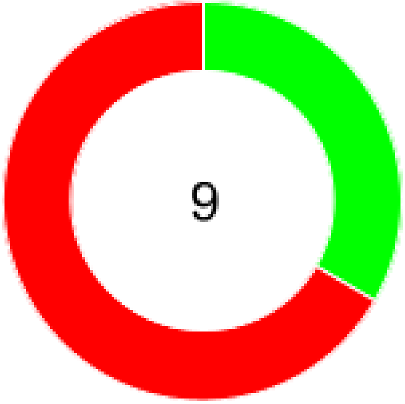
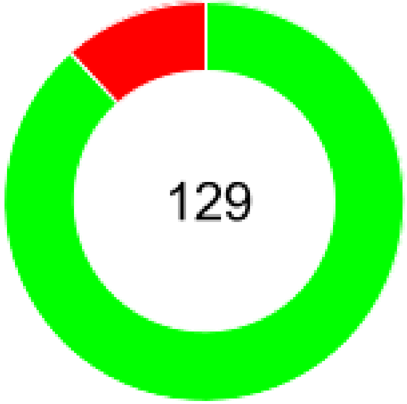
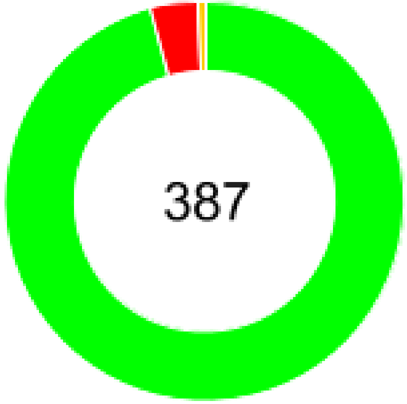
Cucumber Report

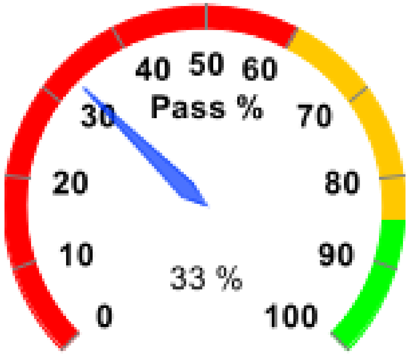
Jun 13, 2025, 1:08:24 PM

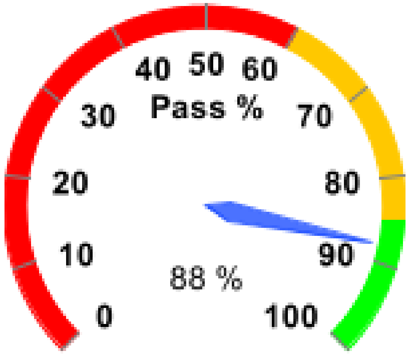
Start : Jun 13, 1:07:15.176 PM	End : Jun 13, 1:08:23.352 PM	Duration : 1 m 8.176 s
Features	*Scenarios*	*Steps*
PASSED - 3 *FAILED - 6* *SKIPPED - 0*	*PASSED - 114* *FAILED - 15* *SKIPPED - 0*	*PASSED - 370* *FAILED - 15* *SKIPPED - 2*

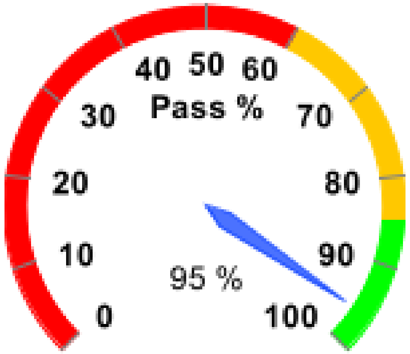








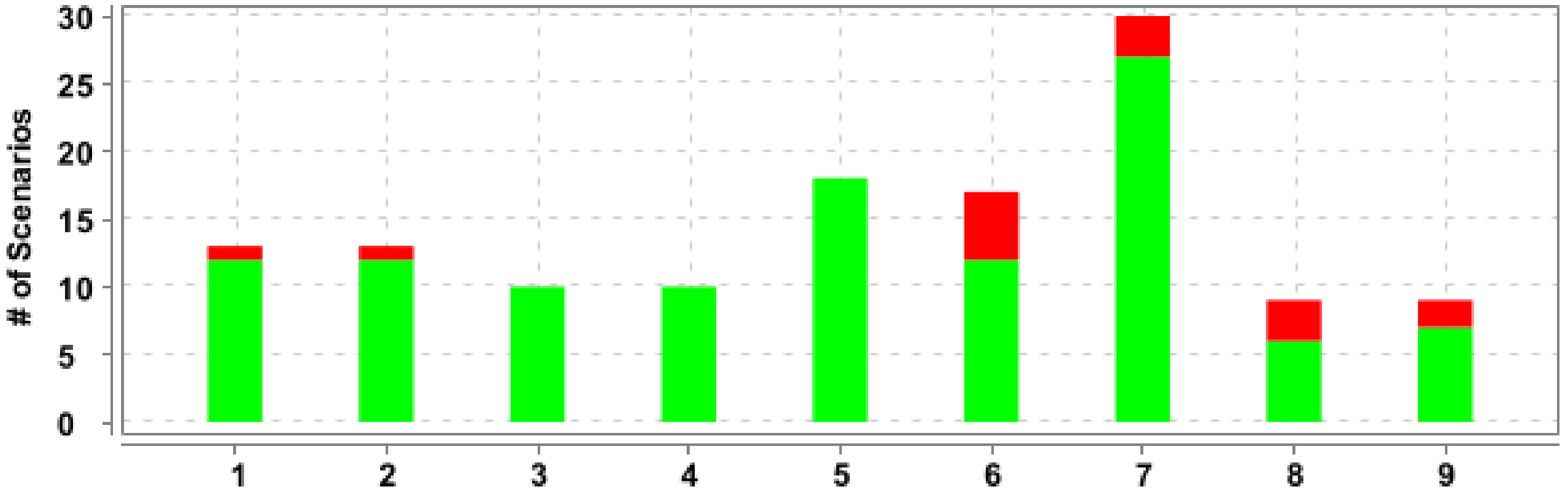




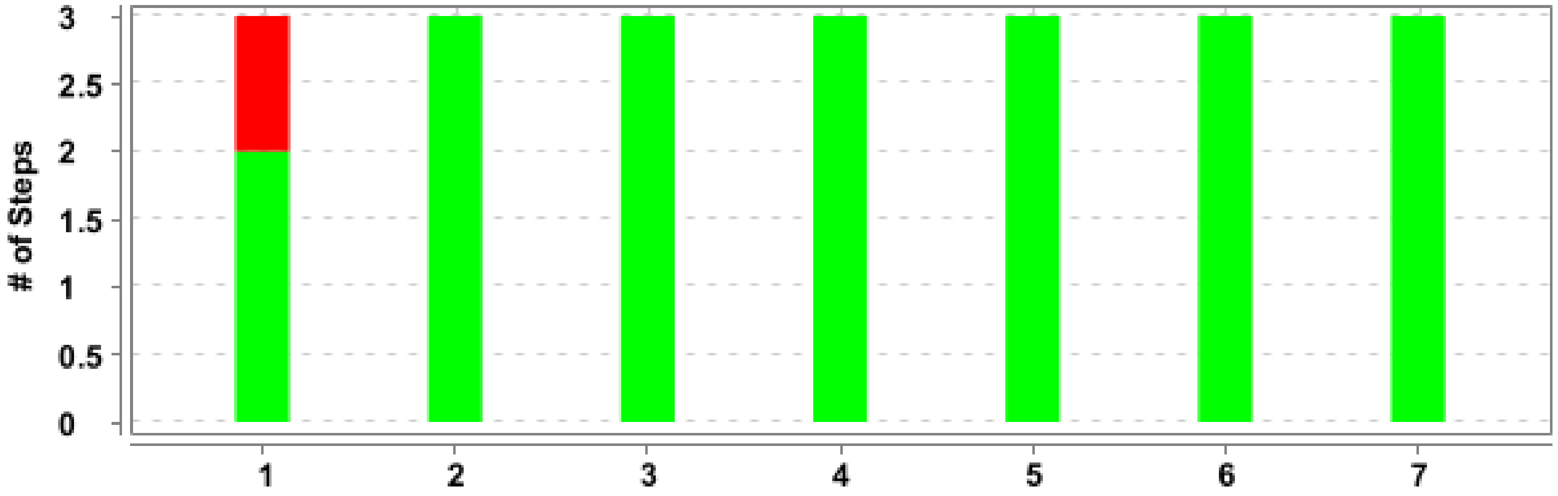
Feature		Scenario				Step			
Name	Duration	T	P	F	S	T	P	F	S
User Deletion API by userFirstName - DELETE Request Scenarios	9.987 s	13	12	1	0	39	38	1	0
User Deletion API - DELETE Request Scenarios	5.254 s	13	12	1	0	39	38	1	0
Get User by FirstName API - GET Request Scenarios	4.409 s	10	10	0	0	30	30	0	0
Get User by ID API - GET Request Scenarios	4.358 s	10	10	0	0	30	30	0	0
Update User Details API - PATCH Request Scenarios	7.556 s	18	18	0	0	54	54	0	0
Add New User API - Negative Scenarios	6.119 s	17	12	5	0	51	44	5	2
Update User Details API - PUT Request Scenarios	13.033 s	30	27	3	0	90	87	3	0
Get All Users API - GET Request Scenarios	10.337 s	9	6	3	0	27	24	3	0
Add New User API - Positive Scenarios	6.980 s	9	7	2	0	27	25	2	0

#	Feature Name	Scenario Name
1	User Deletion API by userFirstName - DELETE Request Scenarios	Verify that the user deletion operation is successful with existing userFirstName
2	User Deletion API - DELETE Request Scenarios	Verify that the user deletion operation is successful with existing userID
3	Add New User API - Negative Scenarios	Add a new user with empty request body
4		Add a new user with only optional fields
5		Add a new user with invalid last name
6		Add a new user with special characters in contact number
7		Add a new user with special characters in zip code
8	Update User Details API - PUT Request Scenarios	UPDATE user details with duplicate contact number
9		UPDATE user details with duplicate email ID
10		UPDATE user details with invalid endpoint

