - 0.471 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:40.652 AM // 11:15:41.123 AM /				
To check The POST method for UserSkills				

#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 1 with valid JSON Schema	PASSED	0.064 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 1 row	PASSED	0.040 s
4	And userSkills JSON schema is valid for "POST"	PASSED	0.001 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_POST" and 1	PASSED	0.051 s
6	And check the Database	PASSED	0.312 s

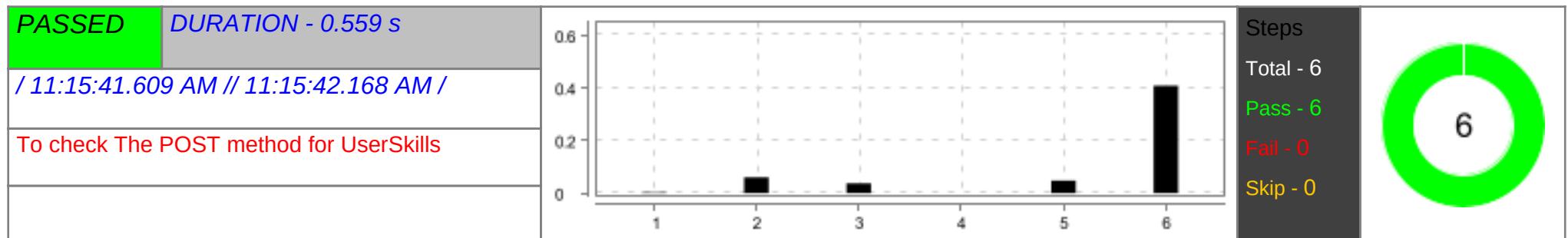
(S)- Validate POST all UserSkills

PASSED	DURATION - 0.444 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:41.144 AM // 11:15:41.588 AM /				
To check The POST method for UserSkills				

#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 2 with valid JSON Schema	PASSED	0.050 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 2 row	PASSED	0.039 s
4	And userSkills JSON schema is valid for "POST"	PASSED	0.000 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_POST" and 2	PASSED	0.033 s

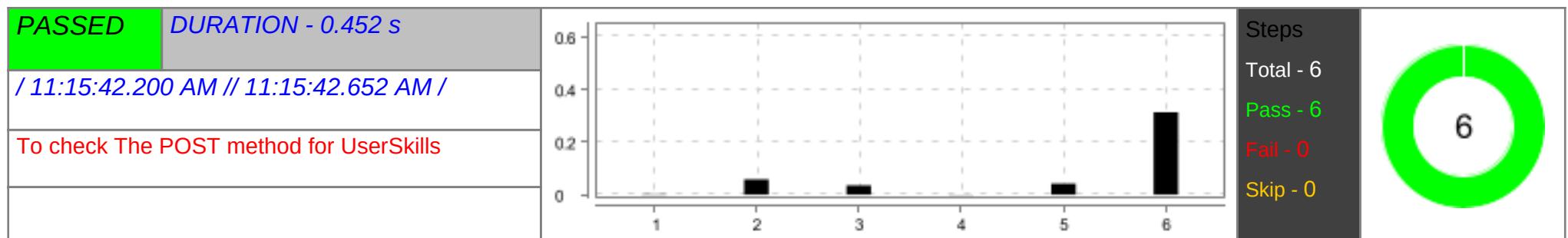
#	Step / Hook Details	Status	Duration
6	And check the Database	PASSED	0.317 s

(S)- Validate POST all UserSkills



#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 3 with valid JSON Schema	PASSED	0.060 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 3 row	PASSED	0.037 s
4	And userSkills JSON schema is valid for "POST"	PASSED	0.000 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_POST" and 3	PASSED	0.048 s
6	And check the Database	PASSED	0.409 s

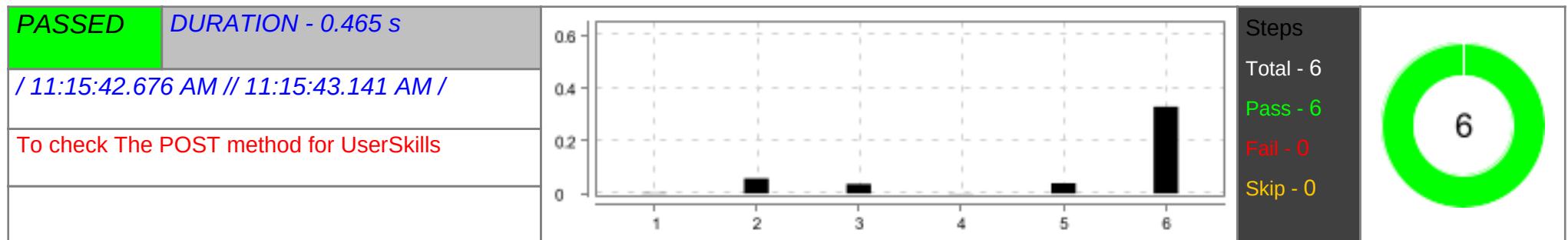
(S)- Validate POST all UserSkills



#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 4 with valid JSON Schema	PASSED	0.058 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 4 row	PASSED	0.035 s

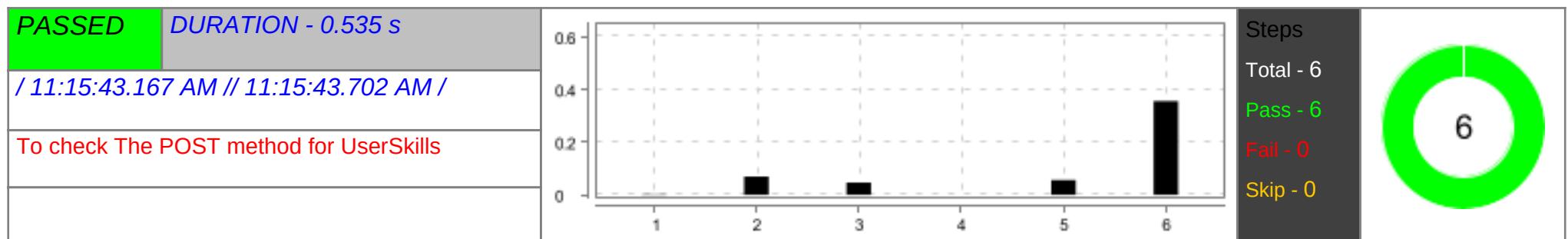
#	Step / Hook Details	Status	Duration
4	And userSkills JSON schema is valid for "POST"	PASSED	0.001 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_POST" and 4	PASSED	0.042 s
6	And check the Database	PASSED	0.314 s

(S)- Validate POST all UserSkills



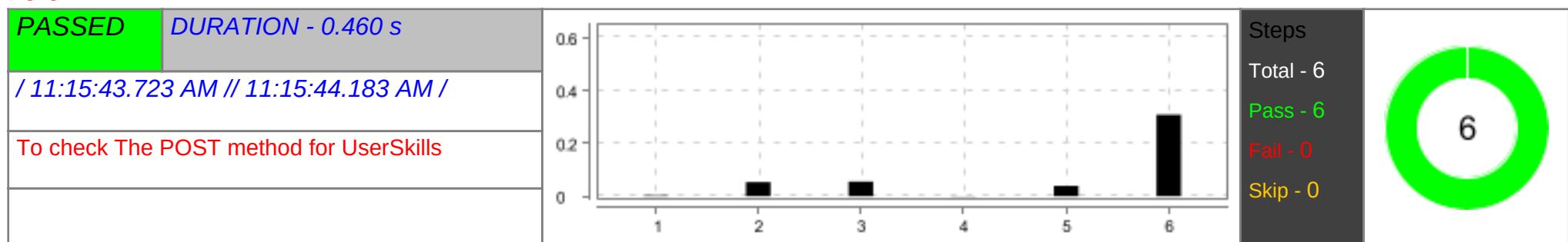
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 5 with valid JSON Schema	PASSED	0.057 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 5 row	PASSED	0.036 s
4	And userSkills JSON schema is valid for "POST"	PASSED	0.001 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_POST" and 5	PASSED	0.039 s
6	And check the Database	PASSED	0.330 s

(S)- Validate POST all UserSkills

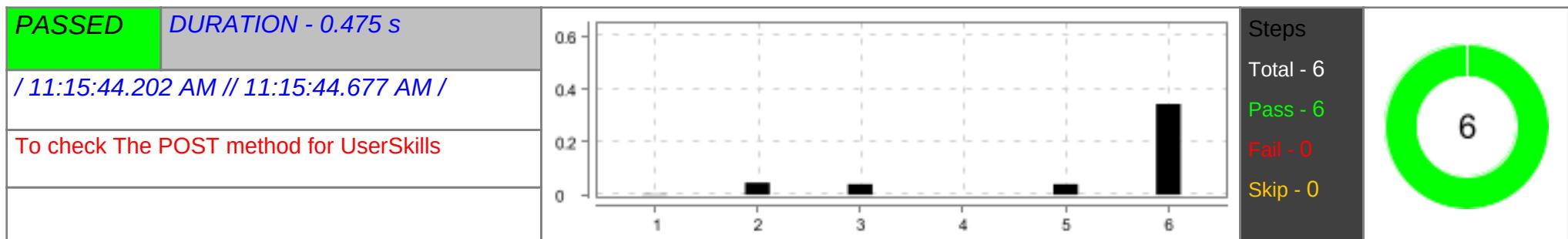


#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 6 with valid JSON Schema	PASSED	0.070 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 6 row	PASSED	0.047 s
4	And userSkills JSON schema is valid for "POST"	PASSED	0.000 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_POST" and 6	PASSED	0.056 s
6	And check the Database	PASSED	0.356 s

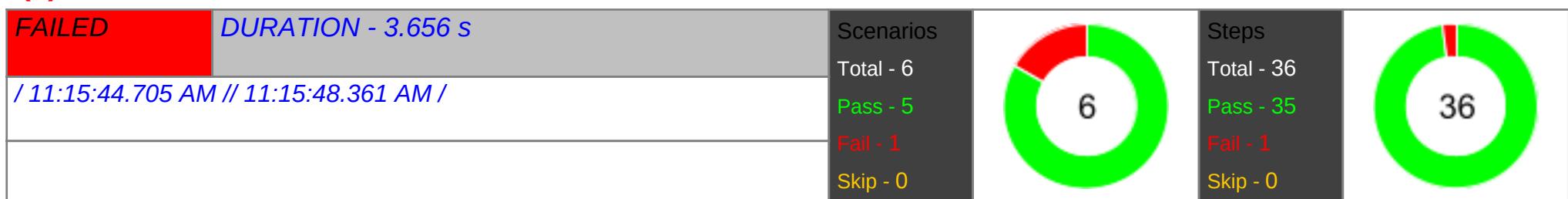
(S)- Validate POST all UserSkills



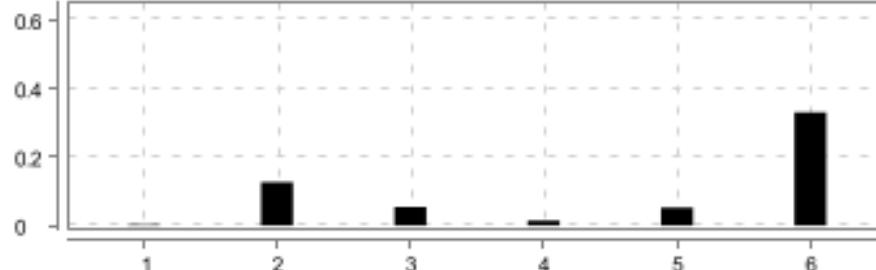
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 7 with valid JSON Schema	PASSED	0.053 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 7 row	PASSED	0.055 s
4	And userSkills JSON schema is valid for "POST"	PASSED	0.001 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_POST" and 7	PASSED	0.039 s
6	And check the Database	PASSED	0.309 s

(S)- Validate POST all UserSkills

#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends POST request body from "InputData_UserSkills_POST" and 8 with valid JSON Schema	PASSED	0.046 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_POST" sheet and 8 row	PASSED	0.040 s
4	And userSkills JSON schema is valid for "POST"	PASSED	0.000 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_POST" and 8	PASSED	0.040 s
6	And check the Database	PASSED	0.344 s

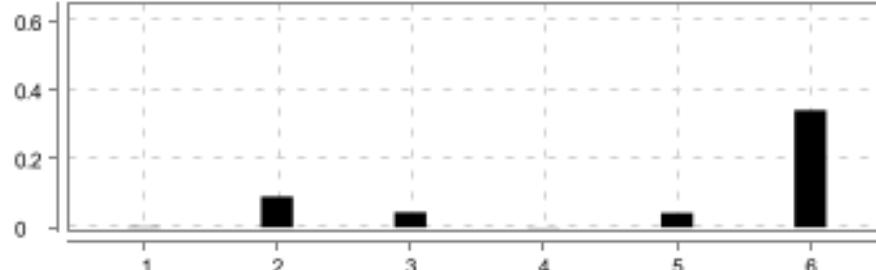
(F)- To check The PUT method for UserSkills

(S)- Validate PUT method for UserSkills

PASSED	DURATION - 0.584 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:44.705 AM // 11:15:45.289 AM /				
To check The PUT method for UserSkills				

#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User sends PUT request on id and request body from "InputData_UserSkills_PUT" and 0 with valid JSON schema	PASSED	0.127 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_PUT" sheet and 0 row	PASSED	0.054 s
4	And userSkills JSON schema is valid for "PUT"	PASSED	0.014 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_PUT" and 0	PASSED	0.052 s
6	And check the Database	PASSED	0.331 s

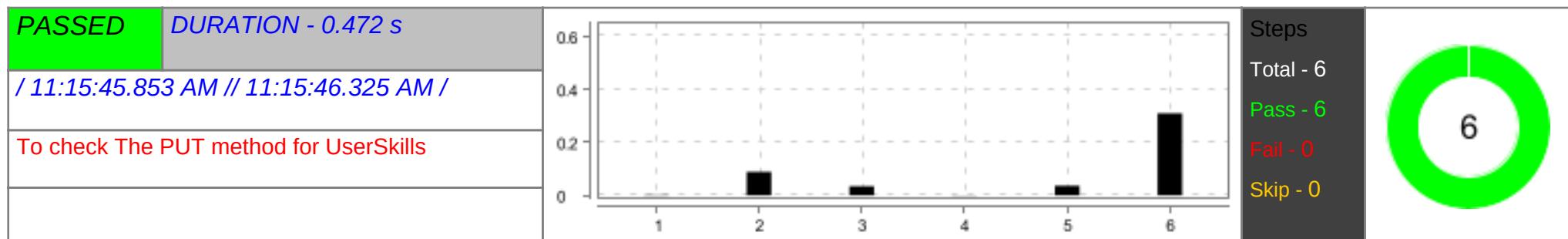
(S)- Validate PUT method for UserSkills

PASSED	DURATION - 0.521 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:15:45.312 AM // 11:15:45.833 AM /				
To check The PUT method for UserSkills				

#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends PUT request on id and request body from "InputData_UserSkills_PUT" and 1 with valid JSON schema	PASSED	0.090 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_PUT" sheet and 1 row	PASSED	0.044 s
4	And userSkills JSON schema is valid for "PUT"	PASSED	0.001 s

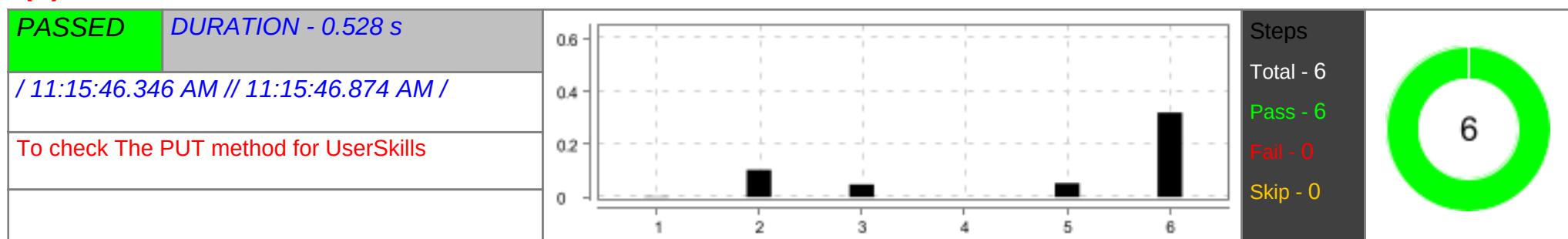
#	Step / Hook Details	Status	Duration
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_PUT" and 1	PASSED	0.042 s
6	And check the Database	PASSED	0.341 s

(S)- Validate PUT method for UserSkills



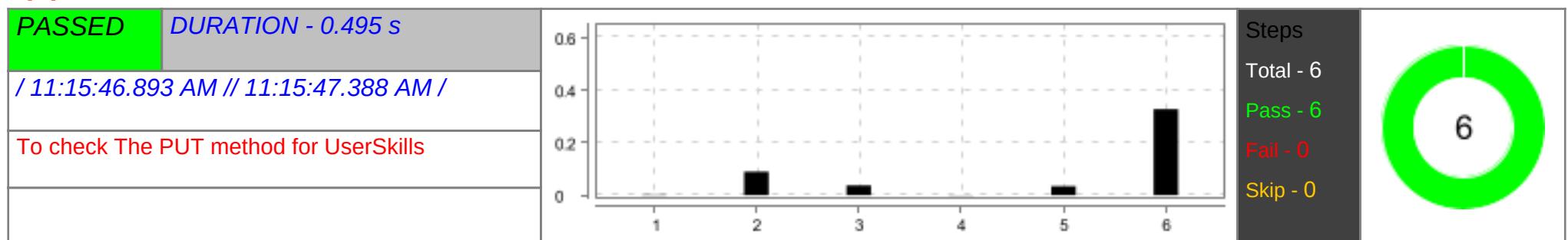
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends PUT request on id and request body from "InputData_UserSkills_PUT" and 2 with valid JSON schema	PASSED	0.089 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_PUT" sheet and 2 row	PASSED	0.034 s
4	And userSkills JSON schema is valid for "PUT"	PASSED	0.001 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_PUT" and 2	PASSED	0.036 s
6	And check the Database	PASSED	0.310 s

(S)- Validate PUT method for UserSkills



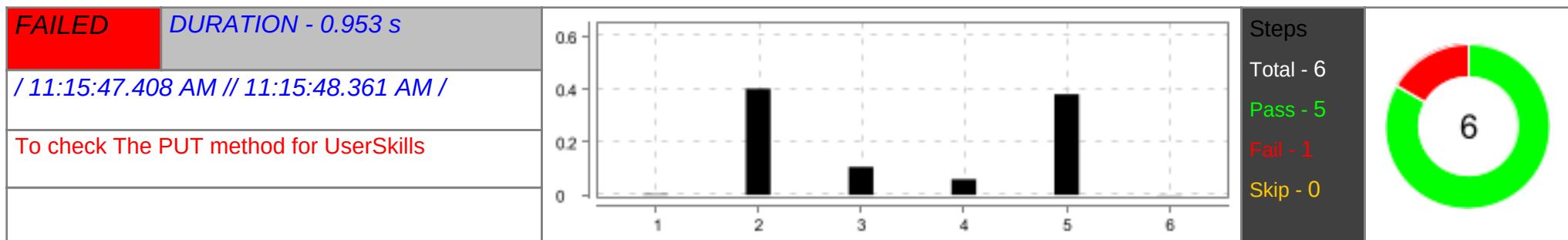
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends PUT request on id and request body from "InputData_UserSkills_PUT" and 3 with valid JSON schema	PASSED	0.103 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_PUT" sheet and 3 row	PASSED	0.048 s
4	And userSkills JSON schema is valid for "PUT"	PASSED	0.000 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_PUT" and 3	PASSED	0.052 s
6	And check the Database	PASSED	0.320 s