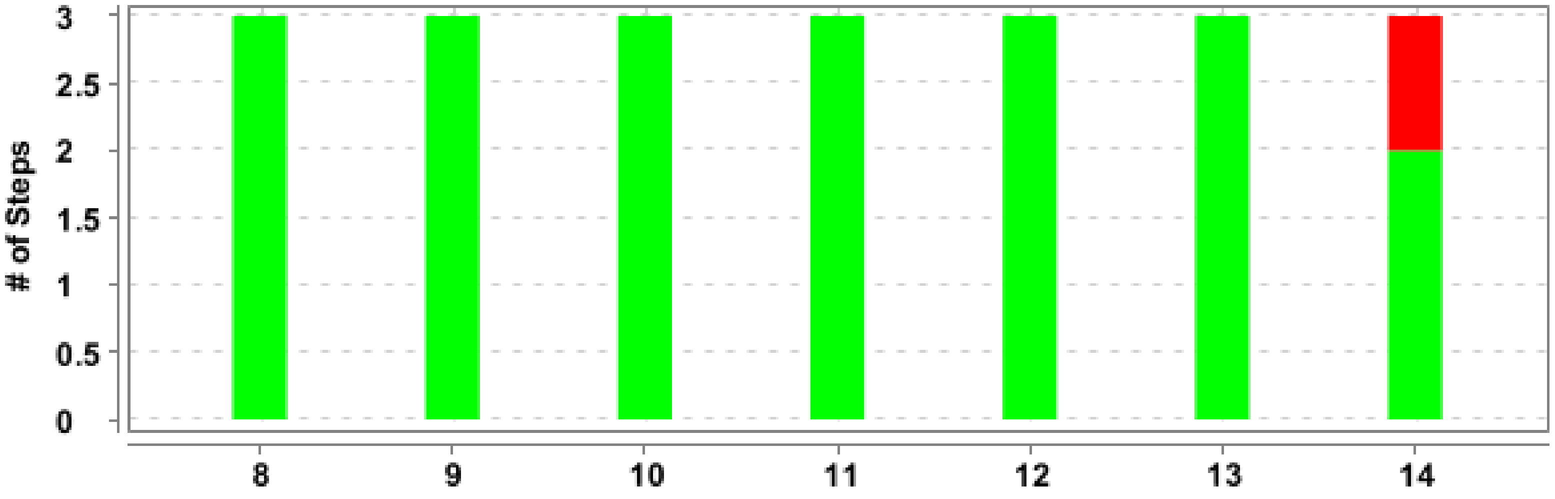
#	Feature Name	Scenario Name
11	Get All Users API - GET Request Scenarios	Validate user_id is integer type in response
12		Validate address_id is integer type in response
13		Get Users with Invalid Base URL
14	Add New User API - Positive Scenarios	Add a new user with only mandatory fields
15		Validate Content-Type is application/json in response



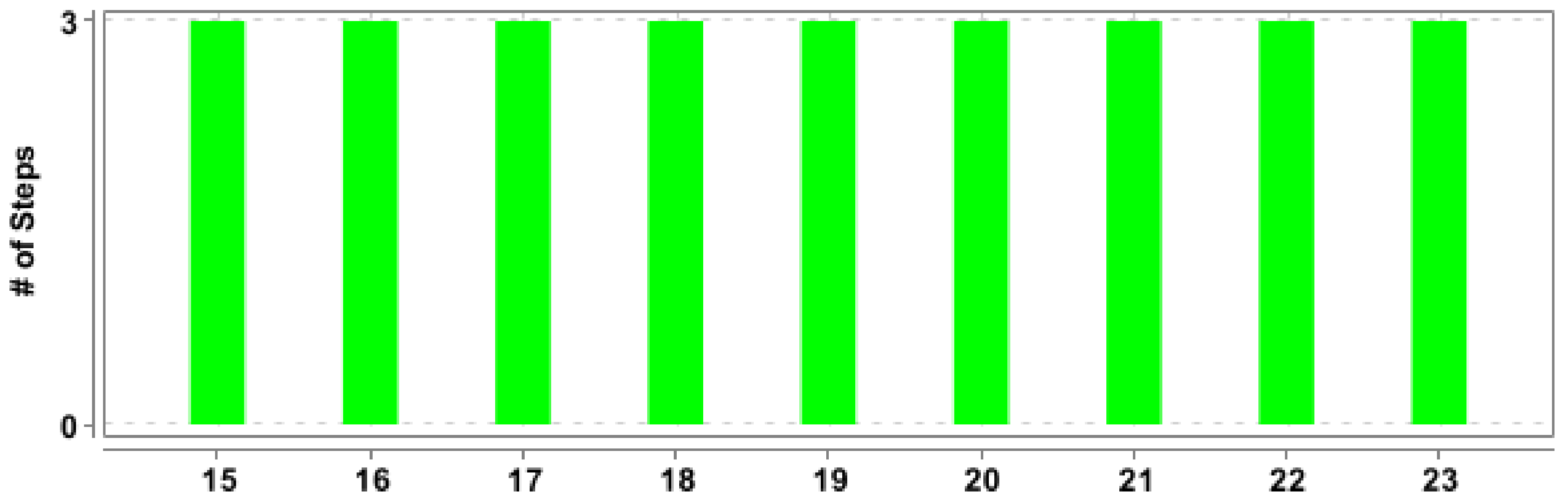
#	Feature Name	T	P	F	S	Duration
1	User Deletion API by userFirstName - DELETE Request Scenarios	13	12	1	0	9.987 s
2	User Deletion API - DELETE Request Scenarios	13	12	1	0	5.254 s
3	Get User by FirstName API - GET Request Scenarios	10	10	0	0	4.409 s
4	Get User by ID API - GET Request Scenarios	10	10	0	0	4.358 s
5	Update User Details API - PATCH Request Scenarios	18	18	0	0	7.556 s
6	Add New User API - Negative Scenarios	17	12	5	0	6.119 s
7	Update User Details API - PUT Request Scenarios	30	27	3	0	13.033 s
8	Get All Users API - GET Request Scenarios	9	6	3	0	10.337 s
9	Add New User API - Positive Scenarios	9	7	2	0	6.980 s



#	Feature Name	Scenario Name	T	P	F	S	Duration
1	User Deletion API by userFirstName - DELETE Request Scenarios	Verify that the user deletion operation is successful with existing userFirstName	3	2	1	0	3.279 s
2		Verify that the user deletion operation again after successful deletion with same userFirstName	3	3	0	0	0.383 s
3		Verify userFirstName in response is deleted with requested userFirstName	3	3	0	0	0.907 s
4		Verify that the user deletion operation with invalid userFirstName (alphanumeric and special characters)	3	3	0	0	0.386 s
5		Verify that the userFirstName deletion operation with only special characters	3	3	0	0	0.388 s
6		Verify that the userFirstName deletion operation with only digits	3	3	0	0	0.400 s
7		Verify that the userFirstName deletion operation with non-existent/invalid userFirstName	3	3	0	0	0.392 s

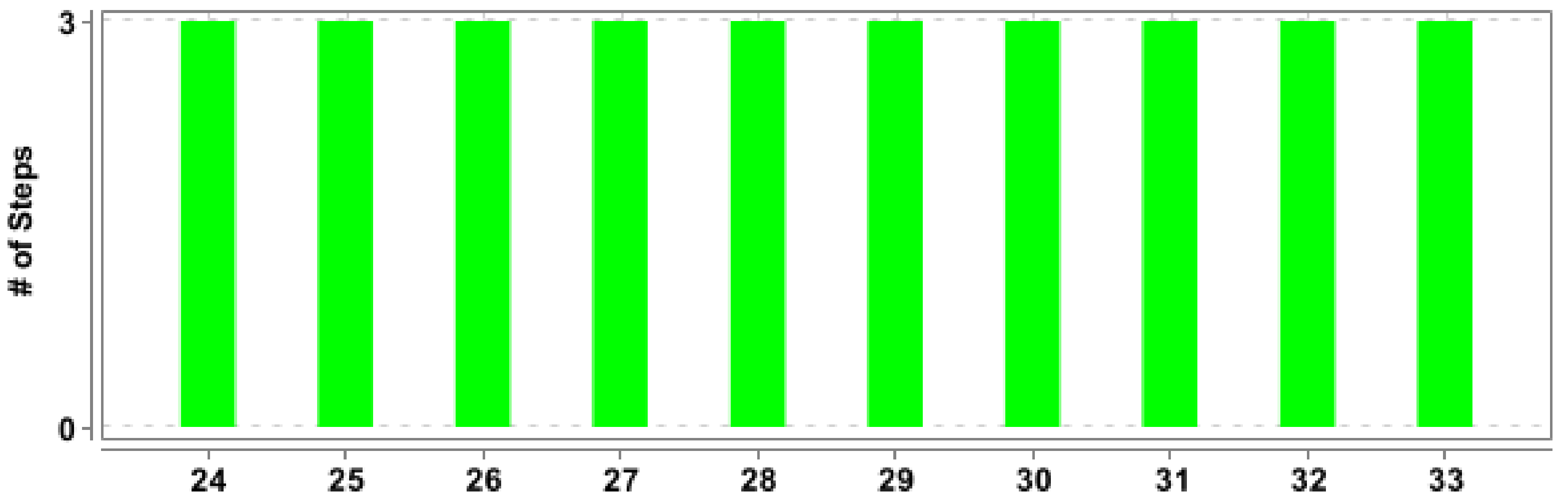


#	Feature Name	Scenario Name	T	P	F	S	Duration
8	User Deletion API by userFirstName - DELETE Request Scenarios	Verify that the userFirstName deletion operation with zero value	3	3	0	0	0.497 s
9		Verify that the userFirstName deletion operation with negative integer value	3	3	0	0	0.473 s
10		Verify that the userFirstName deletion operation with out of range value	3	3	0	0	0.382 s
11		Verify the userFirstName deletion operation with invalid BaseURL	3	3	0	0	0.771 s
12		Verify the userFirstName deletion operation with invalid Endpoint	3	3	0	0	0.835 s
13		Verify the userFirstName deletion operation with empty Endpoint	3	3	0	0	0.744 s
14	User Deletion API - DELETE Request Scenarios	Verify that the user deletion operation is successful with existing userID	3	2	1	0	0.501 s



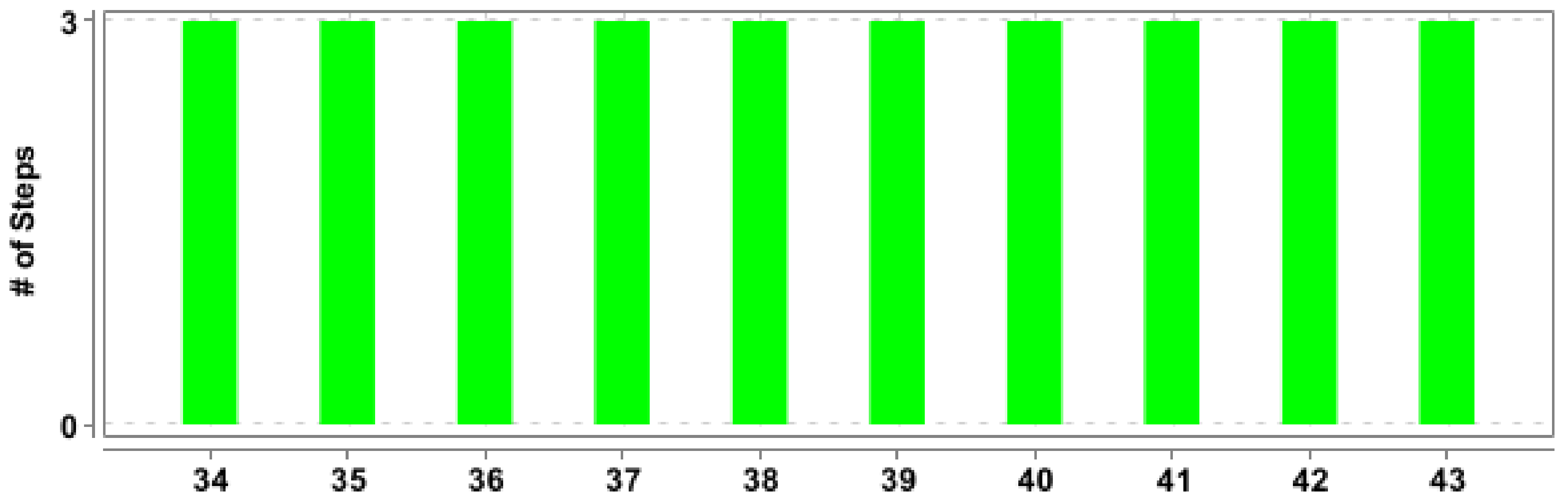
#	Feature Name	Scenario Name	T	P	F	S	Duration
15	User Deletion API - DELETE Request Scenarios	Verify that the user deletion operation again after successful deletion with same userID	3	3	0	0	0.364 s
16		Verify userID in response is deleted with requested userID	3	3	0	0	0.424 s
17		Verify that the user deletion operation with invalid userID (alphanumeric and special characters)	3	3	0	0	0.302 s
18		Verify that the user deletion operation with invalid userID (only special characters)	3	3	0	0	0.365 s
19		Verify that the user deletion operation with non-existent userID	3	3	0	0	0.368 s
20		Verify that the user deletion operation with blank userID	3	3	0	0	0.393 s
21		Verify that the user deletion operation with zero value userID	3	3	0	0	0.502 s
22		Verify that the user deletion operation with negative integer userID	3	3	0	0	0.380 s

#	Feature Name	Scenario Name	T	P	F	S	Duration
23		Verify that the user deletion operation with out of range userID	3	3	0	0	0.366 s

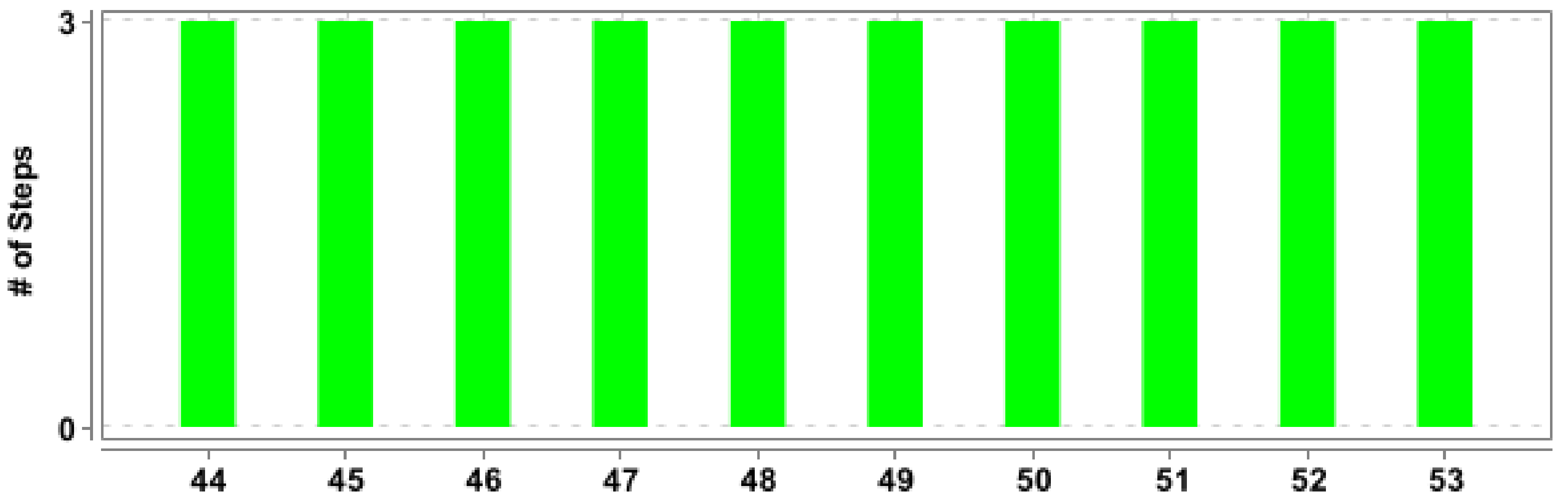


#	Feature Name	Scenario Name	T	P	F	S	Duration
24	User Deletion API - DELETE Request Scenarios	Verify the user deletion operation with invalid BaseURL	3	3	0	0	0.286 s
25		Verify the user deletion operation with invalid Endpoint	3	3	0	0	0.349 s
26		Verify the user deletion operation with empty Endpoint	3	3	0	0	0.449 s
27	Get User by FirstName API - GET Request Scenarios	Get User by valid FirstName	3	3	0	0	0.365 s
28		Validate FirstName in Response Matches Requested FirstName	3	3	0	0	0.424 s
29		Validate Response Body Content-Type is JSON	3	3	0	0	0.409 s
30		Validate Required Fields in Response	3	3	0	0	0.728 s
31		Validate FirstName is String Type	3	3	0	0	0.447 s
32		Validate address_id is integer type in response	3	3	0	0	0.393 s

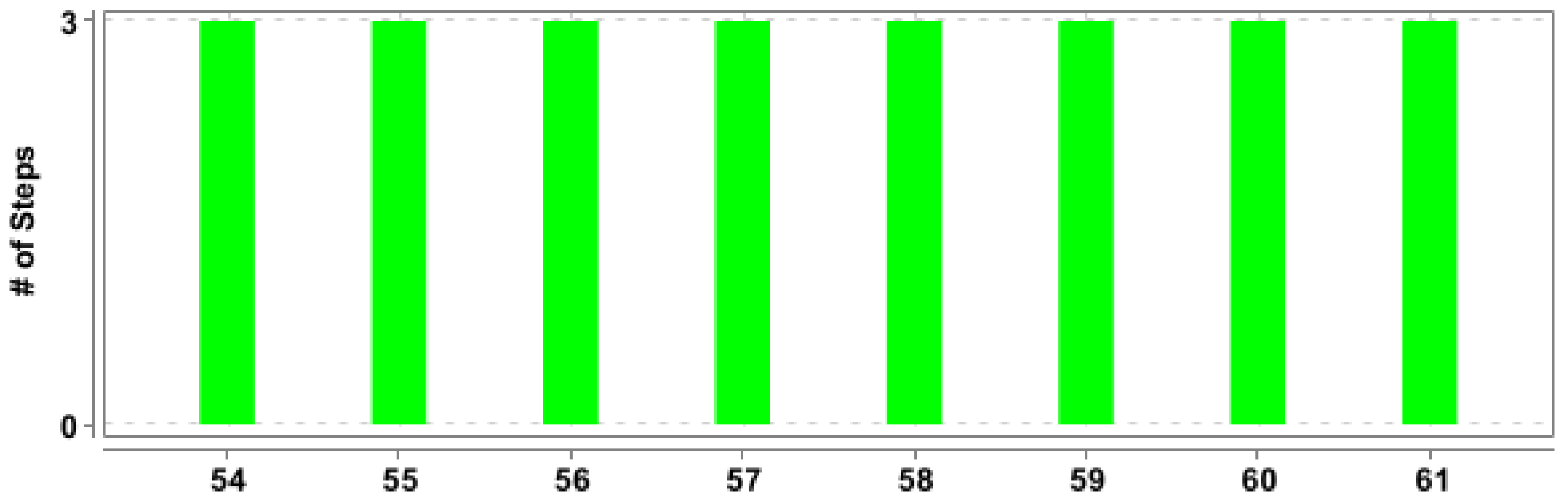
#	Feature Name	Scenario Name	T	P	F	S	Duration
33		Get User with Non-Existing FirstName	3	3	0	0	0.359 s



#	Feature Name	Scenario Name	T	P	F	S	Duration
34	Get User by FirstName API - GET Request Scenarios	Get User with Invalid FirstName Format (e.g. integer instead of string)	3	3	0	0	0.454 s
35		Get User with Invalid Endpoint	3	3	0	0	0.394 s
36		Get User without Authorization	3	3	0	0	0.294 s
37	Get User by ID API - GET Request Scenarios	Get User by Valid ID	3	3	0	0	0.361 s
38		Validate user_id in Response Matches Requested ID	3	3	0	0	0.434 s
39		Validate Response Body Content-Type is JSON	3	3	0	0	0.401 s
40		Validate Required Fields in Response	3	3	0	0	0.624 s
41		Validate user_id is Integer Type	3	3	0	0	0.475 s
42		Validate address_id is integer type in response	3	3	0	0	0.538 s
43		Get User with Non-Existing ID	3	3	0	0	0.384 s