(S)- Validate PUT method for UserSkills



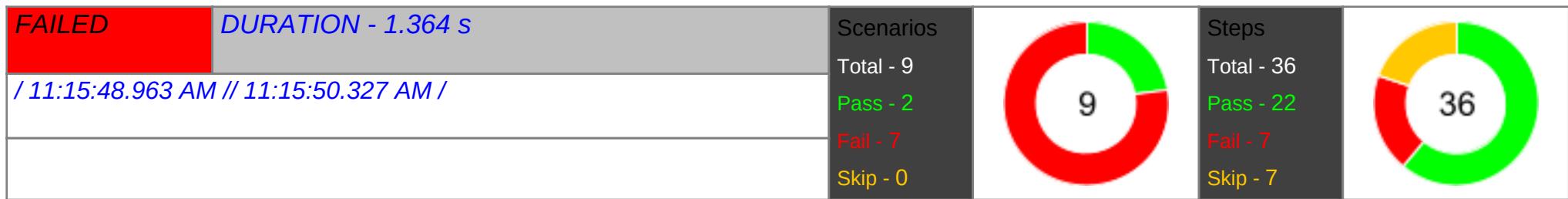
#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.002 s
2	When userSkills User sends PUT request on id and request body from "InputData_UserSkills_PUT" and 4 with valid JSON schema	PASSED	0.090 s
3	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_PUT" sheet and 4 row	PASSED	0.037 s
4	And userSkills JSON schema is valid for "PUT"	PASSED	0.001 s
5	And userSkills User should receive a list of users with skills in JSON body with the fields like user_skill_id,user_id,Skill_Id, months_of_exp from "InputData_UserSkills_PUT" and 4	PASSED	0.034 s
6	And check the Database	PASSED	0.328 s

(S)- Validate creation time & modification time for PUT method for UserSkills

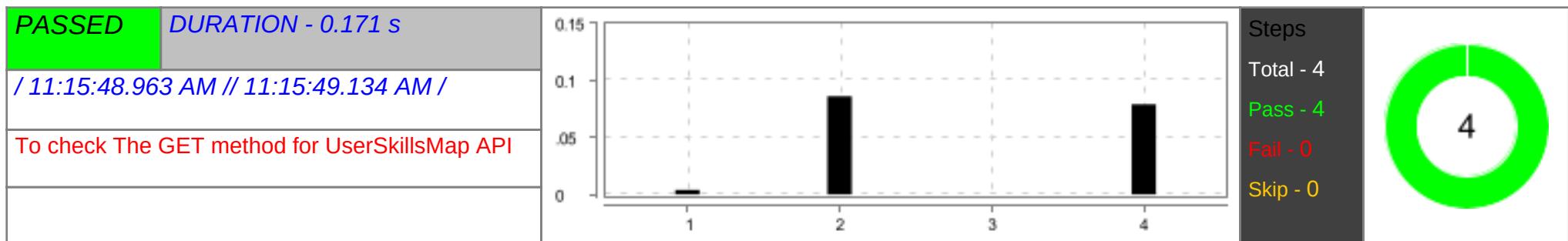


#	Step / Hook Details	Status	Duration
1	Given userSkills User is on Endpoint: url/UserSkills with valid username and password	PASSED	0.003 s
2	When userSkills User check the database for creation time & modification time of a user with specified UserSkillsId from "InputData_UserSkills_PUT" and 0	PASSED	0.402 s
3	And userSkills User sends PUT request on id and request body from "InputData_UserSkills_PUT" and 0 with valid JSON schema	PASSED	0.106 s
4	Then userSkills User validates StatusCode and StatusMessage from "InputData_UserSkills_PUT" sheet and 0 row	PASSED	0.059 s
5	And userSkills User check the database for new creation time & modification time of a user with specified UserSkillsId from "InputData_UserSkills_PUT" and 0	PASSED	0.381 s
6	And userSkills User check that the creation time is same but modification is different. java.lang.AssertionError: expected [2022-01-26 11:15:47.912459] but found [2022-01-26 11:15:44.826462] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:453) at org.testng.Assert.assertEquals(Assert.java:463) at step_definition.UserSkills.user_skills_user_check_that_the_creation_time_is_same_but_modification_is_different(UserSkills.java:485) at ?.userSkills User check that the creation time is same but modification is different.(file:///C:/@Ruchira/Workspace/ LMS_API_HACKATHON2122/src/test/resources/userSkills.feature/UserSkills_PUT.feature:26) * Not displayable characters are replaced by '?'.	FAILED	0.001 s

(F)- To check The GET method for UserSkillsMap API

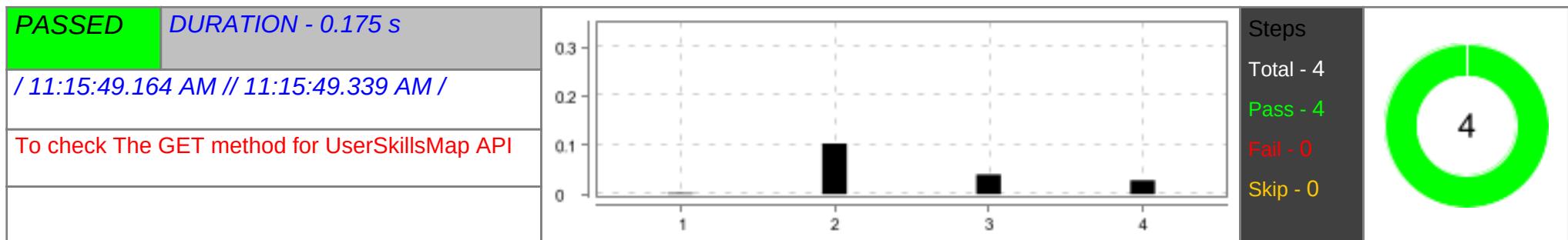


(S)- Validate GET all User-to-skill mappings



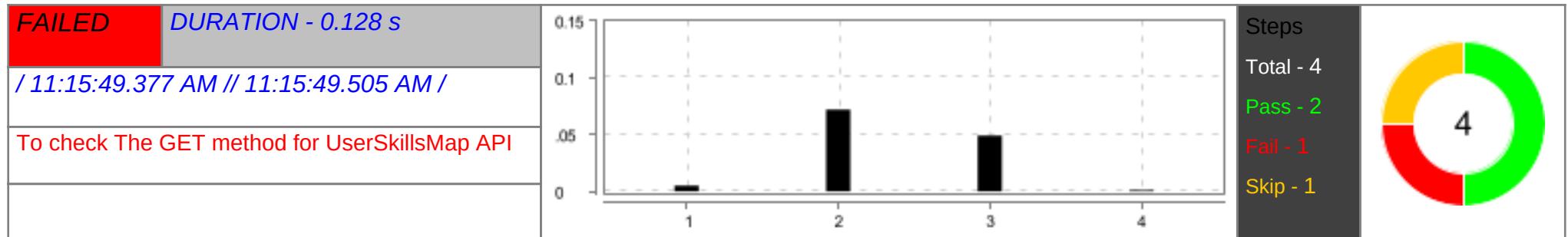
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid username and password	PASSED	0.004 s
2	When UserSkillsMap User sends GET request	PASSED	0.086 s
3	Then UserSkillsMap User validates StatusCode	PASSED	0.000 s
4	And UserSkillsMap JSON schema is valid	PASSED	0.079 s

(S)- Validate GET one particular User-to-skill mappings with specific user ID



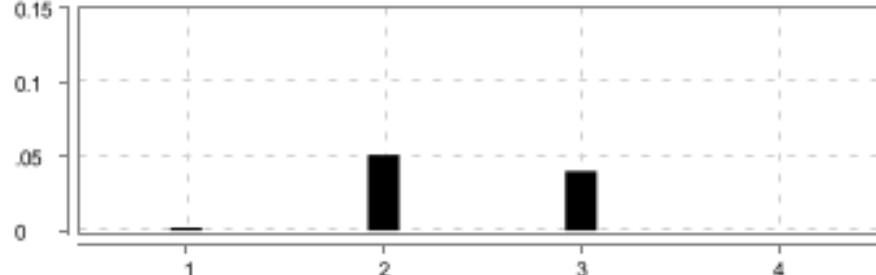
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid username and password	PASSED	0.002 s
2	When UserSkillsMap_user User sends GET request on user id from "InputData_UserSkillsMap_userGET" and 0	PASSED	0.103 s
3	Then UserSkillsMap_user User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_userGET" sheet and 0 row	PASSED	0.040 s
4	And UserSkillsMap JSON schema is valid	PASSED	0.028 s

(S)- Validate GET one particular User-to-skill mappings with specific user ID



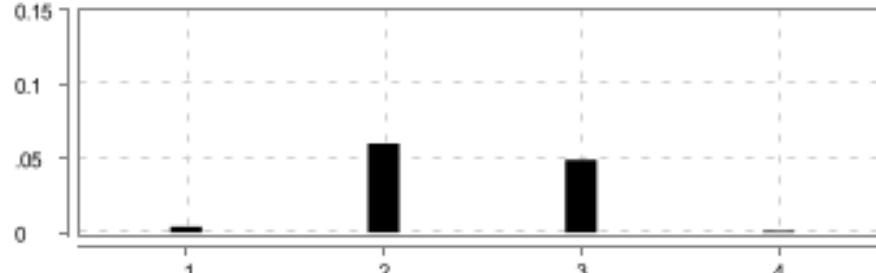
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid username and password	PASSED	0.005 s
2	When UserSkillsMap_user User sends GET request on user id from "InputData_UserSkillsMap_userGET" and 1	PASSED	0.072 s
3	Then UserSkillsMap_user User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_userGET" sheet and 1 row java.lang.AssertionError: expected [null] but found [User not found] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:442) at step_definition.UserSkillsMap. user_skills_map_user_user_validates_status_code_and_status_message_from_sheet_and_row(UserSkillsMap.java:167) at ?.UserSkillsMap_user User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_userGET" sheet and 1 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkillsMap.feature/UserSkillsMap_GET.feature:15) * Not displayable characters are replaced by '?'.	FAILED	0.049 s
4	And UserSkillsMap JSON schema is valid	SKIPPED	0.001 s

(S)- Validate GET one particular User-to-skill mappings with specific user ID

FAILED	DURATION - 0.095 s		Steps Total - 4 Pass - 2 Fail - 1 Skip - 1	
/ 11:15:49.576 AM // 11:15:49.671 AM /				
To check The GET method for UserSkillsMap API				

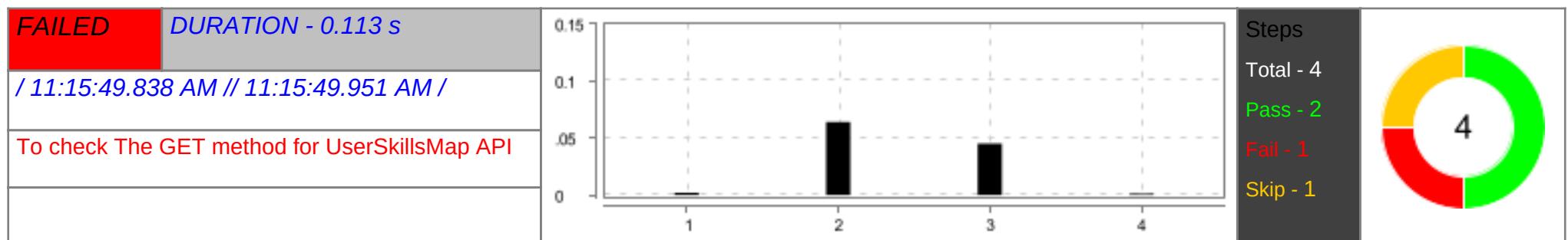
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid_username and password	PASSED	0.002 s
2	When UserSkillsMap_user User sends GET request on user id from "InputData_UserSkillsMap_userGET" and 2	PASSED	0.051 s
3	Then UserSkillsMap_user User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_userGET" sheet and 2 row java.lang.AssertionError: expected [null] but found [User not found] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:442) at step_definition.UserSkillsMap. user_skills_map_user_user_validates_status_code_and_status_message_from_sheet_and_row(UserSkillsMap.java:167) at ?.UserSkillsMap_user User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_userGET" sheet and 2 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkillsMap.feature/UserSkillsMap_GET.feature:15) * Not displayable characters are replaced by '?'. 	FAILED	0.040 s
4	And UserSkillsMap JSON schema is valid	SKIPPED	0.000 s

(S)- Validate GET one particular User-to-skill mappings with specific user ID

FAILED	DURATION - 0.115 s		Steps Total - 4 Pass - 2 Fail - 1 Skip - 1	
/ 11:15:49.695 AM // 11:15:49.810 AM /				
To check The GET method for UserSkillsMap API				

#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid username and password	PASSED	0.004 s
2	When UserSkillsMap_user User sends GET request on user id from "InputData_UserSkillsMap_userGET" and 3	PASSED	0.060 s
3	Then UserSkillsMap_user User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_userGET" sheet and 3 row java.lang.AssertionError: expected [null] but found [User not found] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:442) at step_definition.UserSkillsMap. user_skills_map_user_user_validates_status_code_and_status_message_from_sheet_and_row(UserSkillsMap.java:167) at ?.UserSkillsMap_user User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_userGET" sheet and 3 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkillsMap.feature/UserSkillsMap_GET.feature:15) * Not displayable characters are replaced by '?'. 	FAILED	0.049 s
4	And UserSkillsMap JSON schema is valid	SKIPPED	0.001 s

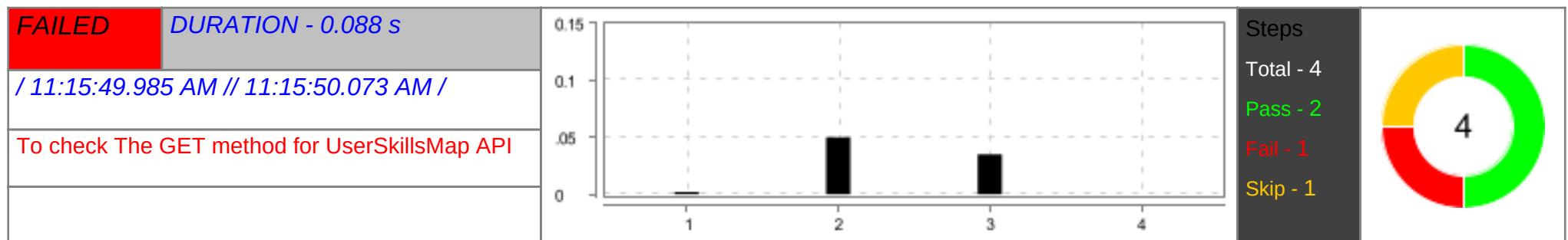
(S)- Validate GET one particular Skill-to-users mappings with specific skill ID



#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid username and password	PASSED	0.002 s
2	When UserSkillsMap_skill User sends GET request on user id from "InputData_UserSkillsMap_skilGET" and 0	PASSED	0.064 s
3	Then UserSkillsMap_skill User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_skilGET" sheet and 0 row java.lang.AssertionError: expected [200] but found [404] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652)	FAILED	0.045 s

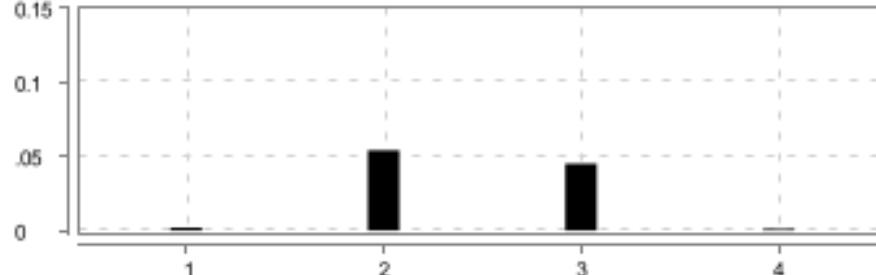
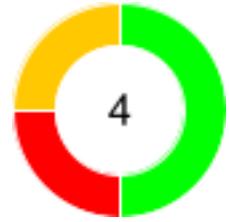
#	Step / Hook Details	Status	Duration
	<pre>at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.UserSkillsMap. user_skills_map_skill_user_validates_status_code_and_status_message_from_sheet_and_row(UserSkillsMap.java:210) at ?.UserSkillsMap_skill User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_skilGET" sheet and 0 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkillsMap.feature/UserSkillsMap_GET.feature:28) * Not displayable characters are replaced by '?'.</pre>		
4	And UserSkillsMap_skill JSON schema is valid	SKIPPED	0.001 s

(S)- Validate GET one particular Skill-to-users mappings with specific skill ID



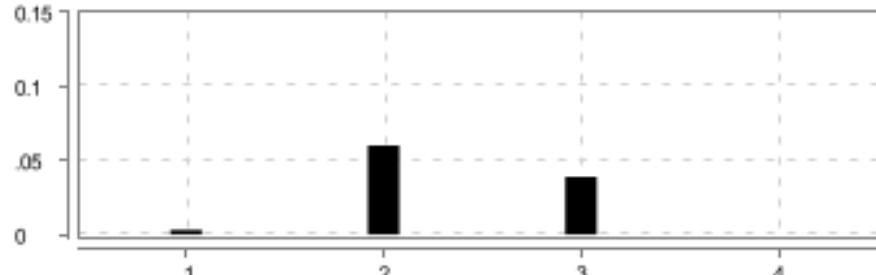
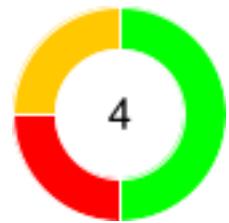
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid username and password	PASSED	0.002 s
2	When UserSkillsMap_skill User sends GET request on user id from "InputData_UserSkillsMap_skilGET" and 1	PASSED	0.050 s
3	Then UserSkillsMap_skill User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_skilGET" sheet and 1 row <pre>java.lang.AssertionError: expected [null] but found [Skill not found] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:442) at step_definition.UserSkillsMap. user_skills_map_skill_user_validates_status_code_and_status_message_from_sheet_and_row(UserSkillsMap.java:215) at ?.UserSkillsMap_skill User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_skilGET" sheet and 1 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkillsMap.feature/UserSkillsMap_GET.feature:28) * Not displayable characters are replaced by '?'.</pre>	FAILED	0.035 s
4	And UserSkillsMap_skill JSON schema is valid	SKIPPED	0.000 s

(S)- Validate GET one particular Skill-to-users mappings with specific skill ID

FAILED	DURATION - 0.104 s		Steps Total - 4 Pass - 2 Fail - 1 Skip - 1	
/ 11:15:50.094 AM // 11:15:50.198 AM /				
To check The GET method for UserSkillsMap API				