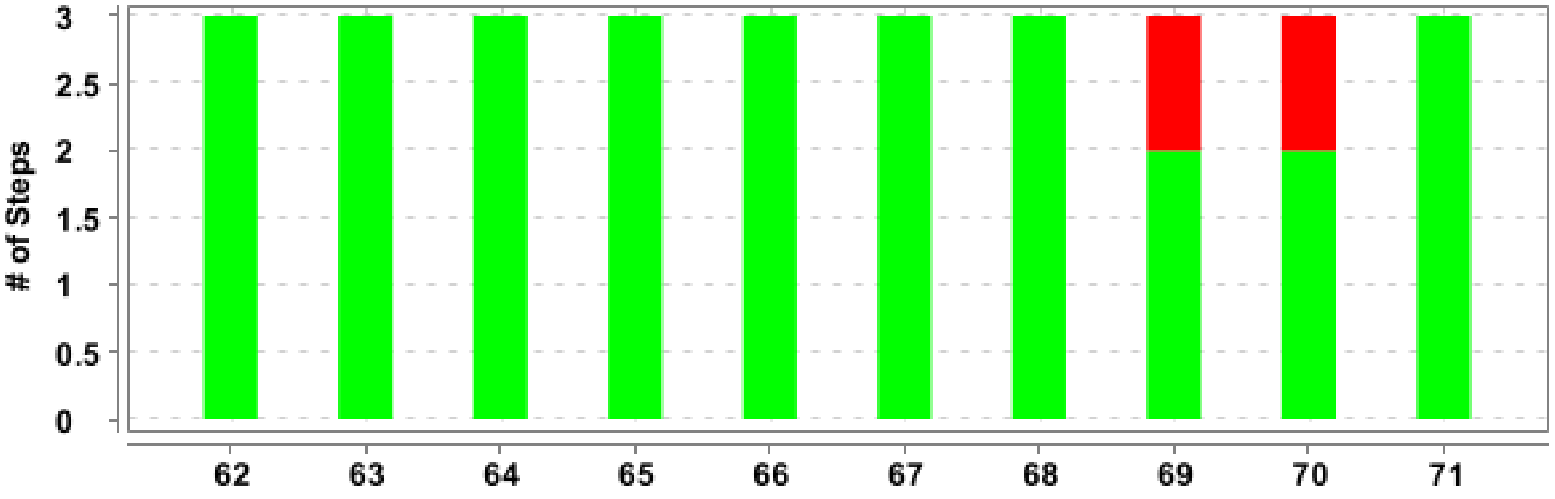


#	Feature Name	Scenario Name	T	P	F	S	Duration
44	Get User by ID API - GET Request Scenarios	Get User with Invalid ID Format (e.g. string instead of integer)	3	3	0	0	0.371 s
45		Get User with Invalid Endpoint	3	3	0	0	0.364 s
46		Get User without Authorization	3	3	0	0	0.287 s
47	Update User Details API - PATCH Request Scenarios	Verify that user is able to update FirstName	3	3	0	0	0.408 s
48		Verify that user is able to update LastName	3	3	0	0	0.501 s
49		Verify that user is able to update contact_number	3	3	0	0	0.516 s
50		Verify that user is able to update email_id	3	3	0	0	0.605 s
51		Verify that user is able to update plotNumber	3	3	0	0	0.356 s
52		Verify that user is able to update Street	3	3	0	0	0.409 s
53		Verify that user is able to update state	3	3	0	0	0.352 s



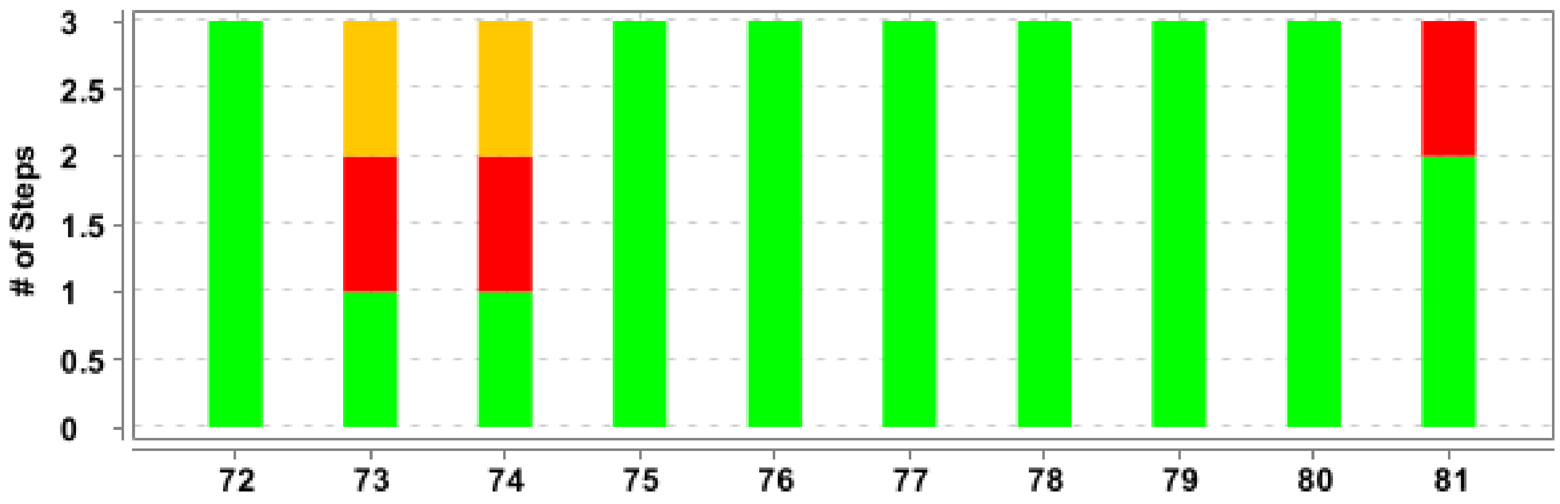
#	Feature Name	Scenario Name	T	P	F	S	Duration
54	Update User Details API - PATCH Request Scenarios	Verify that user is able to update Country	3	3	0	0	0.394 s
55		Verify that user is able to update zipCode	3	3	0	0	0.401 s
56		Verify that user is not able to update contact_number with Characters	3	3	0	0	0.354 s
57		Verify that user is not able to update Invalid email_id	3	3	0	0	0.383 s
58		Verify that user is not able to update Invalid state with Numeric Data	3	3	0	0	0.384 s
59		Verify that user is not able to update Invalid Country with Numeric Data	3	3	0	0	0.433 s
60		Verify that user is not able to update Invalid zipCode with Characters	3	3	0	0	0.357 s

#	Feature Name	Scenario Name	T	P	F	S	Duration
61		Verify that user is not able to update FirstName with invalid BaseURL	3	3	0	0	0.327 s

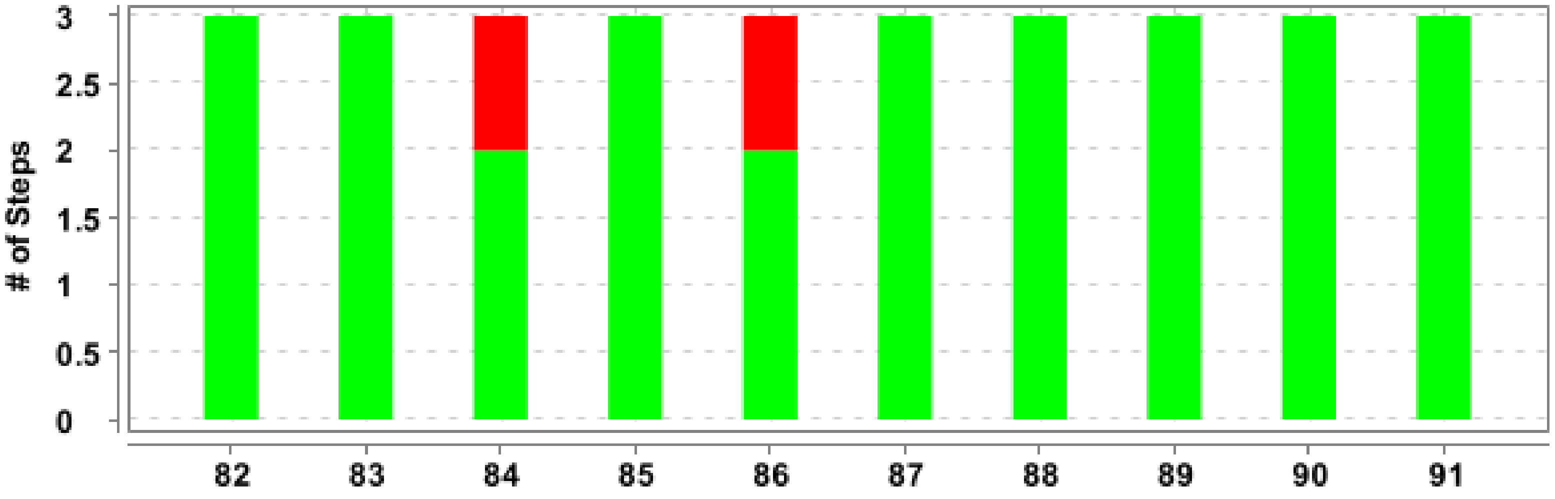


#	Feature Name	Scenario Name	T	P	F	S	Duration
62	Update User Details API - PATCH Request Scenarios	Verify that user is not able to update FirstName with valid BaseURL, Invalid EndPoint	3	3	0	0	0.401 s
63		Verify that user is not able to update FirstName with valid BaseURL, empty EndPoint	3	3	0	0	0.410 s
64		Verify that user is not able to update FirstName with BaseURL, valid Endpoint and No Auth	3	3	0	0	0.362 s
65	Add New User API - Negative Scenarios	Add a new user without Authorization	3	3	0	0	0.298 s
66		Add a new user with invalid BaseURL	3	3	0	0	0.289 s
67		Add a new user with invalid endpoint	3	3	0	0	0.402 s
68		Add a new user with empty endpoint	3	3	0	0	0.432 s
69		Add a new user with empty request body	3	2	1	0	0.376 s
70		Add a new user with only optional fields	3	2	1	0	0.370 s

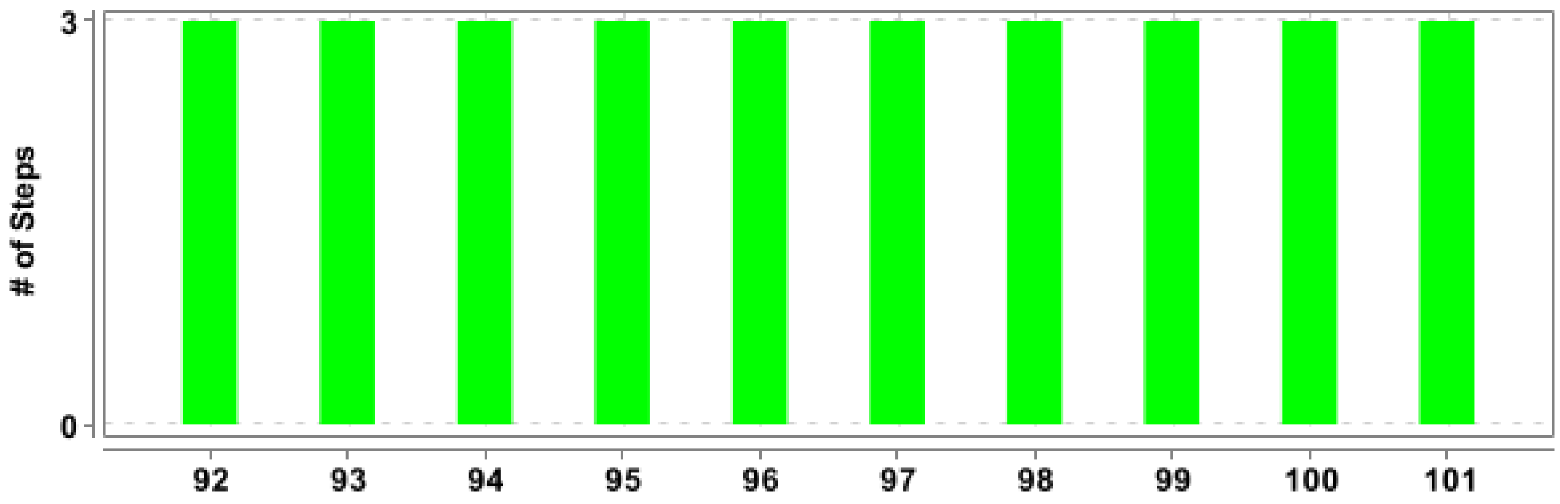
#	Feature Name	Scenario Name	T	P	F	S	Duration
71		Add a new user with existing email and contact	3	3	0	0	0.455 s



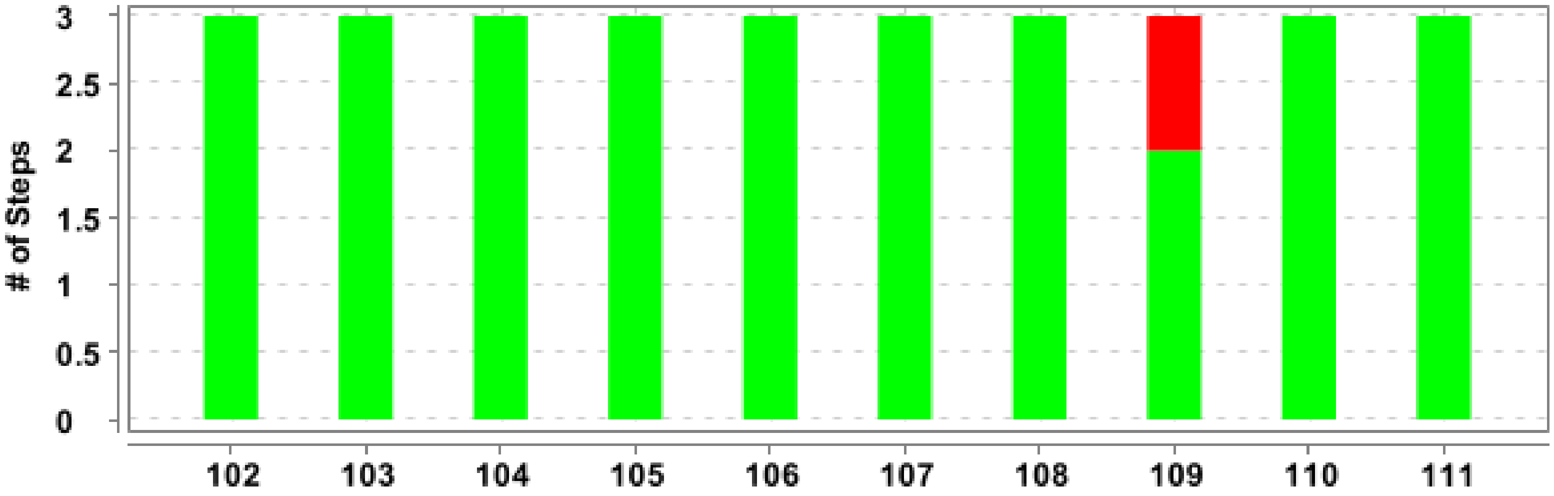
#	Feature Name	Scenario Name	T	P	F	S	Duration
72	Add New User API - Negative Scenarios	Add a new user with invalid first name	3	3	0	0	0.405 s
73		Add a new user with invalid last name	3	1	1	1	0.017 s
74		Add a new user with special characters in contact number	3	1	1	1	0.004 s
75		Add a new user with contact number less than 10 digits	3	3	0	0	0.439 s
76		Add a new user with contact number more than 10 digits	3	3	0	0	0.429 s
77		Add a new user with integer plot number	3	3	0	0	0.349 s
78		Add a new user with special characters in street name	3	3	0	0	0.392 s
79		Add a new user with numeric state value	3	3	0	0	0.457 s
80		Add a new user with numeric country value	3	3	0	0	0.456 s
81		Add a new user with special characters in zip code	3	2	1	0	0.445 s



#	Feature Name	Scenario Name	T	P	F	S	Duration
82	Update User Details API - PUT Request Scenarios	Update user details with valid BaseURL and valid Endpoints	3	3	0	0	0.383 s
83		UPDATE user details with no request body	3	3	0	0	0.409 s
84		UPDATE user details with duplicate contact number	3	2	1	0	0.405 s
85		Update user details with less than ten digits contact number	3	3	0	0	0.432 s
86		UPDATE user details with duplicate email ID	3	2	1	0	0.398 s
87		UPDATE user detail with invalid emailID	3	3	0	0	0.396 s
88		UPDATE user details with invalid plot number	3	3	0	0	0.388 s
89		UPDATE user details with valid new plot number	3	3	0	0	0.386 s
90		UPDATE user details with invalid new street name	3	3	0	0	0.524 s
91		UPDATE user details with valid new street name	3	3	0	0	0.574 s

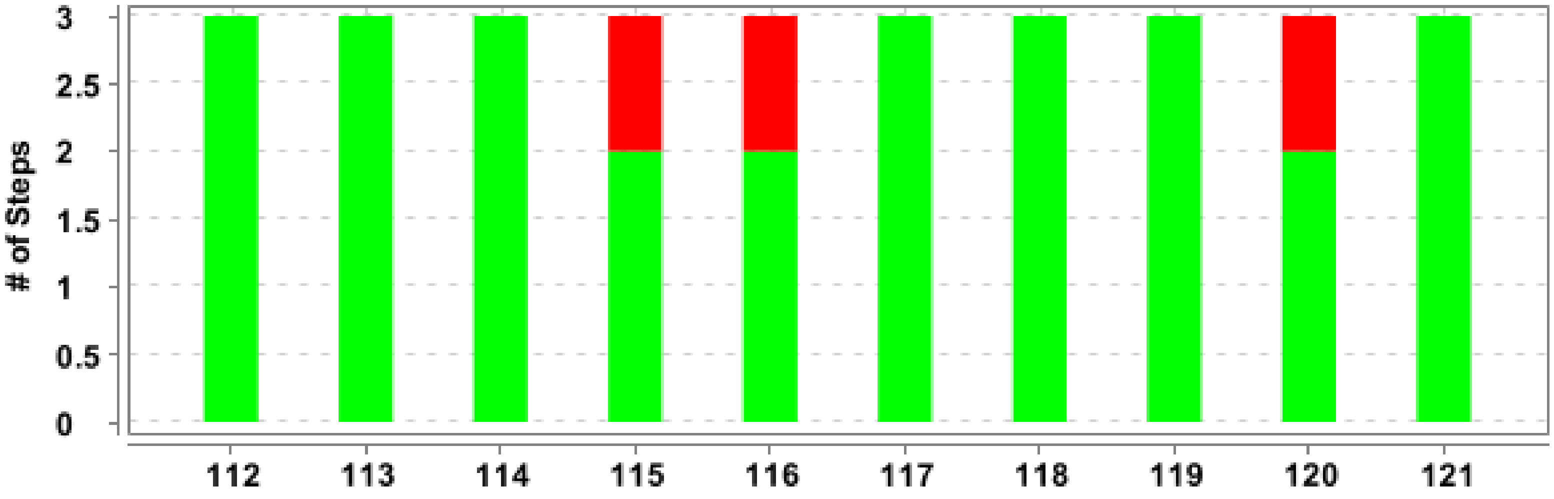


#	Feature Name	Scenario Name	T	P	F	S	Duration
92	Update User Details API - PUT Request Scenarios	UPDATE user details with invalid country name	3	3	0	0	0.382 s
93		UPDATE user details with valid new country name	3	3	0	0	0.376 s
94		UPDATE user details with invalid zipcode	3	3	0	0	0.480 s
95		UPDATE user details with valid new zipcode	3	3	0	0	0.435 s
96		UPDATE user details with invalid firstname	3	3	0	0	0.404 s
97		UPDATE user details with valid firstname	3	3	0	0	0.569 s
98		UPDATE user details with invalid lastname	3	3	0	0	0.469 s
99		UPDATE user details with valid lastname	3	3	0	0	0.486 s
100		UPDATE user details with empty contact number	3	3	0	0	0.455 s
101		UPDATE user details with empty emailID	3	3	0	0	0.466 s