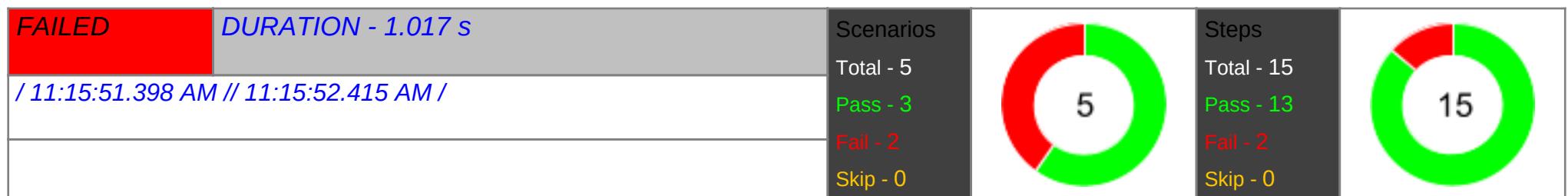
#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid username and password	PASSED	0.002 s
2	When UserSkillsMap_skill User sends GET request on user id from "InputData_UserSkillsMap_skilGET" and 2	PASSED	0.054 s
3	Then UserSkillsMap_skill User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_skilGET" sheet and 2 row java.lang.AssertionError: expected [null] but found [Skill not found] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:442) at step_definition.UserSkillsMap. user_skills_map_skill_user_validates_status_code_and_status_message_from_sheet_and_row(UserSkillsMap.java:215) at ?.UserSkillsMap_skill User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_skilGET" sheet and 2 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkillsMap.feature/UserSkillsMap_GET.feature:28) * Not displayable characters are replaced by '?'. 	FAILED	0.045 s
4	And UserSkillsMap_skill JSON schema is valid	SKIPPED	0.001 s

(S)- Validate GET one particular Skill-to-users mappings with specific skill ID

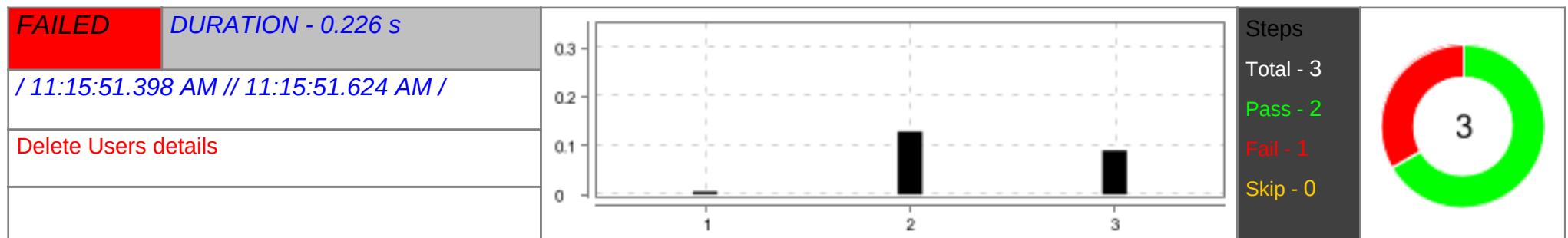
FAILED	DURATION - 0.104 s		Steps Total - 4 Pass - 2 Fail - 1 Skip - 1	
/ 11:15:50.223 AM // 11:15:50.327 AM /				
To check The GET method for UserSkillsMap API				

#	Step / Hook Details	Status	Duration
1	Given UserSkillsMap User is on Endpoint: url/UserSkillsMap with valid username and password	PASSED	0.003 s
2	When UserSkillsMap_skill User sends GET request on user id from "InputData_UserSkillsMap_skilGET" and 3	PASSED	0.060 s
3	Then UserSkillsMap_skill User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_skilGET" sheet and 3 row java.lang.AssertionError: expected [null] but found [Skill not found] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:442) at step_definition.UserSkillsMap. user_skills_map_skill_user_validates_status_code_and_status_message_from_sheet_and_row(UserSkillsMap.java:215) at ?.UserSkillsMap_skill User validates StatusCode and StatusMessage from "InputData_UserSkillsMap_skilGET" sheet and 3 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/userSkillsMap.feature/UserSkillsMap_GET.feature:28) * Not displayable characters are replaced by '?'. 	FAILED	0.039 s
4	And UserSkillsMap_skill JSON schema is valid	SKIPPED	0.000 s

(F)- Delete Users details

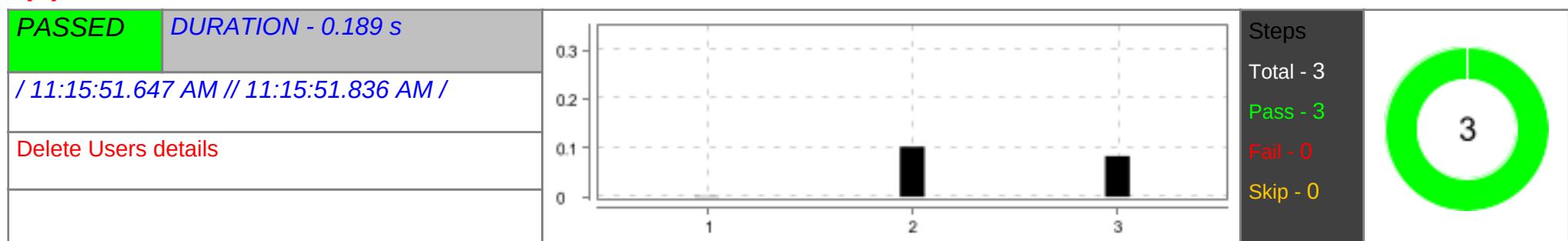


(S)- Validate the delete method for Users



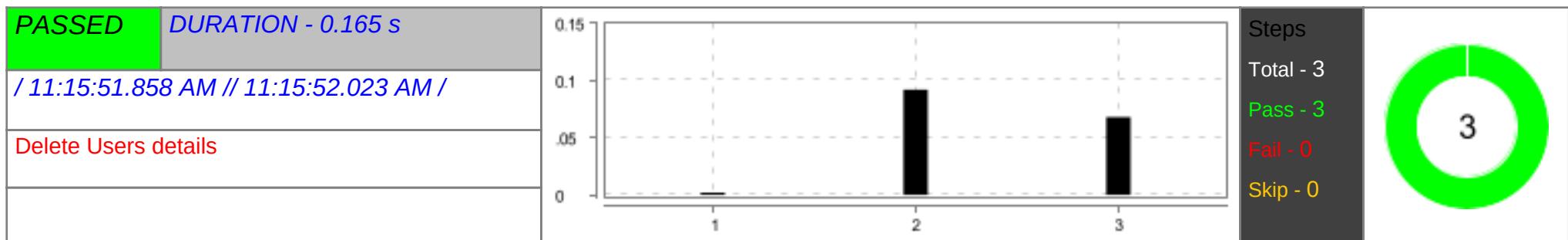
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.006 s
2	When UsersAPI user sends request id ON DELETE Method from "InputData_Users_POST" and 0	PASSED	0.129 s
3	<p>Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_DELETE" sheet and 0 row</p> <pre>java.lang.AssertionError: expected [200] but found [404] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:196) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_DELETE" sheet and 0 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_DELETE.feature:7) * Not displayable characters are replaced by '?'.</pre>	FAILED	0.090 s

(S)- Validate the delete method for Users



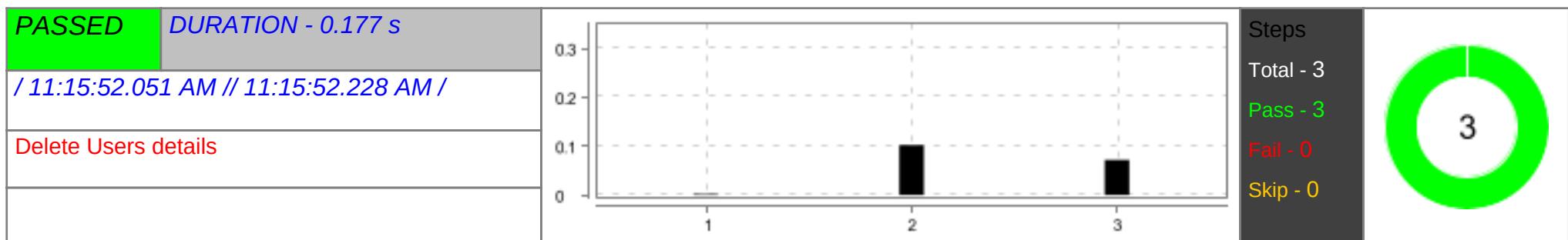
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI user sends request id ON DELETE Method from "InputData_Users_DELETE" and 1	PASSED	0.102 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_DELETE" sheet and 1 row	PASSED	0.083 s

(S)- Validate the delete method for Users



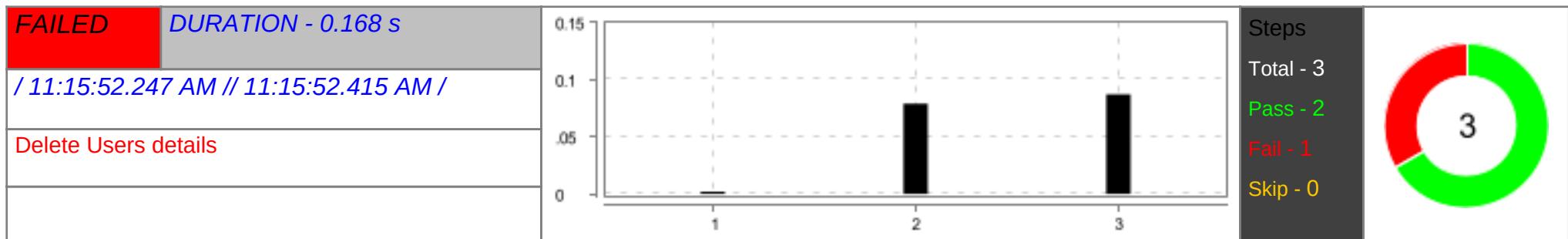
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI user sends request id ON DELETE Method from "InputData_Users_DELETE" and 2	PASSED	0.092 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_DELETE" sheet and 2 row	PASSED	0.068 s

(S)- Validate the delete method for Users



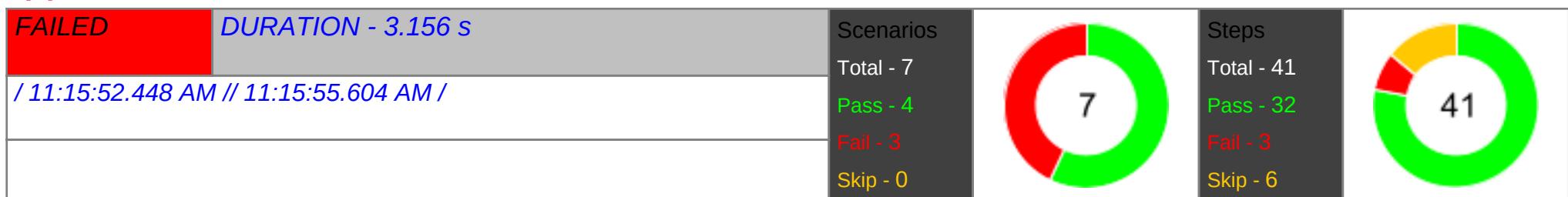
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI user sends request id ON DELETE Method from "InputData_Users_DELETE" and 3	PASSED	0.102 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_DELETE" sheet and 3 row	PASSED	0.072 s

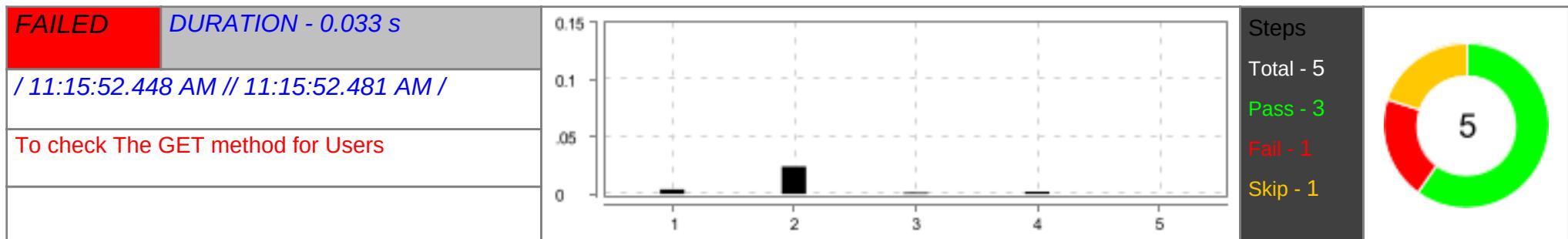
(S)- Validate the delete method for Users



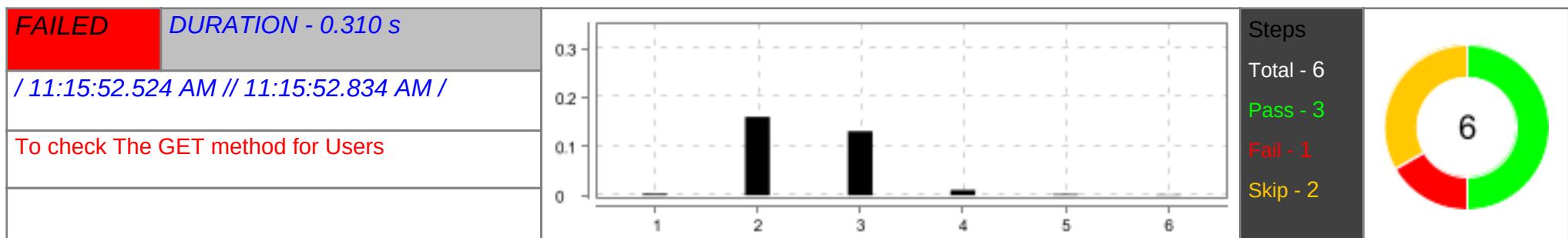
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI user sends request id ON DELETE Method from "InputData_Users_DELETE" and 4	PASSED	0.079 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_DELETE" sheet and 4 row java.lang.AssertionError: expected object to not be null at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.assertNotNull(Assert.java:686) at org.testng.Assert.assertNotNull(Assert.java:671) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:199) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_DELETE" sheet and 4 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_DELETE.feature:7) * Not displayable characters are replaced by '?'.	FAILED	0.087 s

(F)- To check The GET method for Users



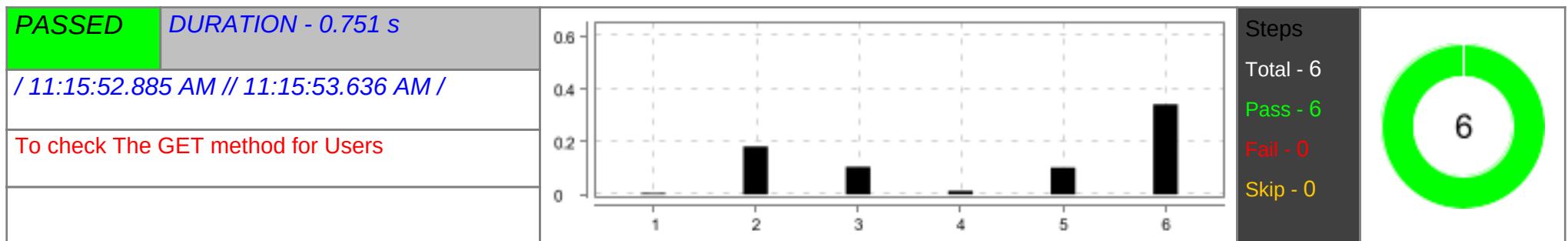
(S)- Validate GET all Users

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.004 s
2	When UsersAPI User sends GET request	PASSED	0.024 s
3	Then UsersAPI User validates StatusCode	PASSED	0.001 s
4	And UsersAPI JSON schema is valid java.lang.NullPointerException at java.base/java.util.Objects.requireNonNull(Objects.java:221) at java.base/java.lang.ClassLoader.getResource(ClassLoader.java:1392) at io.restassured.module.jsv.JsonSchemaValidator.matchesJsonSchemaInClasspath(JsonSchemaValidator.java:117) at util.JSON_Schema_Validation.cls_JSON_SchemaValidation(JSON_Schema_Validation.java:29) at step_definition.Users.UsersAPI_json_schema_is_valid(Users.java:121) at ?.UsersAPI JSON schema is valid(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/ Users_GET.feature:9) * Not displayable characters are replaced by '?'. 	FAILED	0.002 s
5	And check the Database for all users	SKIPPED	0.000 s

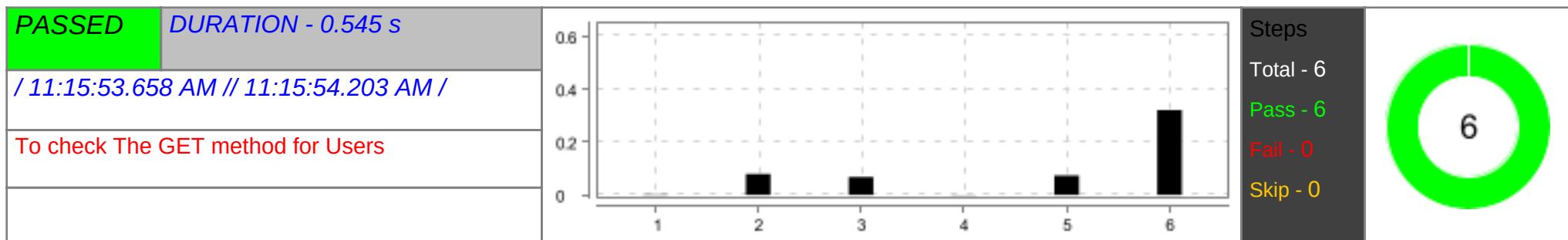
(S)- Validate Users Api with specific ID

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.003 s
2	When UsersAPI User sends GET request on userid from "InputData_User_GET" and 0	PASSED	0.160 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_User_GET" sheet and 0 row	PASSED	0.131 s
4	And UsersAPI JSON schema is valid for "GET" java.lang.NullPointerException at java.base/java.util.Objects.requireNonNull(Objects.java:221) at java.base/java.lang.ClassLoader.getResource(ClassLoader.java:1392) at io.restassured.module.jsv.JsonSchemaValidator.matchesJsonSchemaInClasspath(JsonSchemaValidator.java:117) at util.JSON_Schema_Validation.cls_JSON_SchemaValidation(JSON_Schema_Validation.java:29) at step_definition.Users.UsersAPI_json_schema_is_valid_for(Users.java:150) at ?.UsersAPI JSON schema is valid for "GET"(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_GET.feature:15) * Not displayable characters are replaced by '?'.	FAILED	0.011 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_User_GET" and 0	SKIPPED	0.002 s
6	And UsersAPI check the Database	SKIPPED	0.001 s

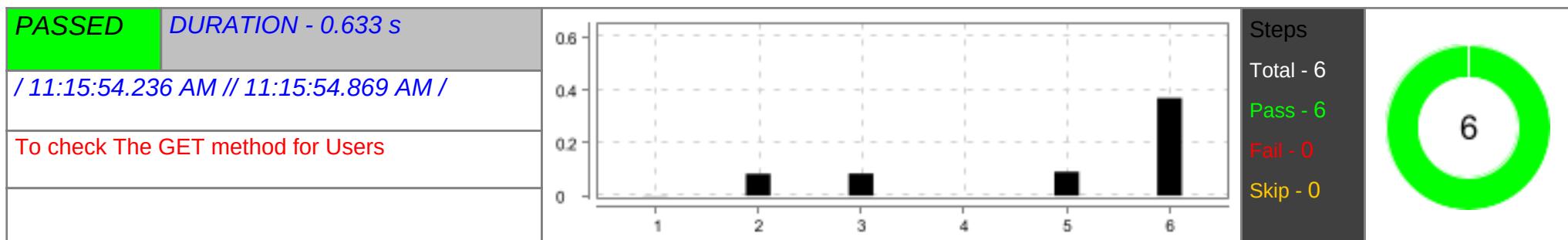
(S)- Validate Users Api with specific ID