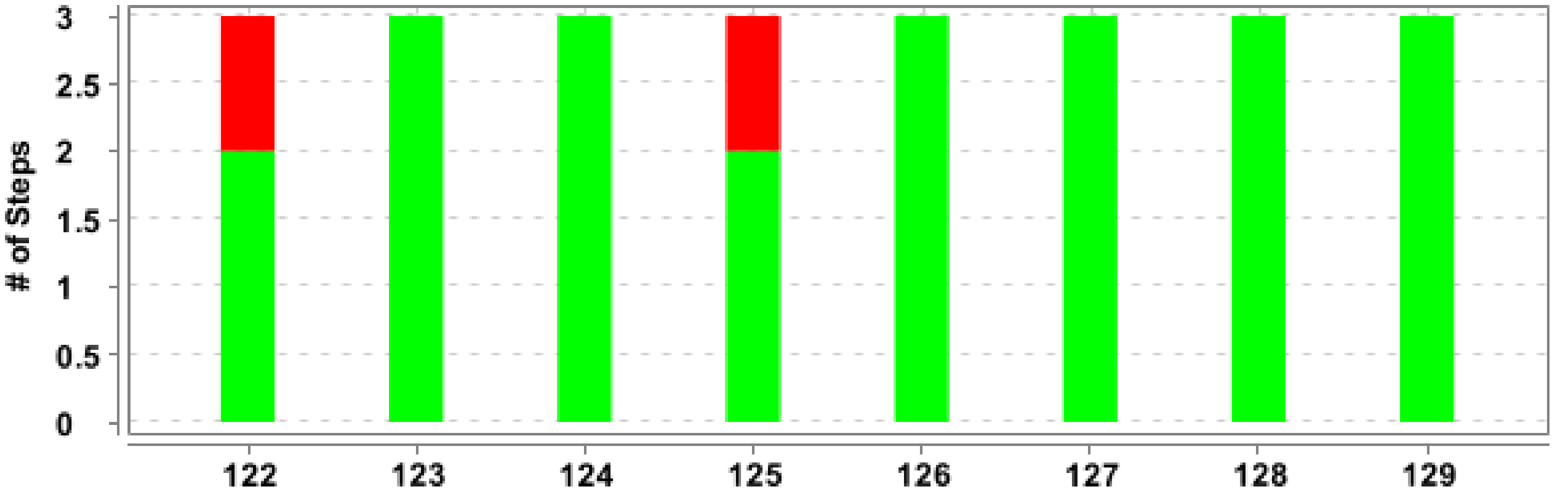
#	Feature Name	Scenario Name	T	P	F	S	Duration
102	Update User Details API - PUT Request Scenarios	UPDATE user details with empty firstname	3	3	0	0	0.478 s
103		UPDATE user details with empty lastname	3	3	0	0	0.403 s
104		UPDATE user details with empty plotnumber	3	3	0	0	0.379 s
105		UPDATE user details with empty street	3	3	0	0	0.385 s
106		UPDATE user details with empty country	3	3	0	0	0.510 s
107		UPDATE user details with empty zipcode	3	3	0	0	0.383 s
108		UPDATE user details with invalid BaseURL	3	3	0	0	0.312 s
109		UPDATE user details with invalid endpoint	3	2	1	0	0.379 s
110		UPDATE user details with empty endpoint	3	3	0	0	0.486 s

#	Feature Name	Scenario Name	T	P	F	S	Duration
11-1		*Add a New User with all fields without Authorization*	*3*	*3*	*0*	*0*	*0.301 s*



#	Feature Name	Scenario Name	T	P	F	S	Duration
11-2	Get All Users API - GET Request Scenarios	Get All Users with Valid Base URL and Endpoint	3	3	0	0	1.988 s
11-3		Validate Response Body Content-Type is JSON	3	3	0	0	1.970 s
11-4		Validate Response Contains Required Fields	3	3	0	0	1.814 s
11-5		Validate user_id is integer type in response	3	2	1	0	1.668 s
11-6		Validate address_id is integer type in response	3	2	1	0	1.553 s
11-7		Get Users with Invalid Endpoint	3	3	0	0	0.396 s
11-8		Get Users with Empty Endpoint	3	3	0	0	0.366 s
11-9		Get USERS without Authorization and receives 401	3	3	0	0	0.282 s
12-0		Get Users with Invalid Base URL	3	2	1	0	0.273 s

#	Feature Name	Scenario Name	T	P	F	S	Duration
12-1	*Add New User API - Positive Scenarios*	*Add a new user with all mandatory and optional fields*	*3*	*3*	*0*	*0*	*0.792 s*



#	Feature Name	Scenario Name	T	P	F	S	Duration
12-2	Add New User API - Positive Scenarios	Add a new user with only mandatory fields	3	2	1	0	0.498 s
12-3		Validate autogenerated user_id in response	3	3	0	0	0.796 s
12-4		Validate autogenerated addressID in response	3	3	0	0	0.731 s
12-5		Validate Content-Type is application/json in response	3	2	1	0	1.072 s
12-6		Validate response status code is 201	3	3	0	0	0.742 s
12-7		Validate response status text is Created	3	3	0	0	0.735 s
12-8		Validate user_id is integer in response	3	3	0	0	0.748 s
12-9		Validate addressID is integer in response	3	3	0	0	0.720 s

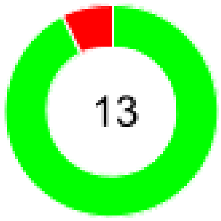
User Deletion API by userFirstName - DELETE Request Scenarios

FAILED

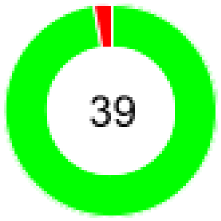
DURATION - 9.987 s

/ 1:07:15.176 PM // 1:07:25.163 PM /

Scenarios
Total - 13
Pass - 12
Fail - 1
Skip - 0



Steps
Total - 39
Pass - 38
Fail - 1
Skip - 0



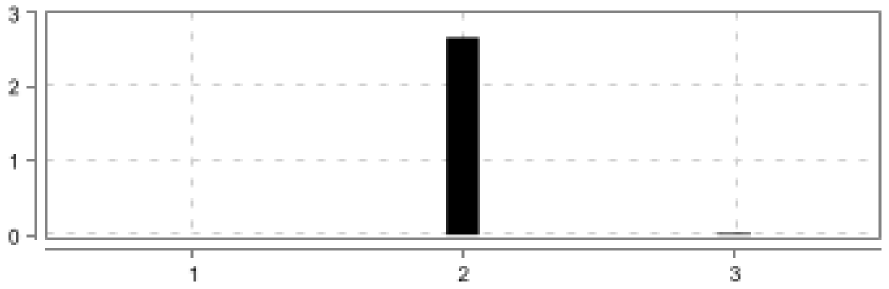
Verify that the user deletion operation is successful with existing userFirstName

FAILED

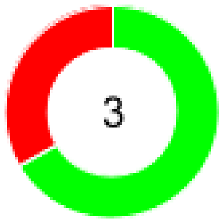
DURATION - 3.279 s

/ 1:07:15.180 PM // 1:07:18.459 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given The userFirstName exists already	PASSED	0.002 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	2.665 s
3	Then User receives 200 OK status with success message	FAILED	0.023 s

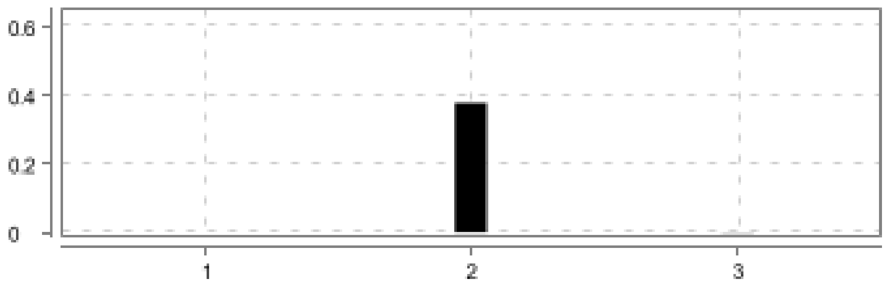
```
java.lang.AssertionError: expected [200] but found [404]
    at org.testng.Assert.fail(Assert.java:111)
    at org.testng.Assert.failNotEquals(Assert.java:1590)
    at org.testng.Assert.assertEqualsImpl(Assert.java:150)
    at org.testng.Assert.assertEquals(Assert.java:132)
    at org.testng.Assert.assertEquals(Assert.java:1431)
    at org.testng.Assert.assertEquals(Assert.java:1395)
    at org.testng.Assert.assertEquals(Assert.java:1441)
    at StepDefinition.Delete_UserFirstName.user_receives_200_ok_status(Delete_UserFirstName.java:32)
    at ?.User receives 200 OK status with success message(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/
Features/Delete_UserFirstname.feature:6)
* Not displayable characters are replaced by '?'.
```

Verify that the user deletion operation again after successful deletion with same userFirstName

PASSED | *DURATION - 0.383 s*

/ 1:07:18.484 PM // 1:07:18.867 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



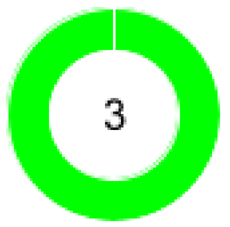
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



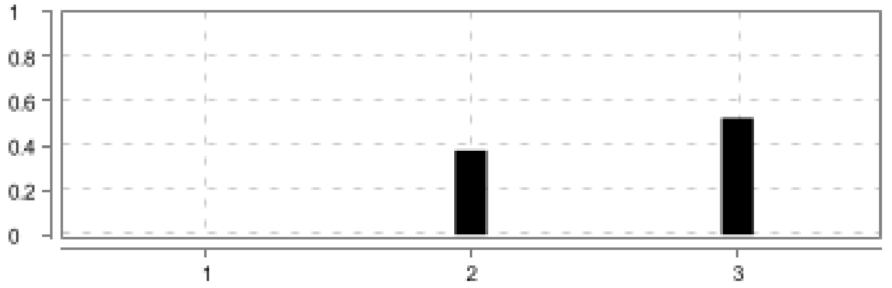
#	Step / Hook Details	Status	Duration
1	Given The userFirstName is already deleted	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.378 s
3	Then User receives 404 Not Found error with message	PASSED	0.001 s

Verify userFirstName in response is deleted with requested userFirstName

PASSED | *DURATION - 0.907 s*

/ 1:07:18.881 PM // 1:07:19.788 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



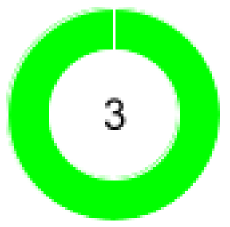
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



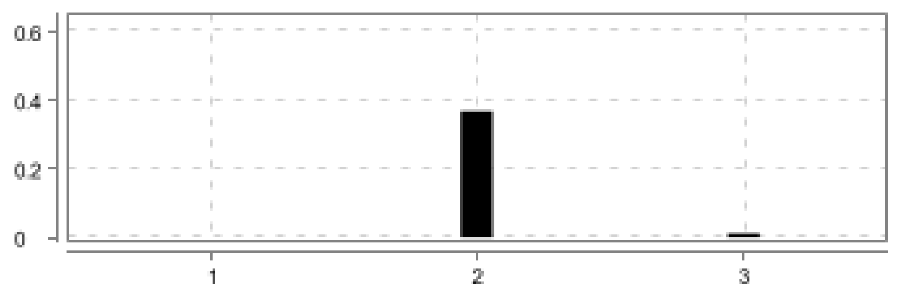
#	Step / Hook Details	Status	Duration
1	Given The userFirstName is already deleted	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.379 s
3	Then Response body contains the same userFirstName as requested	PASSED	0.524 s

Verify that the user deletion operation with invalid userFirstName (alphanumeric and special characters)

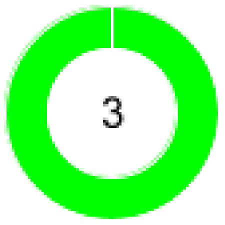
PASSED *DURATION - 0.386 s*

/ 1:07:19.797 PM // 1:07:20.183 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



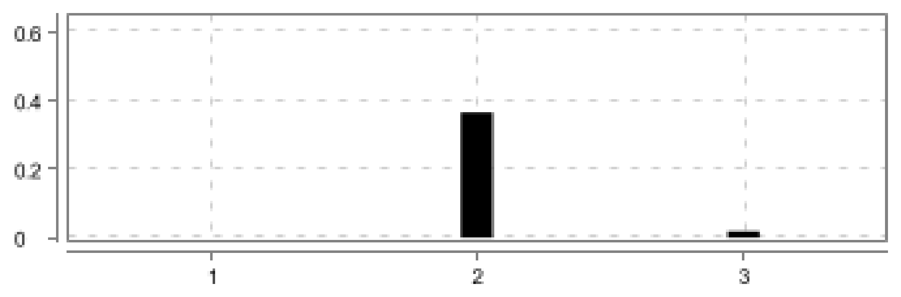
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with combination of number, alphabet and special characters	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.369 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.011 s

Verify that the userFirstName deletion operation with only special characters

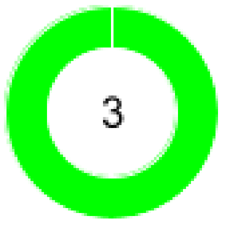
PASSED *DURATION - 0.388 s*

/ 1:07:20.190 PM // 1:07:20.578 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



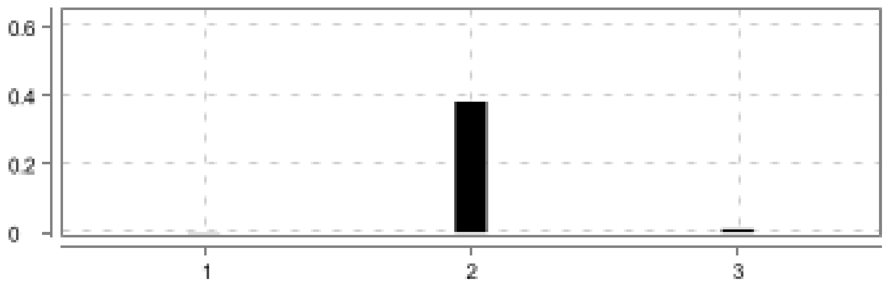
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with only special characters	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.366 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.018 s

Verify that the userFirstName deletion operation with only digits

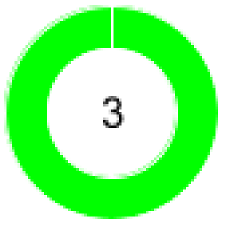
PASSED *DURATION - 0.400 s*

/ 1:07:20.594 PM // 1:07:20.994 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



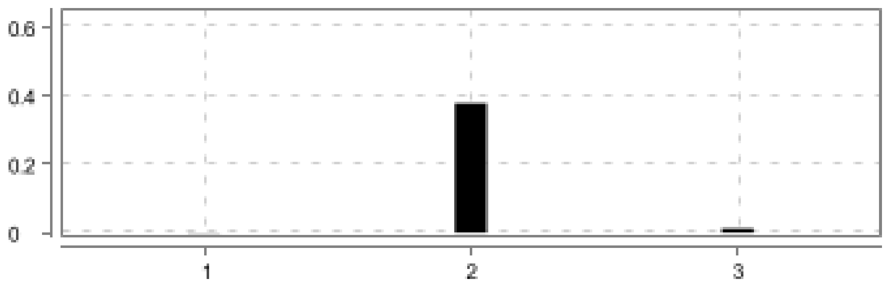
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with only 5 digits	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.382 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.010 s

Verify that the userFirstName deletion operation with non-existent/invalid userFirstName

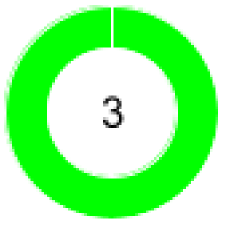
PASSED *DURATION - 0.392 s*

/ 1:07:20.999 PM // 1:07:21.391 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



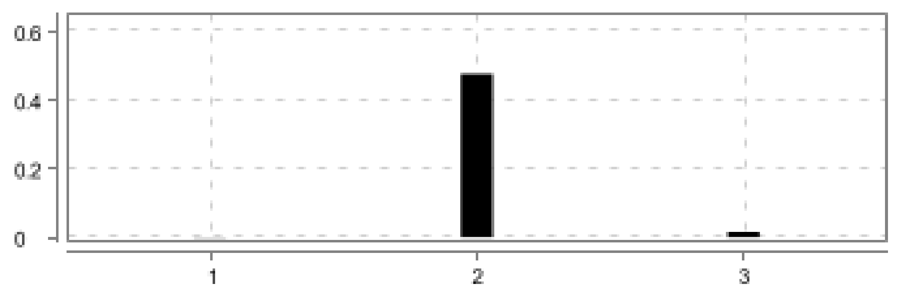
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with non-existent userFirstName	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.377 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.011 s

Verify that the userFirstName deletion operation with zero value

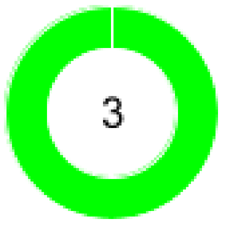
PASSED *DURATION - 0.497 s*

/ 1:07:21.403 PM // 1:07:21.900 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



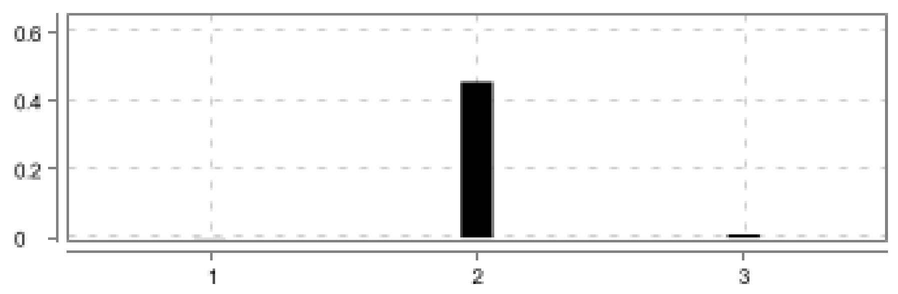
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with zero value	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.477 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.016 s

Verify that the userFirstName deletion operation with negative integer value

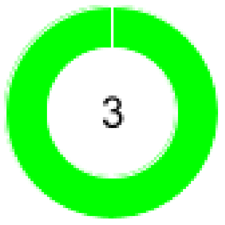
PASSED *DURATION - 0.473 s*

/ 1:07:21.914 PM // 1:07:22.387 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



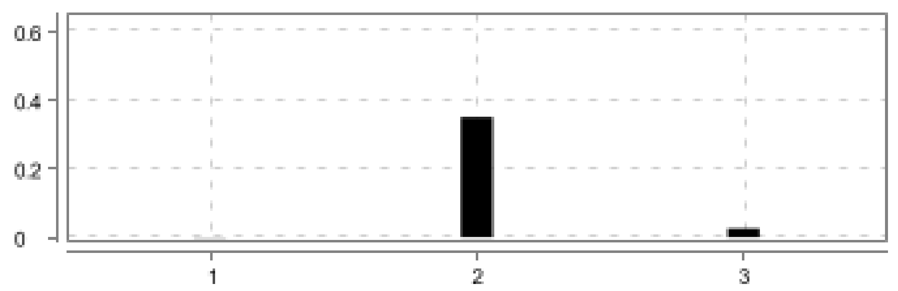
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with negative integer value	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.455 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.009 s

Verify that the userFirstName deletion operation with out of range value

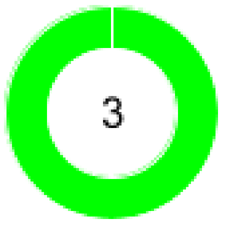
PASSED *DURATION - 0.382 s*

/ 1:07:22.392 PM // 1:07:22.774 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