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.005 s
2	When UsersAPI User sends GET request on userid from "InputData_User_GET" and 1	PASSED	0.182 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_User_GET" sheet and 1 row	PASSED	0.105 s
4	And UsersAPI JSON schema is valid for "GET"	PASSED	0.013 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_User_GET" and 1	PASSED	0.102 s
6	And UsersAPI check the Database	PASSED	0.342 s

(S)- Validate Users Api with specific ID

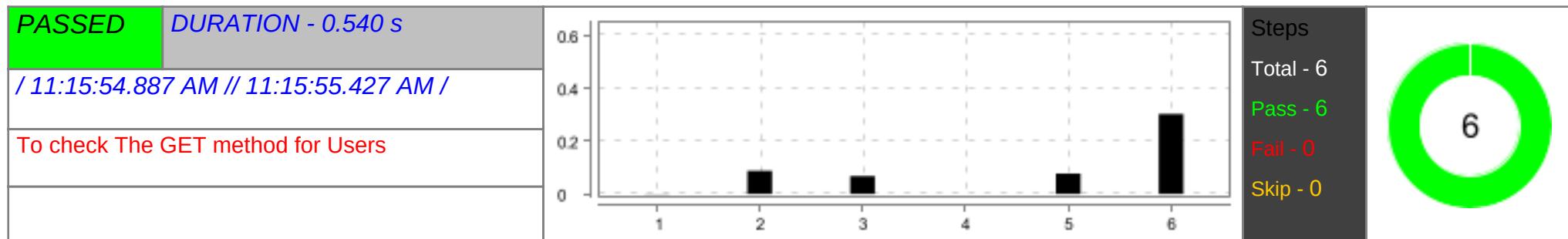
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends GET request on userid from "InputData_User_GET" and 2	PASSED	0.079 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_User_GET" sheet and 2 row	PASSED	0.068 s
4	And UsersAPI JSON schema is valid for "GET"	PASSED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_User_GET" and 2	PASSED	0.073 s
6	And UsersAPI check the Database	PASSED	0.321 s

(S)- Validate Users Api with specific ID

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends GET request on userid from "InputData_User_GET" and 3	PASSED	0.083 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_User_GET" sheet and 3 row	PASSED	0.084 s
4	And UsersAPI JSON schema is valid for "GET"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_User_GET" and 3	PASSED	0.091 s

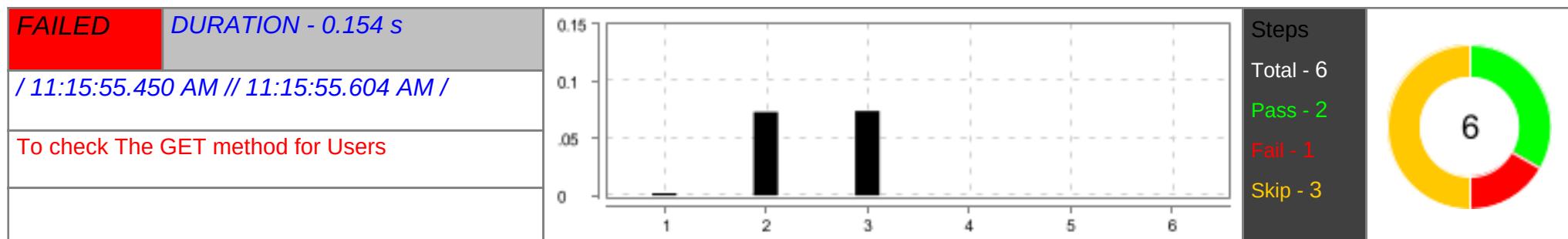
#	Step / Hook Details	Status	Duration
6	And UsersAPI check the Database	PASSED	0.370 s

(S)- Validate Users Api with specific ID



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends GET request on userid from "InputData_User_GET" and 4	PASSED	0.088 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_User_GET" sheet and 4 row	PASSED	0.068 s
4	And UsersAPI JSON schema is valid for "GET"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_User_GET" and 4	PASSED	0.077 s
6	And UsersAPI check the Database	PASSED	0.303 s

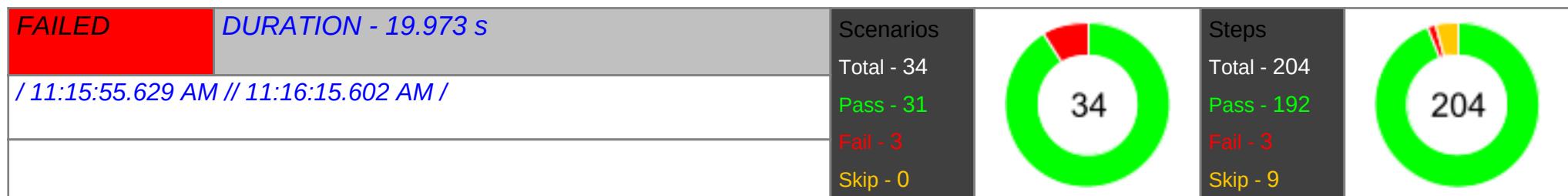
(S)- Validate Users Api with specific ID



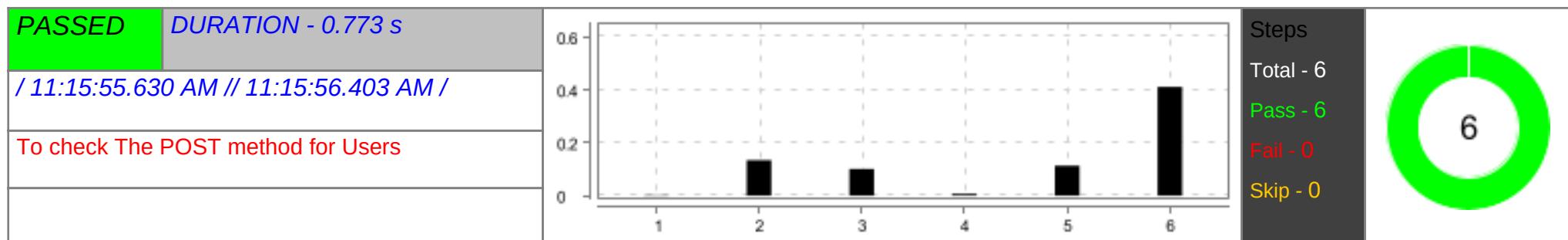
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends GET request on userid from "InputData_User_GET" and 5	PASSED	0.073 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_User_GET" sheet and 5 row	FAILED	0.074 s

#	Step / Hook Details	Status	Duration
	java.lang.AssertionError: expected object to not be null at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.assertNotNull(Assert.java:686) at org.testng.Assert.assertNotNull(Assert.java:671) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:199) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_User_GET" sheet and 5 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_GET.feature:14) * Not displayable characters are replaced by '?'. 4 And UsersAPI JSON schema is valid for "GET" 5 And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_User_GET" and 5 6 And UsersAPI check the Database		
4	And UsersAPI JSON schema is valid for "GET"	SKIPPED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_User_GET" and 5	SKIPPED	0.000 s
6	And UsersAPI check the Database	SKIPPED	0.000 s

(F)- To check The POST method for Users



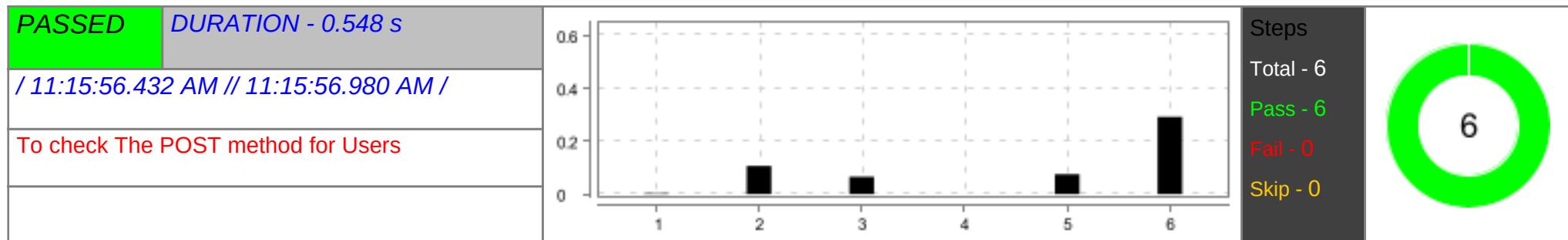
(S)- Validate POST Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 0 with valid JSON Schema	PASSED	0.134 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 0 row	PASSED	0.101 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.007 s

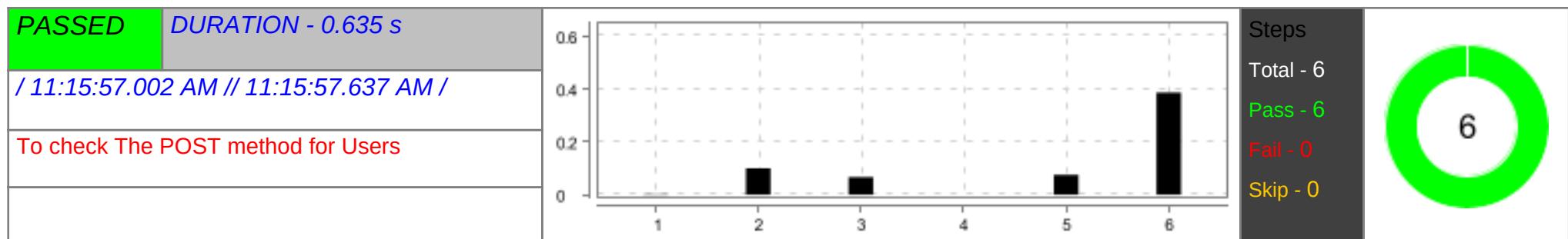
#	Step / Hook Details	Status	Duration
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 0	PASSED	0.113 s
6	And UsersAPI check the Database	PASSED	0.412 s

(S)- Validate POST Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.003 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 1 with valid JSON Schema	PASSED	0.107 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 1 row	PASSED	0.066 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 1	PASSED	0.075 s
6	And UsersAPI check the Database	PASSED	0.293 s

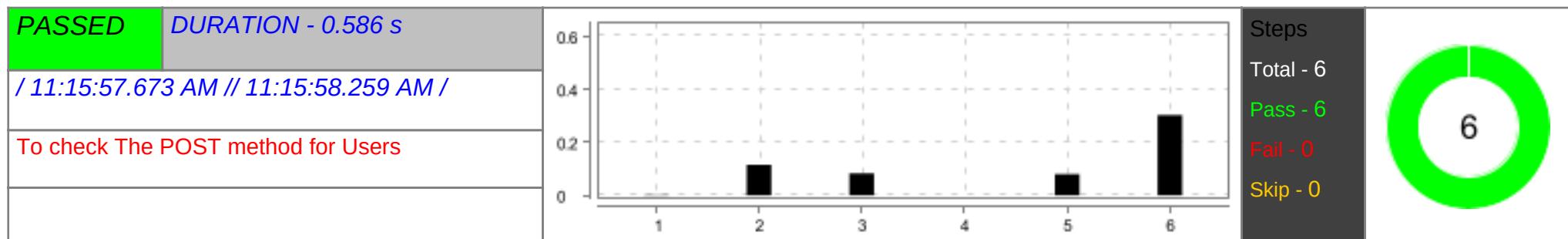
(S)- Validate POST Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 2 with valid JSON Schema	PASSED	0.100 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 2 row	PASSED	0.067 s

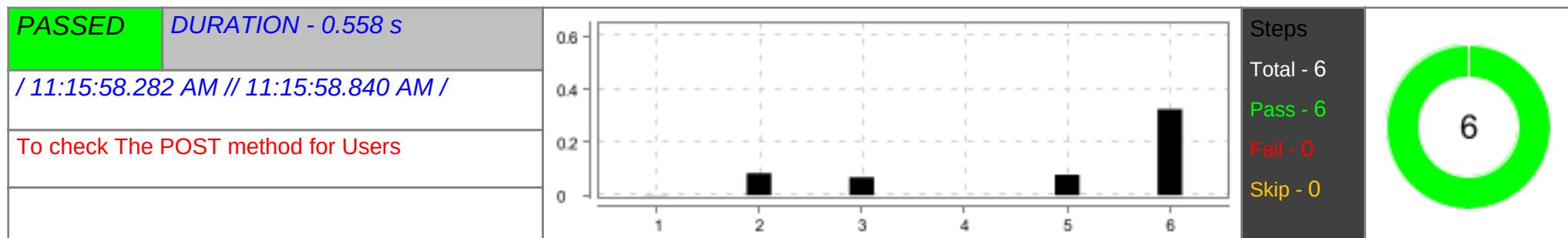
#	Step / Hook Details	Status	Duration
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 2	PASSED	0.075 s
6	And UsersAPI check the Database	PASSED	0.387 s

(S)- Validate POST Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 3 with valid JSON Schema	PASSED	0.115 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 3 row	PASSED	0.083 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 3	PASSED	0.080 s
6	And UsersAPI check the Database	PASSED	0.303 s

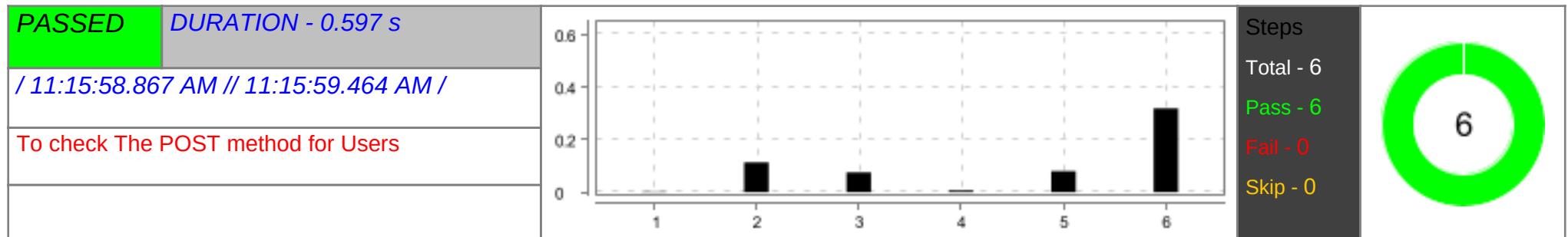
(S)- Validate POST Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 4 with valid JSON Schema	PASSED	0.083 s

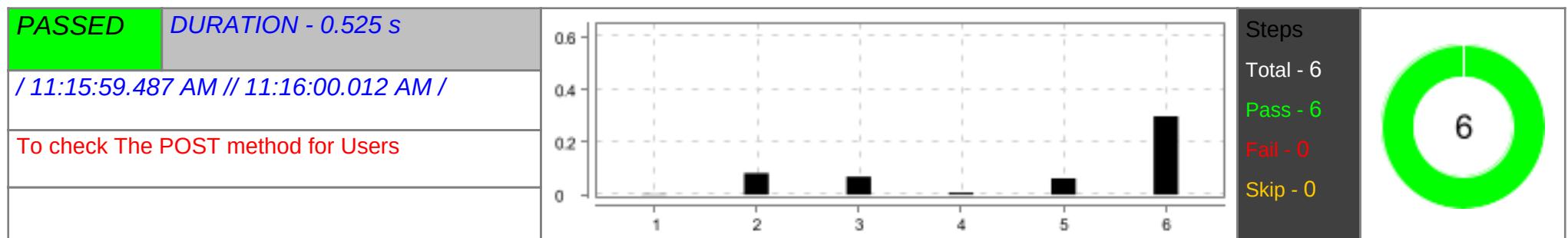
#	Step / Hook Details	Status	Duration
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 4 row	PASSED	0.068 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 4	PASSED	0.077 s
6	And UsersAPI check the Database	PASSED	0.325 s

(S)- Validate POST Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 5 with valid JSON Schema	PASSED	0.113 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 5 row	PASSED	0.075 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.007 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 5	PASSED	0.080 s
6	And UsersAPI check the Database	PASSED	0.319 s

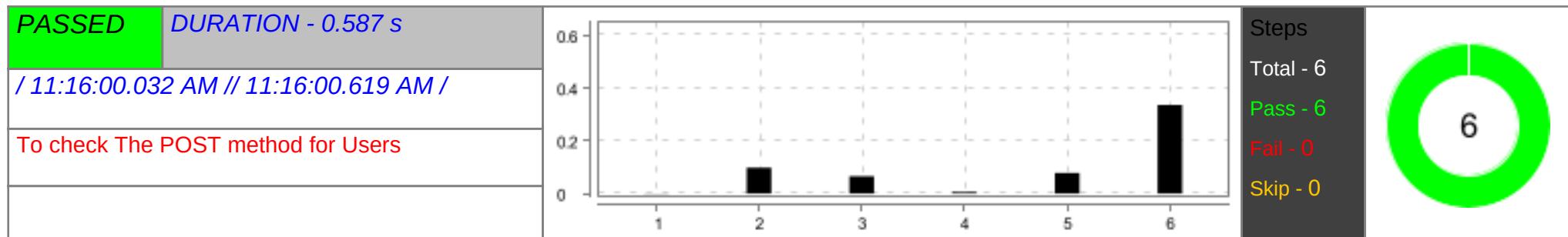
(S)- Validate POST Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s

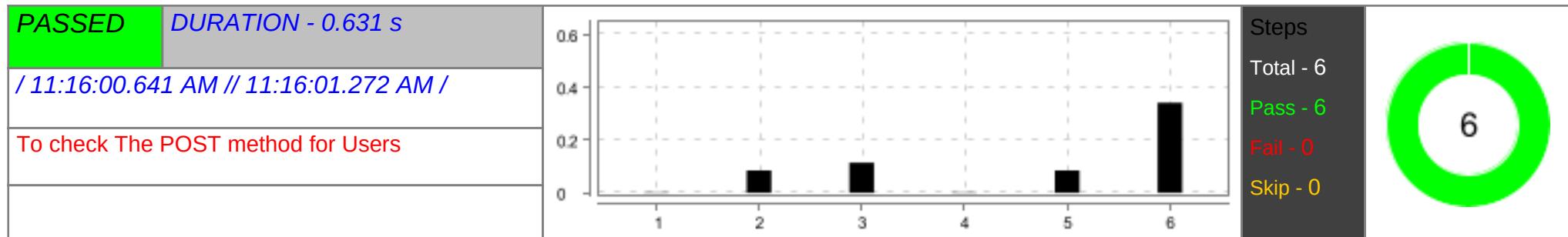
#	Step / Hook Details	Status	Duration
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 6 with valid JSON Schema	PASSED	0.082 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 6 row	PASSED	0.069 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.008 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 6	PASSED	0.062 s
6	And UsersAPI check the Database	PASSED	0.298 s

(S)- Validate POST Users



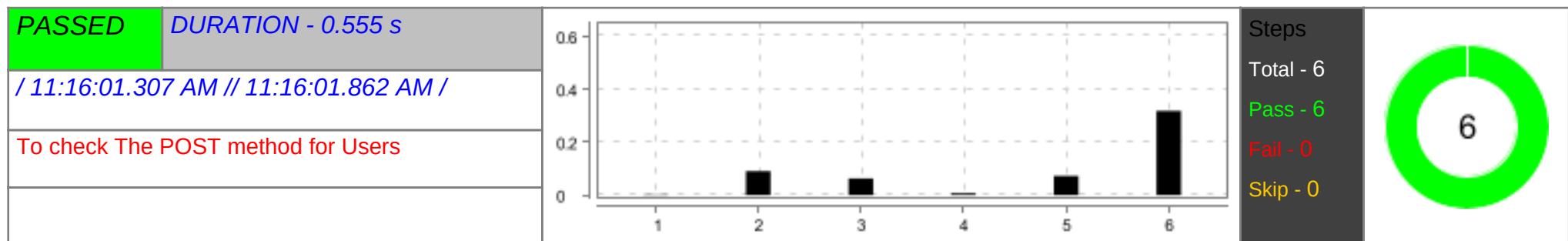
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 7 with valid JSON Schema	PASSED	0.098 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 7 row	PASSED	0.066 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.006 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 7	PASSED	0.078 s
6	And UsersAPI check the Database	PASSED	0.336 s

(S)- Validate POST Users



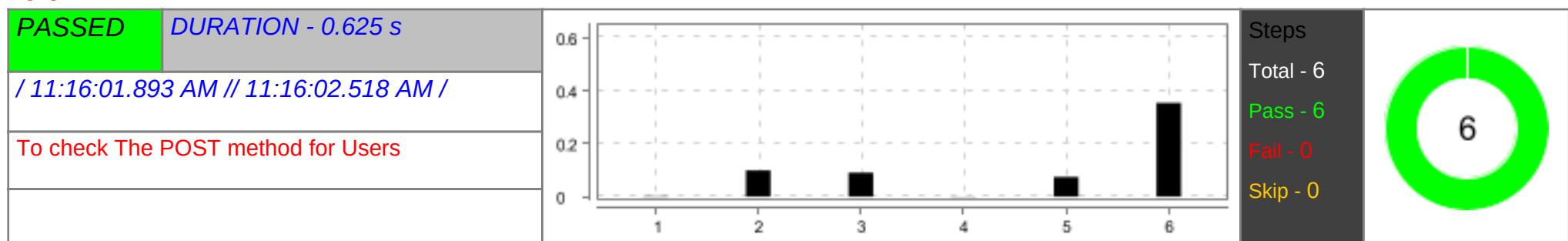
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 8 with valid JSON Schema	PASSED	0.085 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 8 row	PASSED	0.114 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.002 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 8	PASSED	0.085 s
6	And UsersAPI check the Database	PASSED	0.342 s

(S)- Validate POST Users



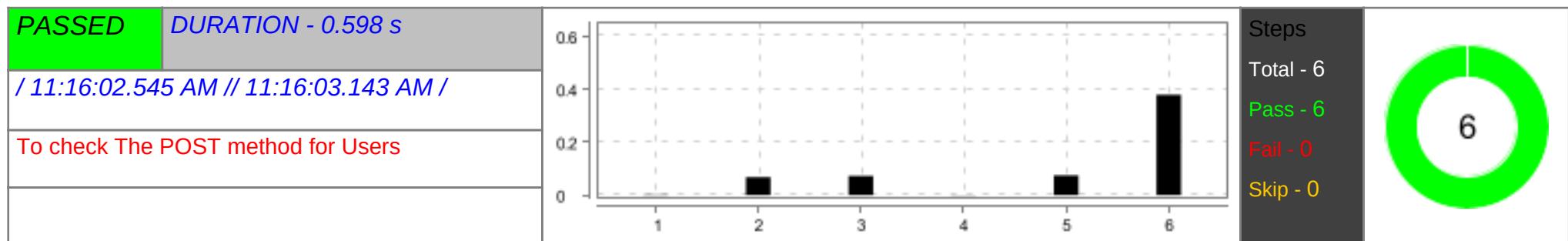
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 9 with valid JSON Schema	PASSED	0.091 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 9 row	PASSED	0.063 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.007 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 9	PASSED	0.072 s
6	And UsersAPI check the Database	PASSED	0.318 s

(S)- Validate POST Users



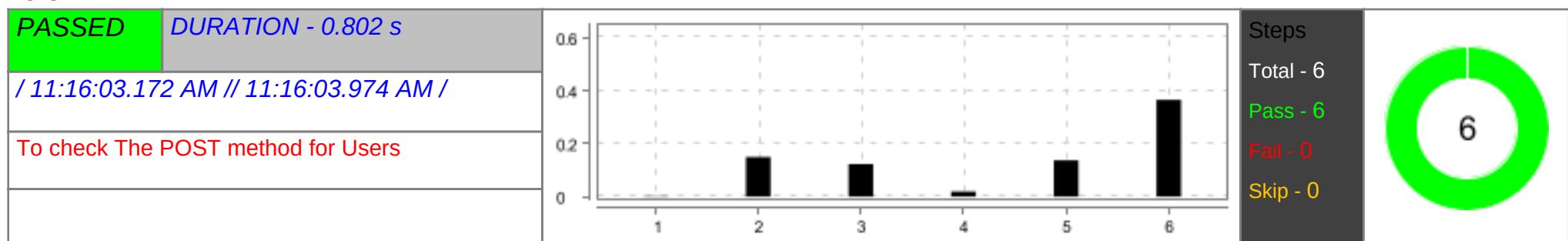
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 10 with valid JSON Schema	PASSED	0.099 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 10 row	PASSED	0.091 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.001 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 10	PASSED	0.074 s
6	And UsersAPI check the Database	PASSED	0.355 s

(S)- Validate POST Users



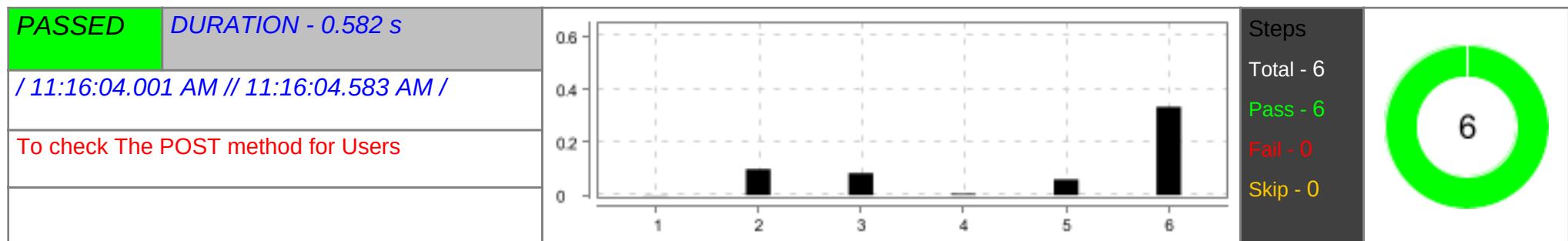
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 11 with valid JSON Schema	PASSED	0.068 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 11 row	PASSED	0.072 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.001 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 11	PASSED	0.074 s
6	And UsersAPI check the Database	PASSED	0.379 s

(S)- Validate POST Users



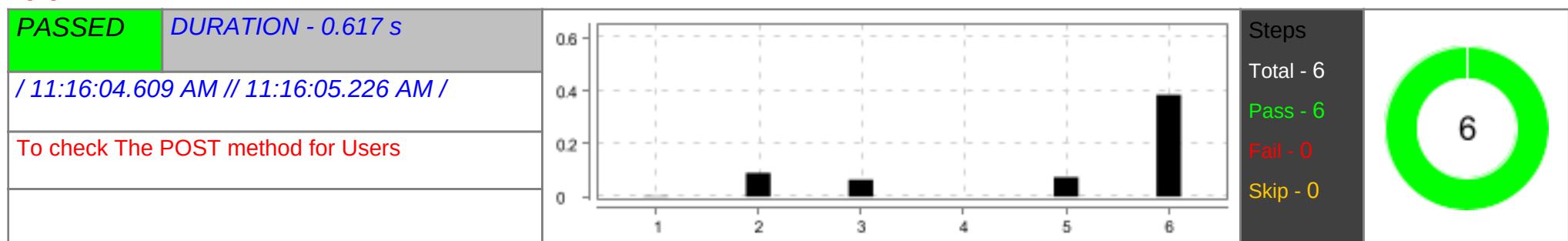
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 12 with valid JSON Schema	PASSED	0.150 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 12 row	PASSED	0.123 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.019 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 12	PASSED	0.137 s
6	And UsersAPI check the Database	PASSED	0.366 s

(S)- Validate POST Users



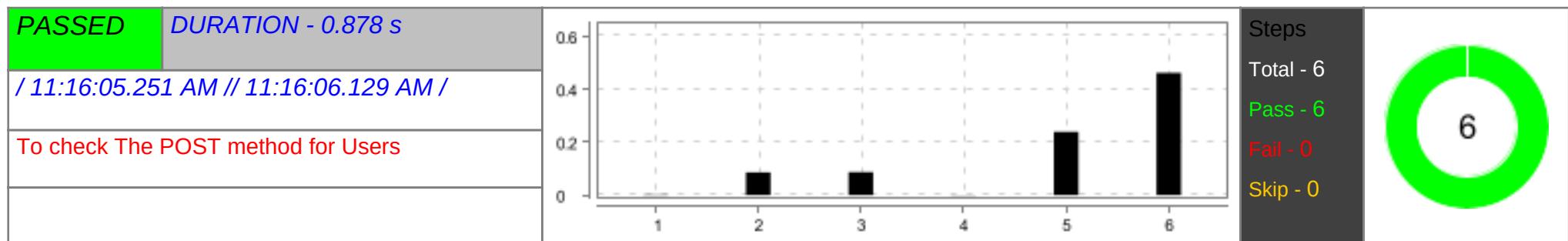
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 13 with valid JSON Schema	PASSED	0.098 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 13 row	PASSED	0.083 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.005 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 13	PASSED	0.059 s
6	And UsersAPI check the Database	PASSED	0.334 s

(S)- Validate POST Users



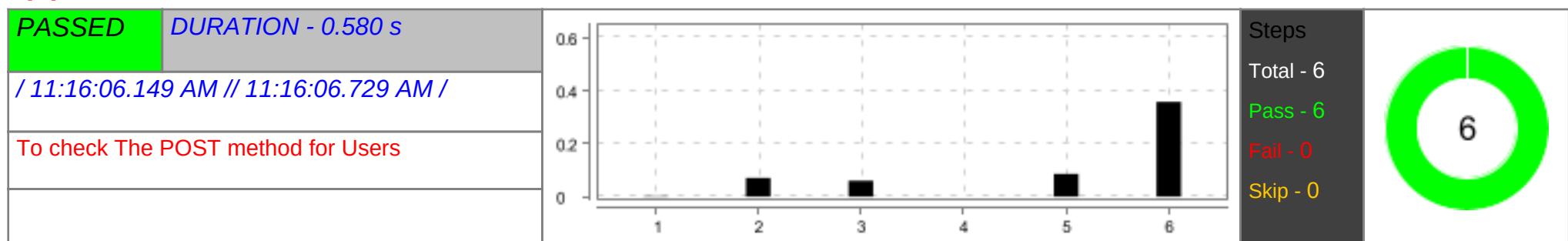
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 14 with valid JSON Schema	PASSED	0.090 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 14 row	PASSED	0.064 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 14	PASSED	0.073 s
6	And UsersAPI check the Database	PASSED	0.385 s

(S)- Validate POST Users



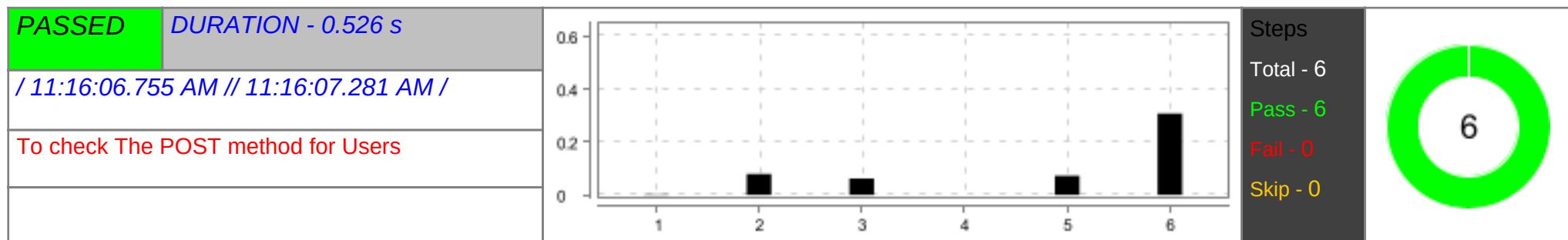
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 15 with valid JSON Schema	PASSED	0.086 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 15 row	PASSED	0.087 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.001 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 15	PASSED	0.239 s
6	And UsersAPI check the Database	PASSED	0.462 s

(S)- Validate POST Users



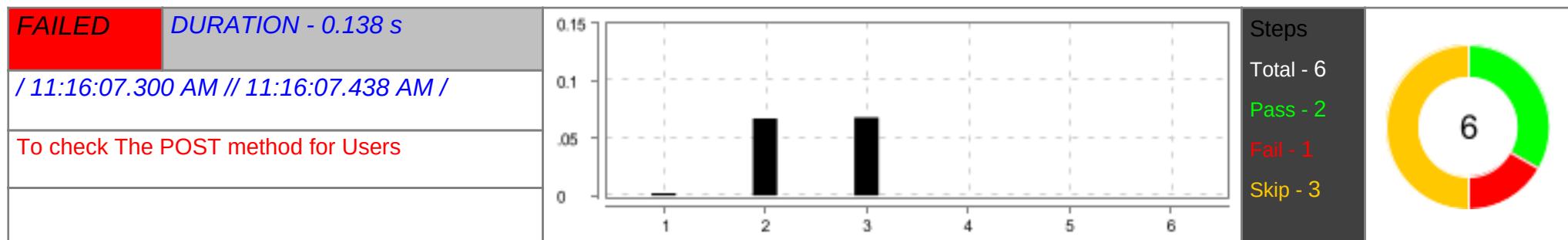
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 16 with valid JSON Schema	PASSED	0.071 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 16 row	PASSED	0.060 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 16	PASSED	0.086 s
6	And UsersAPI check the Database	PASSED	0.358 s

(S)- Validate POST Users



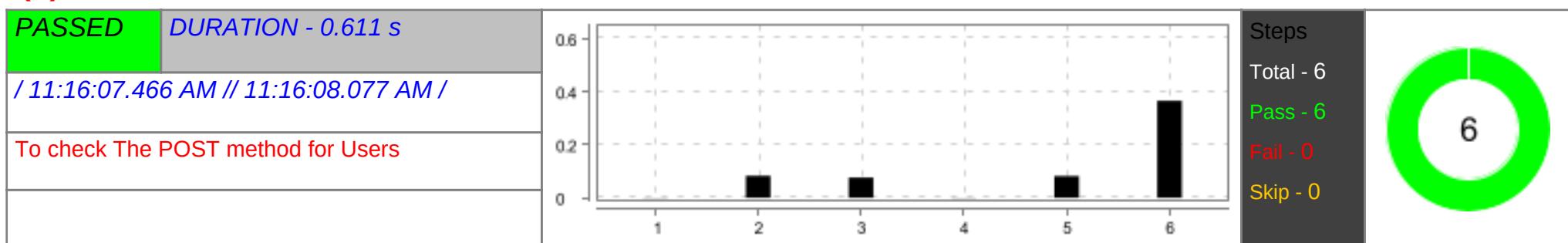
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 17 with valid JSON Schema	PASSED	0.079 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 17 row	PASSED	0.062 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 17	PASSED	0.072 s
6	And UsersAPI check the Database	PASSED	0.308 s

(S)- Validate POST Users

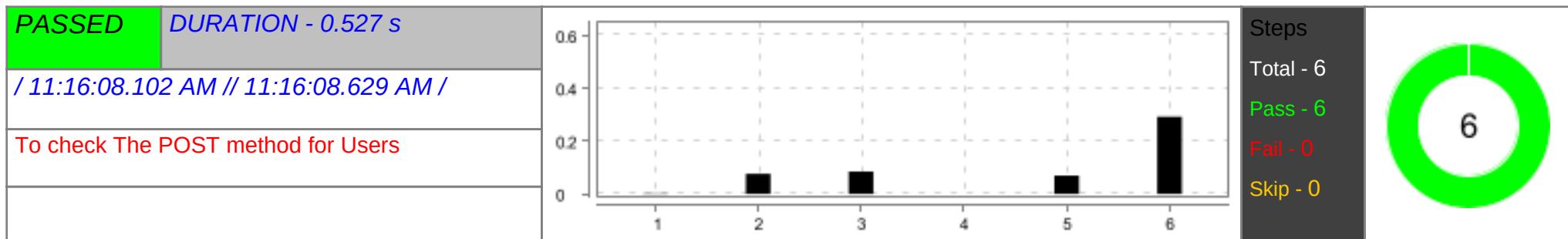


#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 18 with valid JSON Schema	PASSED	0.067 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 18 row java.lang.AssertionError: expected [201] but found [400] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:196) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 18 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_POST.feature:7) * Not displayable characters are replaced by '?'. at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 18 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_POST.feature:7)	FAILED	0.068 s
4	And UsersAPI JSON schema is valid for "POST"	SKIPPED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 18	SKIPPED	0.000 s
6	And UsersAPI check the Database	SKIPPED	0.000 s

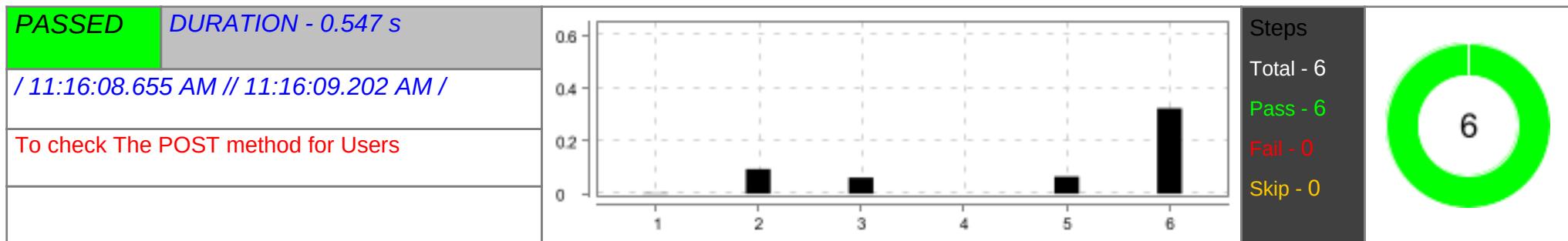
(S)- Validate POST Users



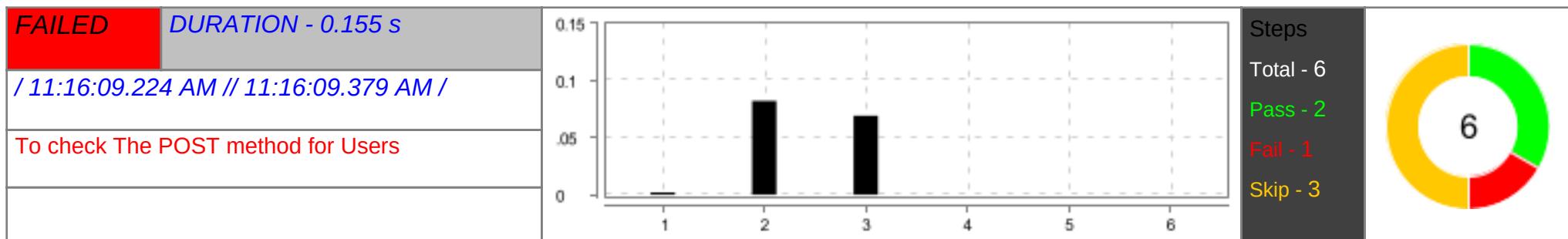
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 19 with valid JSON Schema	PASSED	0.083 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 19 row	PASSED	0.076 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.001 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 19	PASSED	0.082 s
6	And UsersAPI check the Database	PASSED	0.366 s

(S)- Validate POST Users

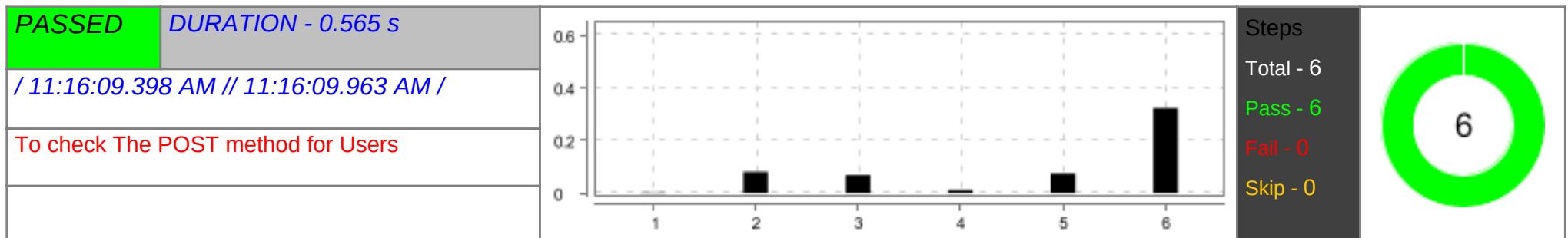
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 20 with valid JSON Schema	PASSED	0.076 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 20 row	PASSED	0.085 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 20	PASSED	0.070 s
6	And UsersAPI check the Database	PASSED	0.293 s

(S)- Validate POST Users

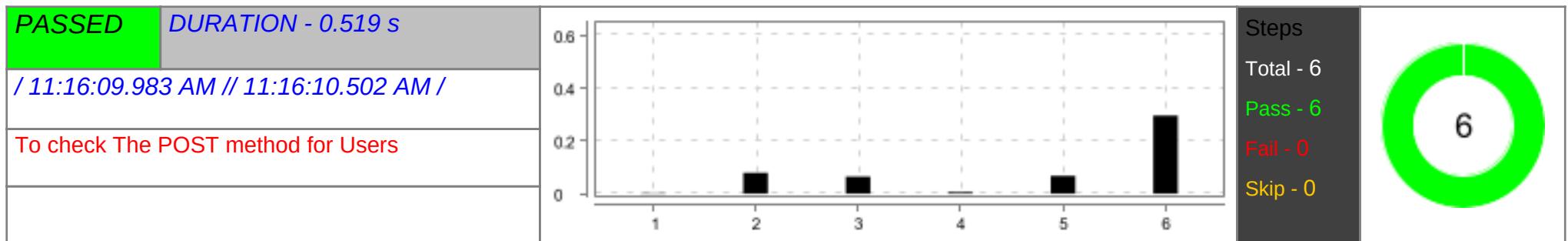
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 21 with valid JSON Schema	PASSED	0.093 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 21 row	PASSED	0.061 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 21	PASSED	0.065 s
6	And UsersAPI check the Database	PASSED	0.324 s

(S)- Validate POST Users

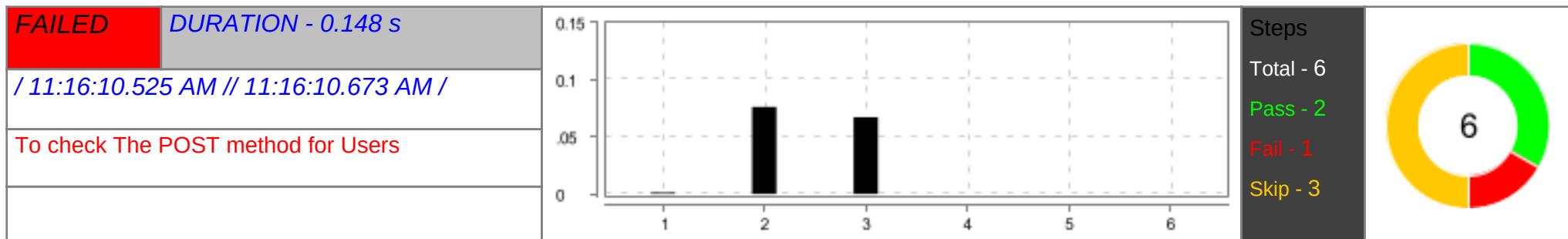
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 22 with valid JSON Schema	PASSED	0.082 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 22 row java.lang.AssertionError: expected [400] but found [201] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:196) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 22 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_POST.feature:7) * Not displayable characters are replaced by '?'.	FAILED	0.069 s
4	And UsersAPI JSON schema is valid for "POST"	SKIPPED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 22	SKIPPED	0.000 s
6	And UsersAPI check the Database	SKIPPED	0.000 s

(S)- Validate POST Users

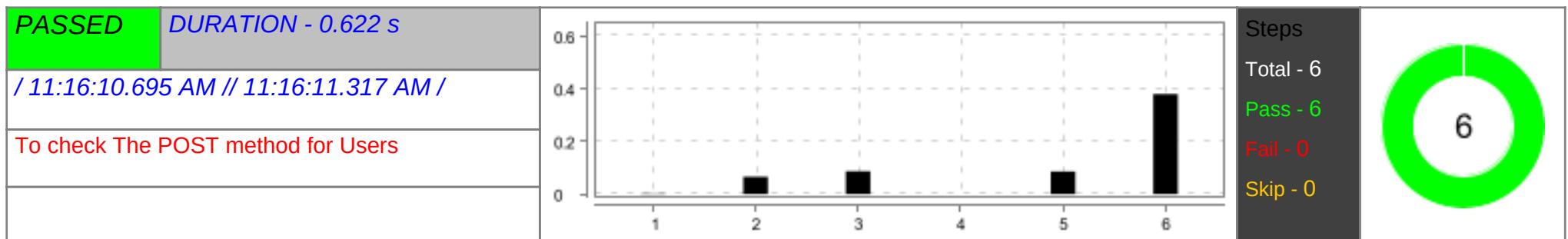
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 23 with valid JSON Schema	PASSED	0.081 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 23 row	PASSED	0.069 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.011 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 23	PASSED	0.075 s
6	And UsersAPI check the Database	PASSED	0.324 s

(S)- Validate POST Users

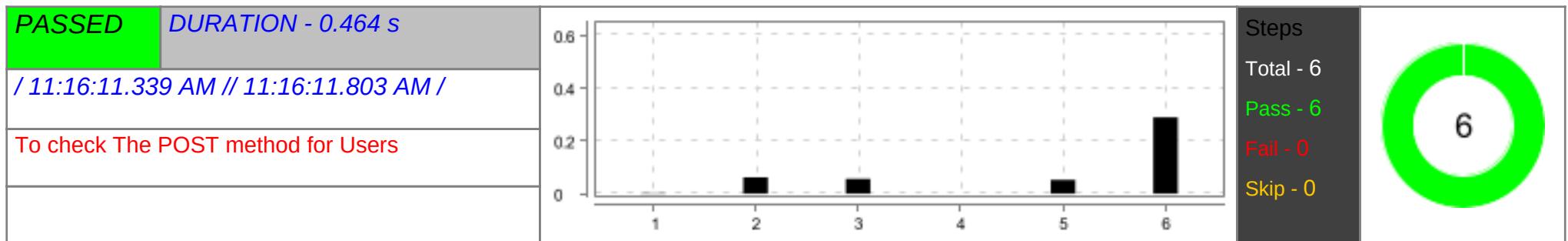
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 24 with valid JSON Schema	PASSED	0.079 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 24 row	PASSED	0.065 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.006 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 24	PASSED	0.068 s
6	And UsersAPI check the Database	PASSED	0.296 s

(S)- Validate POST Users

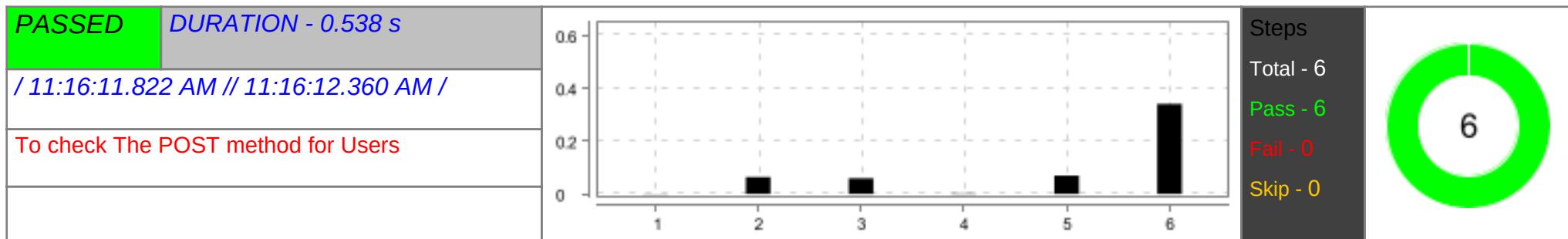
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends POST request body from "InputData_Users_POST" and 25 with valid JSON Schema	PASSED	0.076 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 25 row java.lang.AssertionError: expected [0] but found [400] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:196) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_POST" sheet and 25 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_POST.feature:7) * Not displayable characters are replaced by '?'.	FAILED	0.067 s
4	And UsersAPI JSON schema is valid for "POST"	SKIPPED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "InputData_Users_POST" and 25	SKIPPED	0.000 s
6	And UsersAPI check the Database	SKIPPED	0.000 s

(S)- Validate POST Users

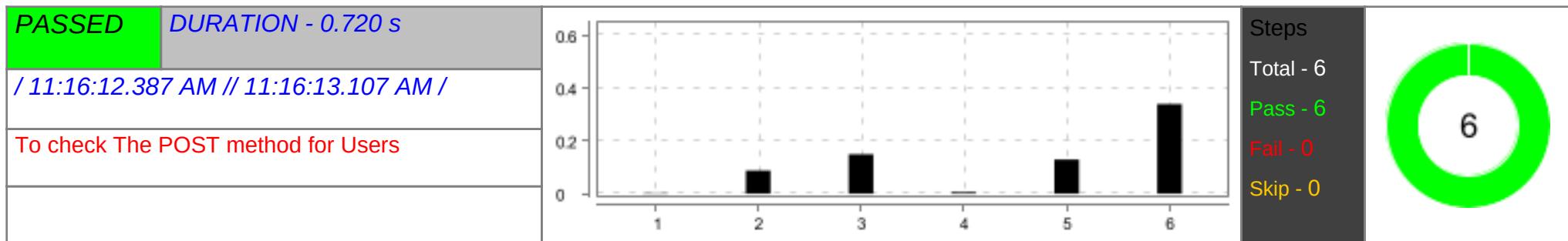
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "location_POST" and 0 with valid JSON Schema	PASSED	0.066 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "location_POST" sheet and 0 row	PASSED	0.087 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "location_POST" and 0	PASSED	0.086 s
6	And UsersAPI check the Database	PASSED	0.380 s

(S)- Validate POST Users

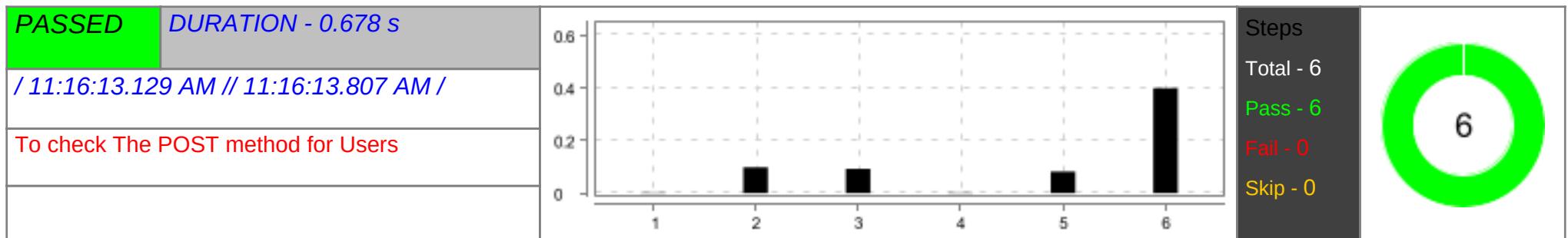
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "time_zone_POST" and 0 with valid JSON Schema	PASSED	0.062 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "time_zone_POST" sheet and 0 row	PASSED	0.056 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "time_zone_POST" and 0	PASSED	0.052 s
6	And UsersAPI check the Database	PASSED	0.290 s

(S)- Validate POST Users

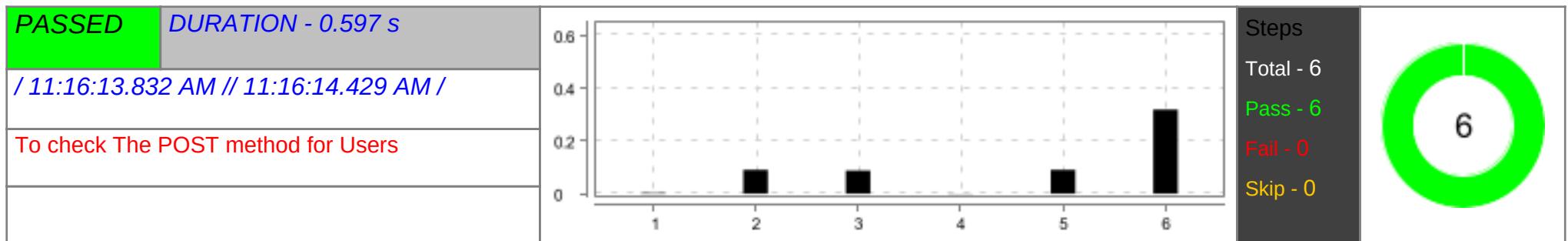
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends POST request body from "linkedin_url_POST" and 0 with valid JSON Schema	PASSED	0.064 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "linkedin_url_POST" sheet and 0 row	PASSED	0.059 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.002 s
5	And UsersAPI User should receive the user details in JSON body from "linkedin_url_POST" and 0	PASSED	0.070 s
6	And UsersAPI check the Database	PASSED	0.341 s

(S)- Validate POST Users

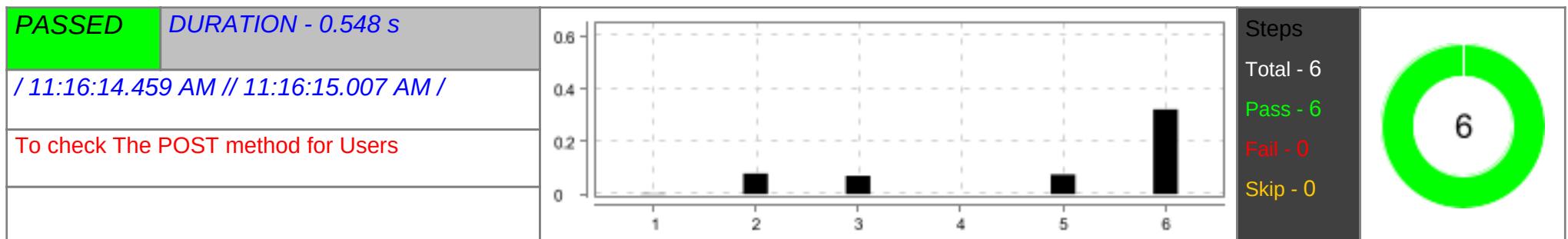
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "education_ug_POST" and 0 with valid JSON Schema	PASSED	0.088 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "education_ug_POST" sheet and 0 row	PASSED	0.150 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.006 s
5	And UsersAPI User should receive the user details in JSON body from "education_ug_POST" and 0	PASSED	0.130 s
6	And UsersAPI check the Database	PASSED	0.340 s

(S)- Validate POST Users

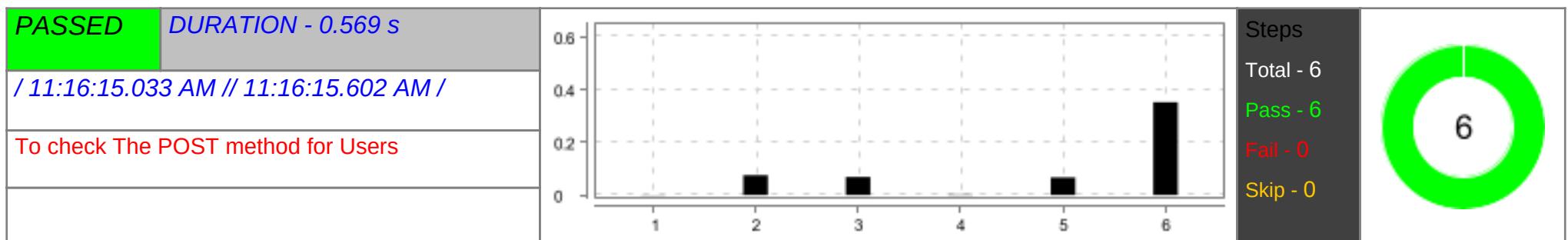
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "education_pg_POST" and 0 with valid JSON Schema	PASSED	0.098 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "education_pg_POST" sheet and 0 row	PASSED	0.092 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.002 s
5	And UsersAPI User should receive the user details in JSON body from "education_pg_POST" and 0	PASSED	0.082 s
6	And UsersAPI check the Database	PASSED	0.399 s

(S)- Validate POST Users

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.003 s
2	When UsersAPI User sends POST request body from "visa_status_POST" and 0 with valid JSON Schema	PASSED	0.091 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "visa_status_POST" sheet and 0 row	PASSED	0.088 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.001 s
5	And UsersAPI User should receive the user details in JSON body from "visa_status_POST" and 0	PASSED	0.091 s
6	And UsersAPI check the Database	PASSED	0.319 s

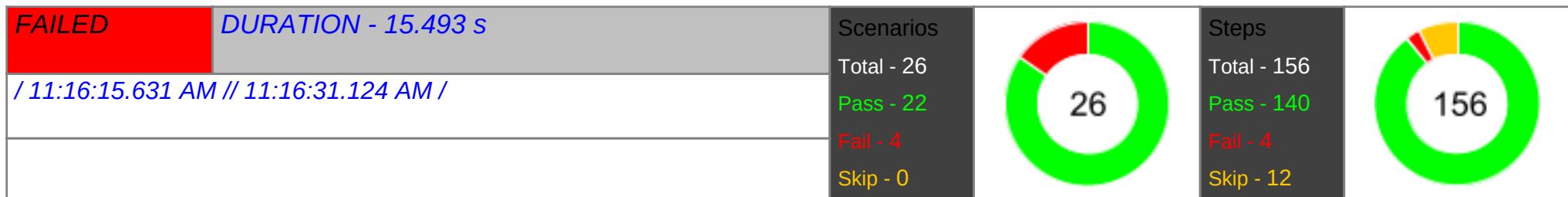
(S)- Validate POST Users

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends POST request body from "comments_POST" and 0 with valid JSON Schema	PASSED	0.078 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "comments_POST" sheet and 0 row	PASSED	0.070 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.000 s
5	And UsersAPI User should receive the user details in JSON body from "comments_POST" and 0	PASSED	0.074 s
6	And UsersAPI check the Database	PASSED	0.322 s

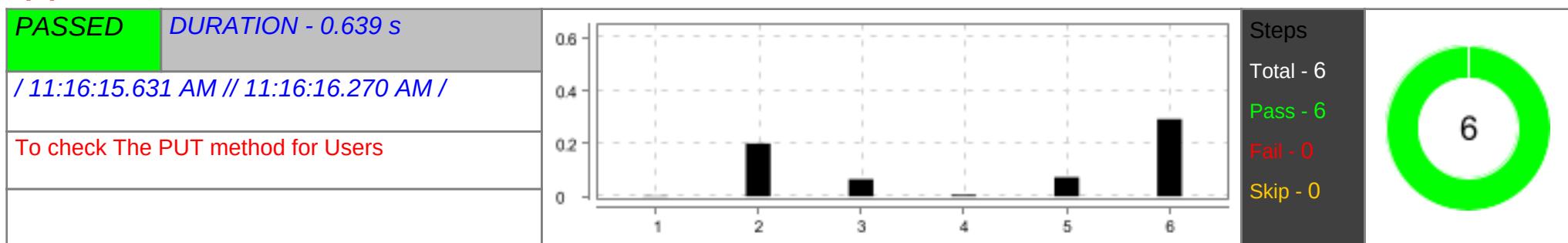
(S)- Validate POST Users

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends POST request body from "Invalid_Content" and 0 with valid JSON Schema	PASSED	0.075 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "Invalid_Content" sheet and 0 row	PASSED	0.069 s
4	And UsersAPI JSON schema is valid for "POST"	PASSED	0.002 s
5	And UsersAPI User should receive the user details in JSON body from "Invalid_Content" and 0	PASSED	0.067 s
6	And UsersAPI check the Database	PASSED	0.353 s

(F)- To check The PUT method for Users

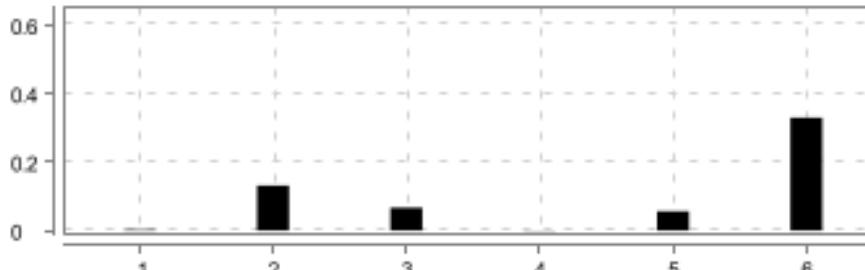
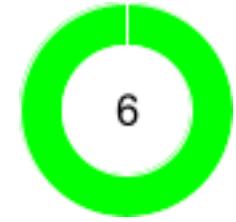


(S)- Validate PUT method for Users



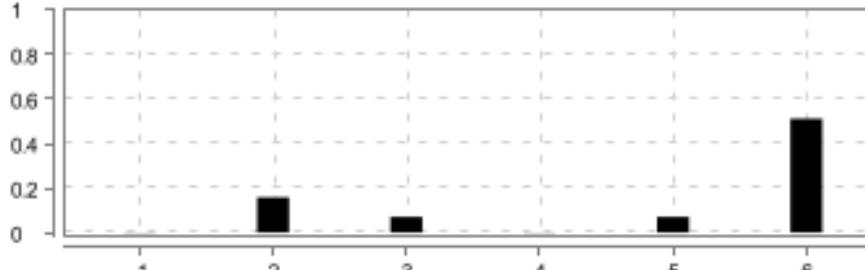
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 0 with valid JSON schema	PASSED	0.200 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 0 row	PASSED	0.065 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.006 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location,time_zone,linkedin_url from "InputData_Users_PUT" and 0	PASSED	0.072 s
6	And UsersAPI check the Database	PASSED	0.293 s

(S)- Validate PUT method for Users

PASSED	DURATION - 0.589 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:16:16.288 AM // 11:16:16.877 AM /				
To check The PUT method for Users				

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.003 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 1 with valid JSON schema	PASSED	0.132 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 1 row	PASSED	0.067 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 1	PASSED	0.056 s
6	And UsersAPI check the Database	PASSED	0.330 s

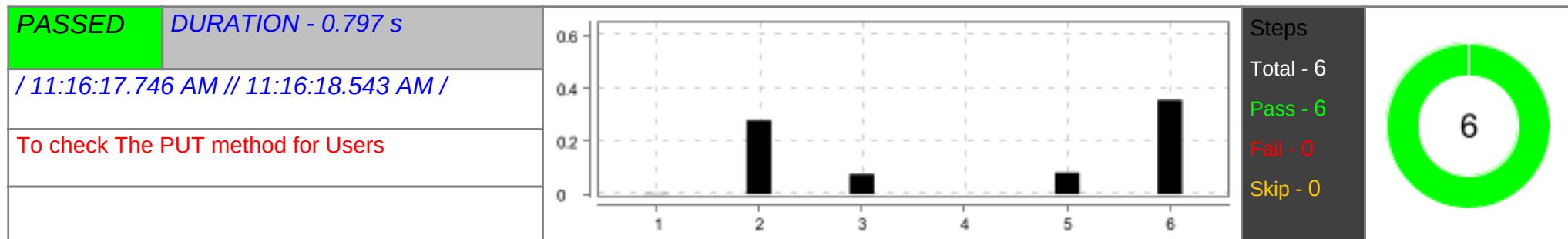
(S)- Validate PUT method for Users

PASSED	DURATION - 0.814 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:16:16.901 AM // 11:16:17.715 AM /				
To check The PUT method for Users				

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 2 with valid JSON schema	PASSED	0.159 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 2 row	PASSED	0.070 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 2	PASSED	0.070 s

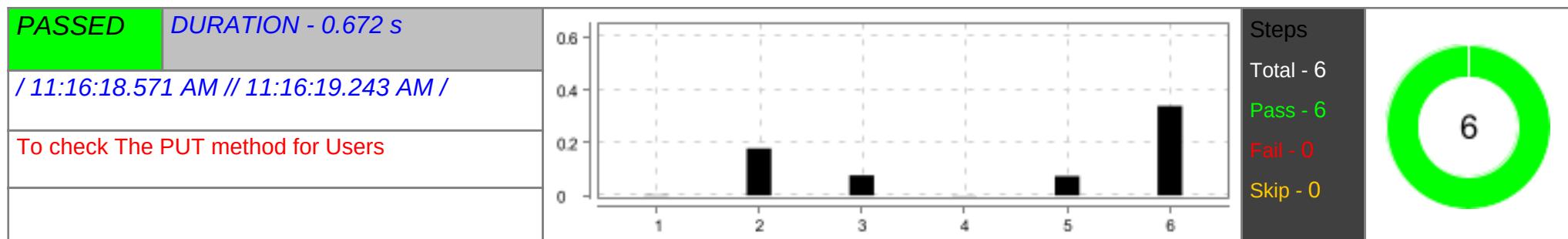
#	Step / Hook Details	Status	Duration
6	And UsersAPI check the Database	PASSED	0.512 s

(S)- Validate PUT method for Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 3 with valid JSON schema	PASSED	0.280 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 3 row	PASSED	0.075 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 3	PASSED	0.080 s
6	And UsersAPI check the Database	PASSED	0.356 s

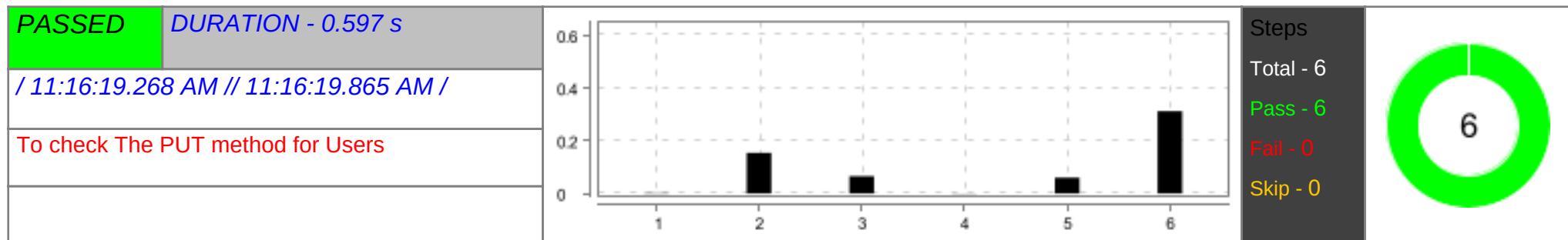
(S)- Validate PUT method for Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 4 with valid JSON schema	PASSED	0.178 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 4 row	PASSED	0.076 s

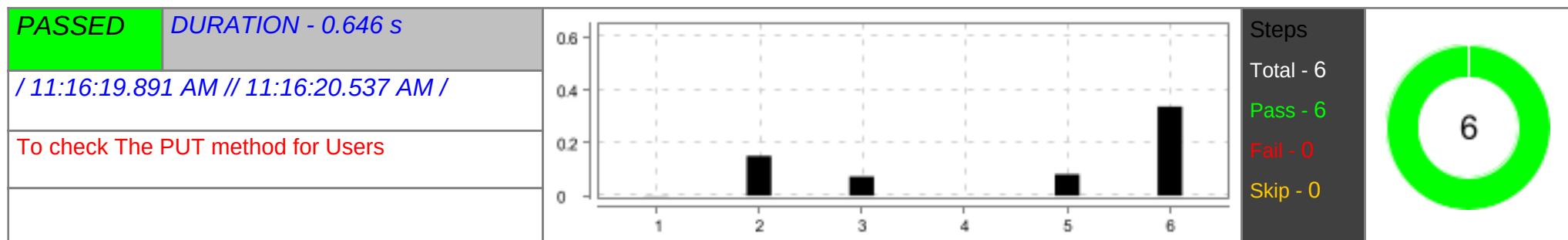
#	Step / Hook Details	Status	Duration
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 4	PASSED	0.073 s
6	And UsersAPI check the Database	PASSED	0.339 s

(S)- Validate PUT method for Users



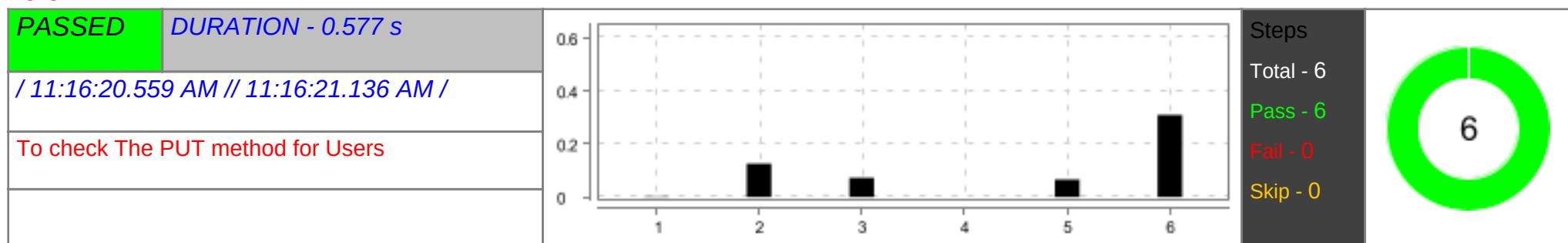
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 5 with valid JSON schema	PASSED	0.154 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 5 row	PASSED	0.066 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 5	PASSED	0.060 s
6	And UsersAPI check the Database	PASSED	0.313 s

(S)- Validate PUT method for Users



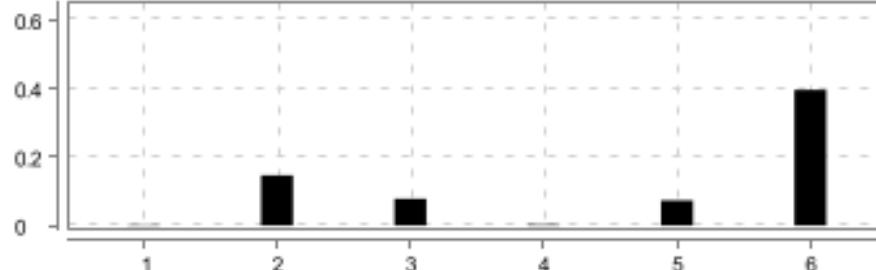
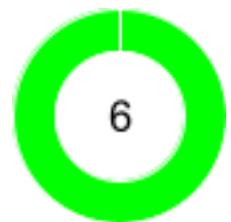
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 6 with valid JSON schema	PASSED	0.151 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 6 row	PASSED	0.072 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 6	PASSED	0.081 s
6	And UsersAPI check the Database	PASSED	0.337 s

(S)- Validate PUT method for Users



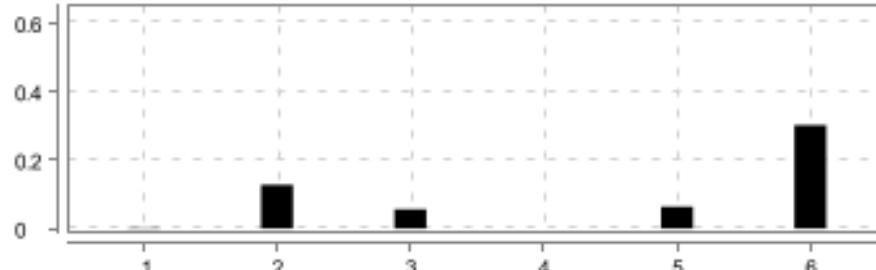
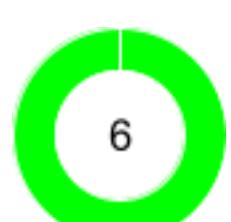
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 7 with valid JSON schema	PASSED	0.126 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 7 row	PASSED	0.072 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 7	PASSED	0.066 s
6	And UsersAPI check the Database	PASSED	0.310 s

(S)- Validate PUT method for Users

PASSED	DURATION - 0.700 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:16:21.157 AM // 11:16:21.857 AM /				
To check The PUT method for Users				

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 8 with valid JSON schema	PASSED	0.147 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 8 row	PASSED	0.078 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.003 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 8	PASSED	0.073 s
6	And UsersAPI check the Database	PASSED	0.396 s

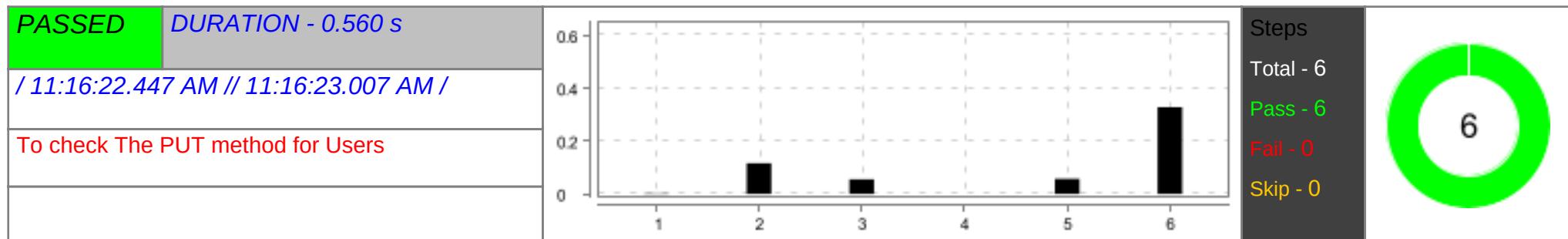
(S)- Validate PUT method for Users

PASSED	DURATION - 0.553 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:16:21.875 AM // 11:16:22.428 AM /				
To check The PUT method for Users				

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 9 with valid JSON schema	PASSED	0.128 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 9 row	PASSED	0.056 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 9	PASSED	0.064 s

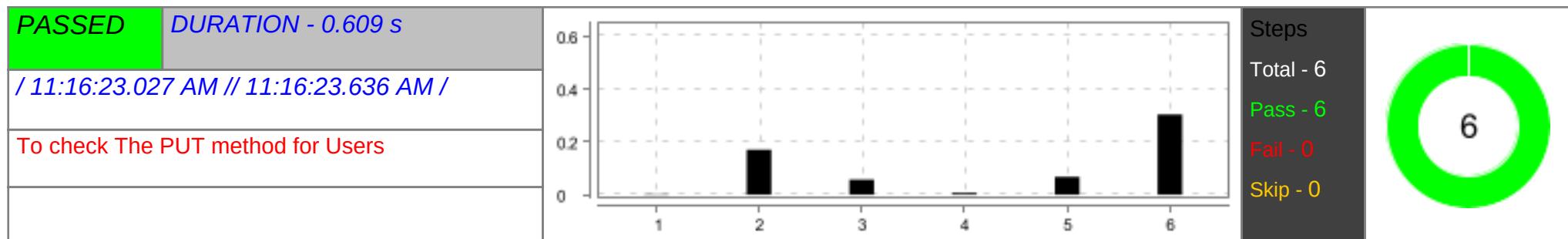
#	Step / Hook Details	Status	Duration
6	And UsersAPI check the Database	PASSED	0.302 s

(S)- Validate PUT method for Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 10 with valid JSON schema	PASSED	0.116 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 10 row	PASSED	0.055 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 10	PASSED	0.057 s
6	And UsersAPI check the Database	PASSED	0.329 s

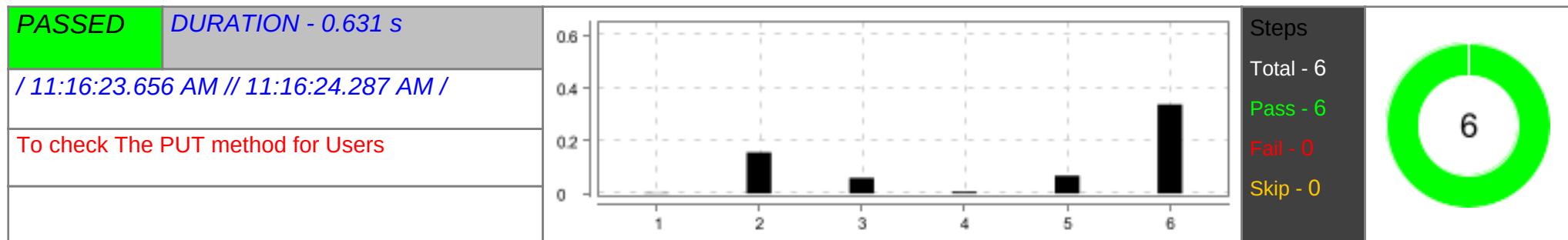
(S)- Validate PUT method for Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 11 with valid JSON schema	PASSED	0.171 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 11 row	PASSED	0.057 s

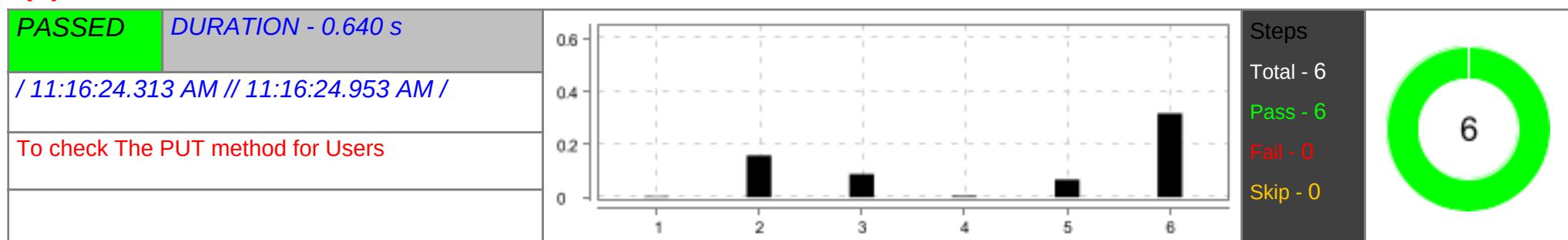
#	Step / Hook Details	Status	Duration
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.007 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 11	PASSED	0.068 s
6	And UsersAPI check the Database	PASSED	0.304 s

(S)- Validate PUT method for Users



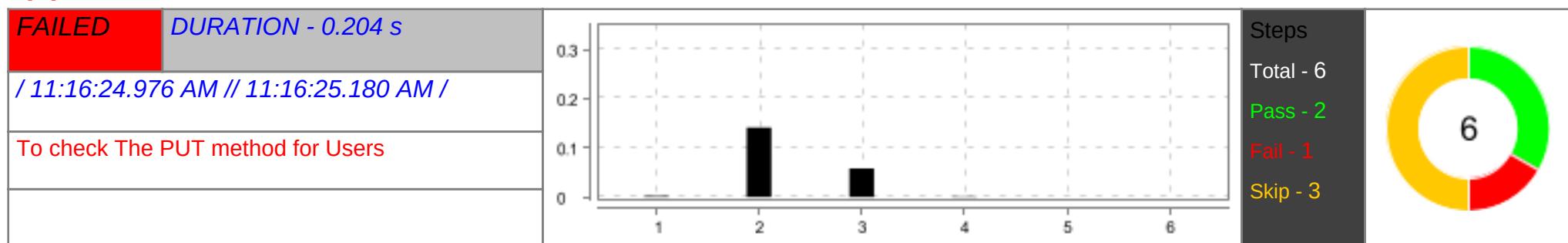
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 12 with valid JSON schema	PASSED	0.156 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 12 row	PASSED	0.059 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.007 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 12	PASSED	0.068 s
6	And UsersAPI check the Database	PASSED	0.338 s

(S)- Validate PUT method for Users



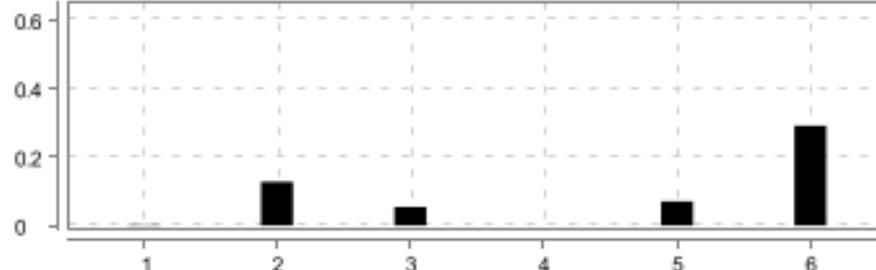
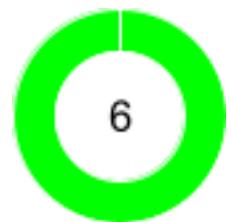
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.003 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 13 with valid JSON schema	PASSED	0.158 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 13 row	PASSED	0.088 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.005 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 13	PASSED	0.067 s
6	And UsersAPI check the Database	PASSED	0.317 s

(S)- Validate PUT method for Users



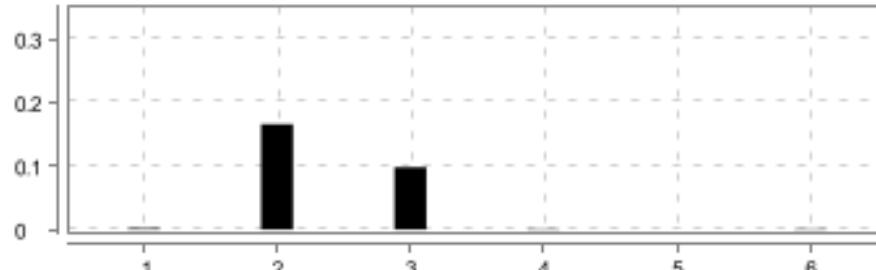
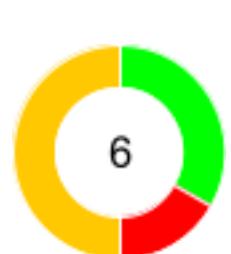
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 14 with valid JSON schema	PASSED	0.141 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 14 row <code>java.lang.AssertionError: expected [400] but found [201]</code> <code>at org.testng.Assert.fail(Assert.java:96)</code> <code>at org.testng.Assert.failNotEquals(Assert.java:776)</code> <code>at org.testng.Assert.assertEqualsImpl(Assert.java:137)</code> <code>at org.testng.Assert.assertEquals(Assert.java:118)</code> <code>at org.testng.Assert.assertEquals(Assert.java:652)</code> <code>at org.testng.Assert.assertEquals(Assert.java:662)</code> <code>at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:196)</code> <code>at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 14 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_PUT.feature:8)</code> * Not displayable characters are replaced by '?'.	FAILED	0.058 s
4	And UsersAPI JSON schema is valid for "PUT"	SKIPPED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 14	SKIPPED	0.000 s
6	And UsersAPI check the Database	SKIPPED	0.000 s

(S)- Validate PUT method for Users

PASSED	DURATION - 0.549 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:16:25.199 AM // 11:16:25.748 AM /				
To check The PUT method for Users				

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 15 with valid JSON schema	PASSED	0.128 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 15 row	PASSED	0.054 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 15	PASSED	0.071 s
6	And UsersAPI check the Database	PASSED	0.292 s

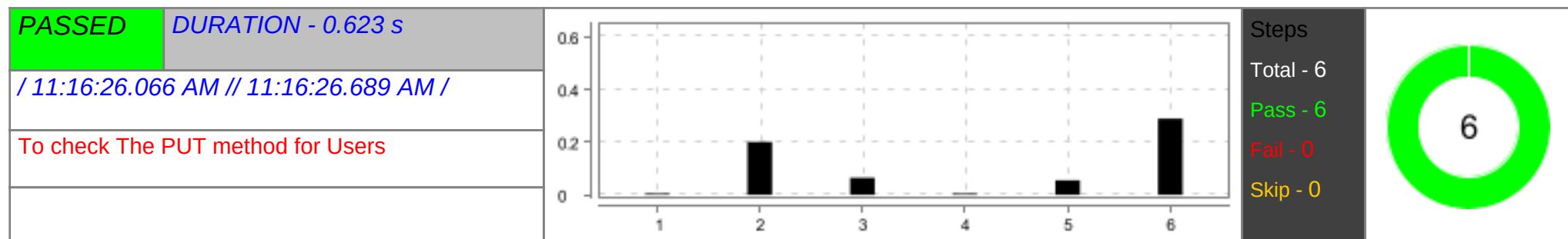
(S)- Validate PUT method for Users

FAILED	DURATION - 0.268 s		Steps Total - 6 Pass - 2 Fail - 1 Skip - 3	
/ 11:16:25.772 AM // 11:16:26.040 AM /				
To check The PUT method for Users				

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 16 with valid JSON schema	PASSED	0.166 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 16 row java.lang.AssertionError: expected [400] but found [201] at org.testng.Assert.failAssert.java:96) at org.testng.Assert.failNotEqualsAssert.java:776)	FAILED	0.098 s

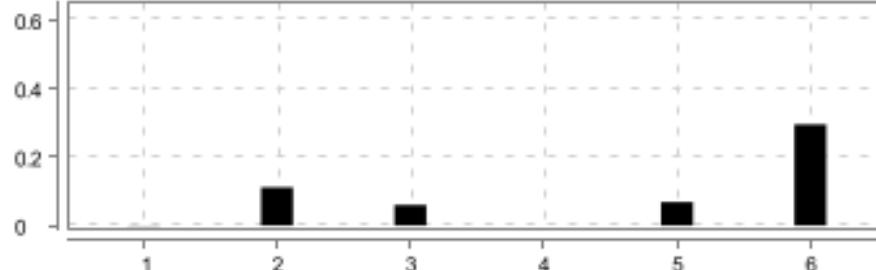
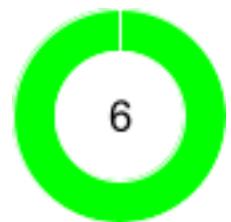
#	Step / Hook Details	Status	Duration
	<pre> at org.testng.Assert.assertEqualsImpl Assert.java:137) at org.testng.Assert.assertEquals Assert.java:118) at org.testng.Assert.assertEquals Assert.java:652) at org.testng.Assert.assertEquals Assert.java:662) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:196) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 16 row(file:///C:/@Ruchira/ Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_PUT.feature:8) * Not displayable characters are replaced by '?'. </pre>		
4	And UsersAPI JSON schema is valid for "PUT"	SKIPPED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 16	SKIPPED	0.000 s
6	And UsersAPI check the Database	SKIPPED	0.001 s

(S)- Validate PUT method for Users



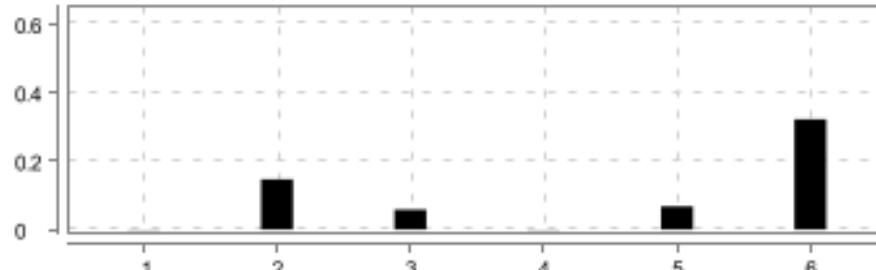
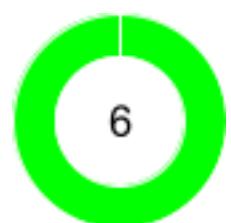
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.005 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 17 with valid JSON schema	PASSED	0.201 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 17 row	PASSED	0.065 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.005 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 17	PASSED	0.055 s
6	And UsersAPI check the Database	PASSED	0.290 s

(S)- Validate PUT method for Users

PASSED	DURATION - 0.540 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
<i>/ 11:16:26.711 AM // 11:16:27.251 AM /</i>				
To check The PUT method for Users				

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 18 with valid JSON schema	PASSED	0.112 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 18 row	PASSED	0.060 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 18	PASSED	0.069 s
6	And UsersAPI check the Database	PASSED	0.295 s

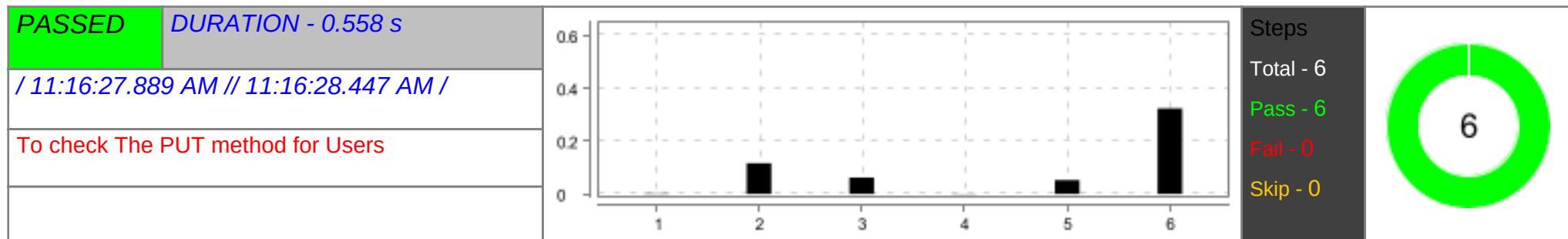
(S)- Validate PUT method for Users

PASSED	DURATION - 0.599 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
<i>/ 11:16:27.268 AM // 11:16:27.867 AM /</i>				
To check The PUT method for Users				

#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 19 with valid JSON schema	PASSED	0.147 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 19 row	PASSED	0.058 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 19	PASSED	0.068 s

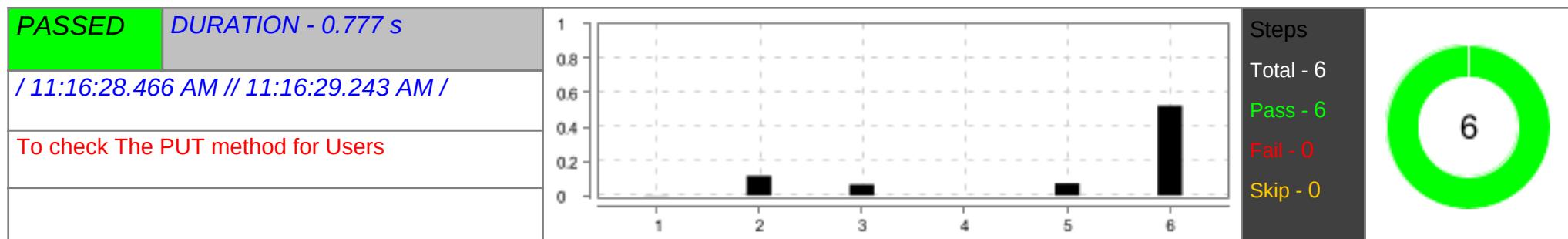
#	Step / Hook Details	Status	Duration
6	And UsersAPI check the Database	PASSED	0.322 s

(S)- Validate PUT method for Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 20 with valid JSON schema	PASSED	0.116 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 20 row	PASSED	0.062 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 20	PASSED	0.052 s
6	And UsersAPI check the Database	PASSED	0.324 s

(S)- Validate PUT method for Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 21 with valid JSON schema	PASSED	0.115 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 21 row	PASSED	0.065 s

#	Step / Hook Details	Status	Duration
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 21	PASSED	0.070 s
6	And UsersAPI check the Database	PASSED	0.523 s

(S)- Validate PUT method for Users

PASSED	DURATION - 0.675 s		Steps Total - 6 Pass - 6 Fail - 0 Skip - 0	
/ 11:16:29.280 AM // 11:16:29.955 AM /				
To check The PUT method for Users				

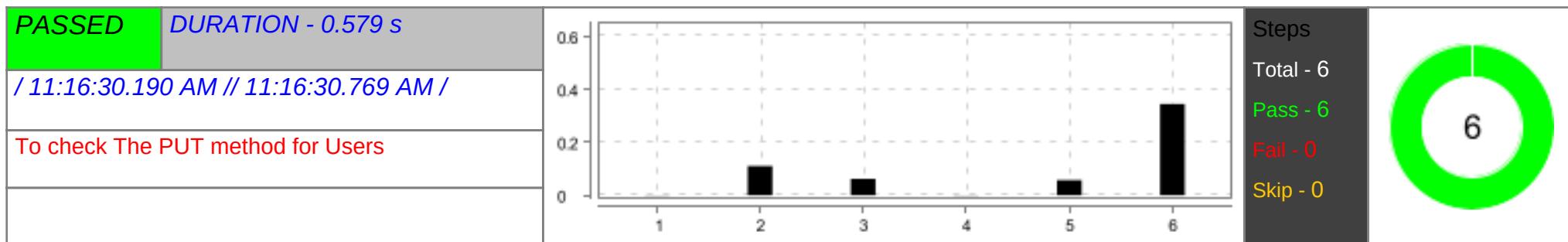
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 22 with valid JSON schema	PASSED	0.190 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 22 row	PASSED	0.067 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 22	PASSED	0.066 s
6	And UsersAPI check the Database	PASSED	0.348 s

(S)- Validate PUT method for Users

FAILED	DURATION - 0.193 s		Steps Total - 6 Pass - 2 Fail - 1 Skip - 3	
/ 11:16:29.973 AM // 11:16:30.166 AM /				
To check The PUT method for Users				

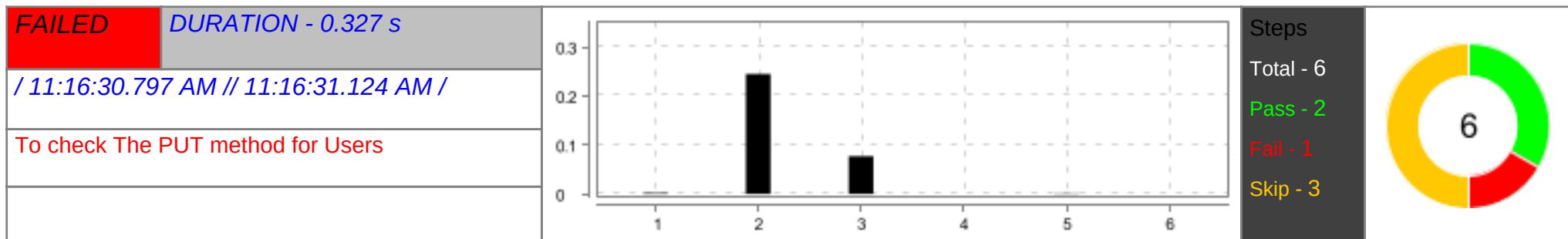
#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 23 with valid JSON schema	PASSED	0.132 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 23 row java.lang.AssertionError: expected [400] but found [201] at org.testng.Assert.failAssert.java:96) at org.testng.Assert.failNotEqualsAssert.java:776) at org.testng.Assert.assertEqualsImplAssert.java:137) at org.testng.Assert.assertEqualsAssert.java:118) at org.testng.Assert.assertEqualsAssert.java:652) at org.testng.Assert.assertEqualsAssert.java:662) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:196) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 23 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_PUT.feature:8) * Not displayable characters are replaced by '?'. 	FAILED	0.058 s
4	And UsersAPI JSON schema is valid for "PUT"	SKIPPED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 23	SKIPPED	0.000 s
6	And UsersAPI check the Database	SKIPPED	0.000 s

(S)- Validate PUT method for Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.001 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 24 with valid JSON schema	PASSED	0.111 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 24 row	PASSED	0.062 s
4	And UsersAPI JSON schema is valid for "PUT"	PASSED	0.001 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location, time_zone,linkedin_url from "InputData_Users_PUT" and 24	PASSED	0.057 s
6	And UsersAPI check the Database	PASSED	0.345 s

(S)- Validate PUT method for Users



#	Step / Hook Details	Status	Duration
1	Given UsersAPI User is on Endpoint: url/Users with valid username and password	PASSED	0.002 s
2	When UsersAPI User sends PUT request on id and request body from "InputData_Users_PUT" and 25 with valid JSON schema	PASSED	0.245 s
3	Then UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 25 row java.lang.AssertionError: expected [400] but found [500] at org.testng.Assert.fail(Assert.java:96) at org.testng.Assert.failNotEquals(Assert.java:776) at org.testng.Assert.assertEqualsImpl(Assert.java:137) at org.testng.Assert.assertEquals(Assert.java:118) at org.testng.Assert.assertEquals(Assert.java:652) at org.testng.Assert.assertEquals(Assert.java:662) at step_definition.Users.UsersAPI_user_receives_status_code_with(Users.java:196) at ?.UsersAPI User validates StatusCode and StatusMessage from "InputData_Users_PUT" sheet and 25 row(file:///C:/@Ruchira/Workspace/LMS_API_HACKATHON2122/src/test/resources/users.feature/Users_PUT.feature:8) * Not displayable characters are replaced by '?'.	FAILED	0.077 s
4	And UsersAPI JSON schema is valid for "PUT"	SKIPPED	0.000 s
5	And UsersAPI User should receive a list of user in JSON body with the fields like user_id,name,phone_number,location,time_zone,linkedin_url from "InputData_Users_PUT" and 25	SKIPPED	0.001 s
6	And UsersAPI check the Database	SKIPPED	0.000 s