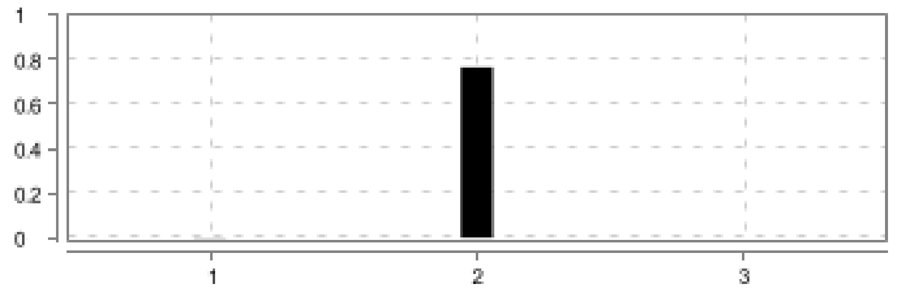
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with out of range value	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.351 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.027 s

Verify the userFirstName deletion operation with invalid BaseURL

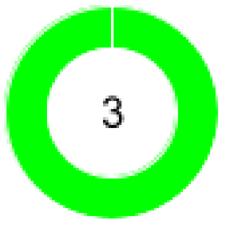
PASSED *DURATION - 0.771 s*

/ 1:07:22.786 PM // 1:07:23.557 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



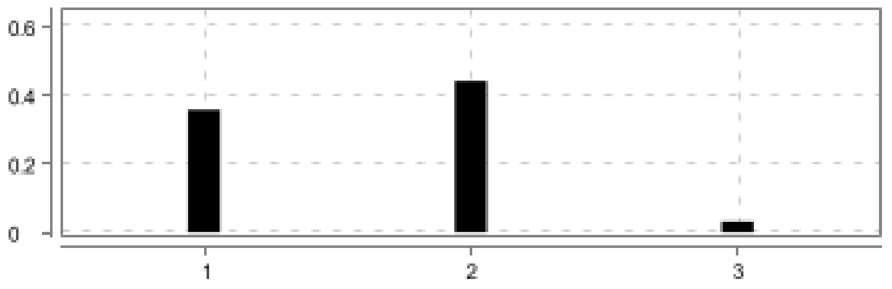
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with invalid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.765 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.000 s

Verify the userFirstName deletion operation with invalid Endpoint

PASSED *DURATION - 0.835 s*

/ 1:07:23.562 PM // 1:07:24.397 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



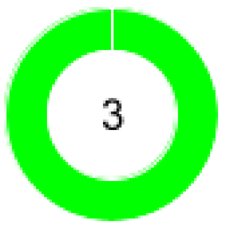
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



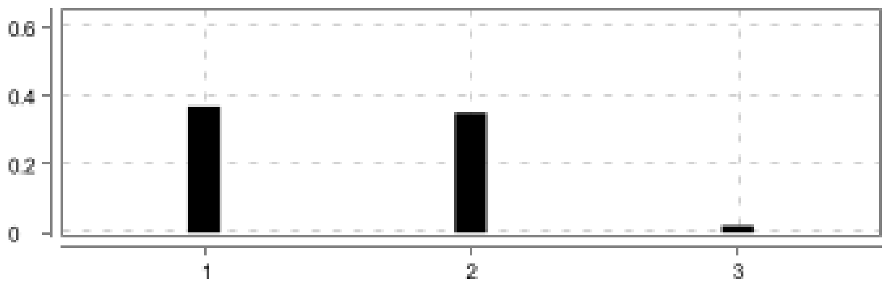
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with valid BaseURL and invalid Endpoint	PASSED	0.357 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.442 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.032 s

Verify the userFirstName deletion operation with empty Endpoint

PASSED *DURATION - 0.744 s*

/ 1:07:24.419 PM // 1:07:25.163 PM /

User Deletion API by userFirstName - DELETE Request Scenarios



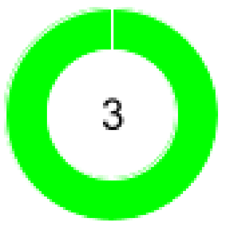
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



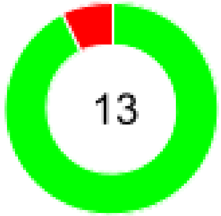
#	Step / Hook Details	Status	Duration
1	Given User enters userFirstName with valid BaseURL and empty Endpoint	PASSED	0.368 s
2	When User sends DELETE request to delete the respective userFirstName	PASSED	0.350 s
3	Then User receives 404 Not Found error with userFirstName message	PASSED	0.020 s

User Deletion API - DELETE Request Scenarios

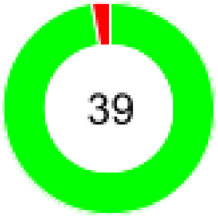
FAILED *DURATION - 5.254 s*

/ 1:07:25.190 PM // 1:07:30.444 PM /

Scenarios
Total - 13
Pass - 12
Fail - 1
Skip - 0



Steps
Total - 39
Pass - 38
Fail - 1
Skip - 0

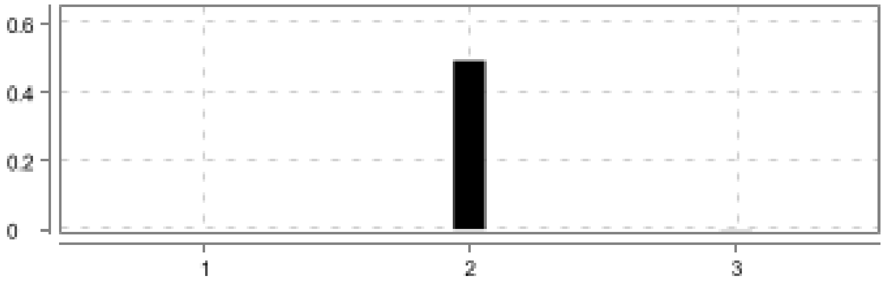


Verify that the user deletion operation is successful with existing userID

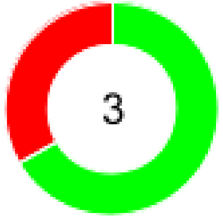
FAILED *DURATION - 0.501 s*

/ 1:07:25.190 PM // 1:07:25.691 PM /

User Deletion API - DELETE Request Scenarios



Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



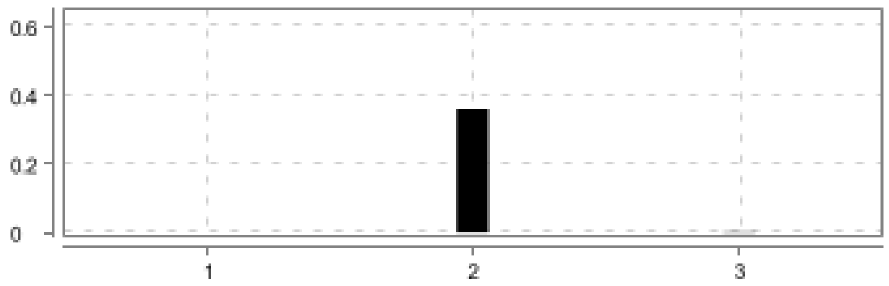
#	Step / Hook Details	Status	Duration
1	Given The userID exists already	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.492 s
3	Then User receives 200 OK status with the success message	FAILED	0.001 s
	java.lang.AssertionError: expected [200] but found [404] at org.testng.Assert.fail(Assert.java:111) at org.testng.Assert.failNotEquals(Assert.java:1590) at org.testng.Assert.assertEqualsImpl(Assert.java:150) at org.testng.Assert.assertEquals(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:1431) at org.testng.Assert.assertEquals(Assert.java:1395) at org.testng.Assert.assertEquals(Assert.java:1441) at StepDefinition.Delete_UserId.user_receives_200_ok_status(Delete_UserId.java:38) at ?.User receives 200 OK status with the success message(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/Delete_Userid.feature:6) * Not displayable characters are replaced by '?'.		

Verify that the user deletion operation again after successful deletion with same userID

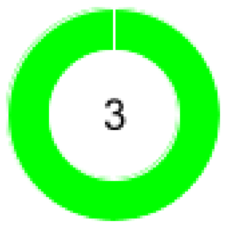
PASSED *DURATION - 0.364 s*

/ 1:07:25.697 PM // 1:07:26.061 PM /

User Deletion API - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



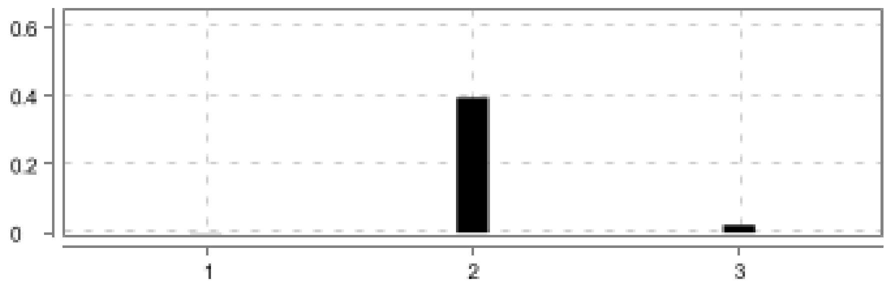
#	Step / Hook Details	Status	Duration
1	Given The userID is already deleted	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.359 s
3	Then User receives 404 Not Found error with the message	PASSED	0.002 s

Verify userID in response is deleted with requested userID

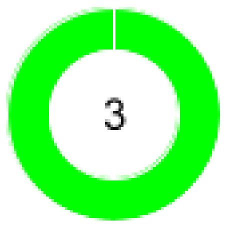
PASSED *DURATION - 0.424 s*

/ 1:07:26.084 PM // 1:07:26.508 PM /

User Deletion API - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



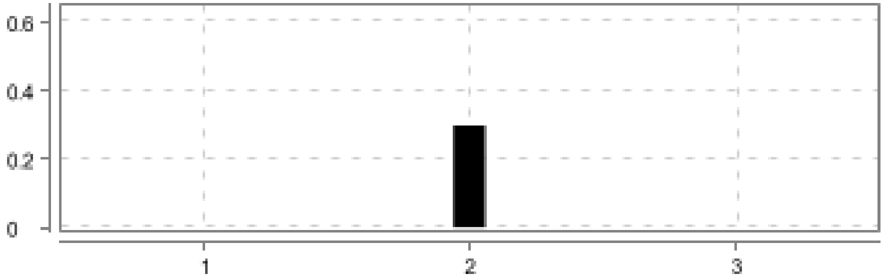
#	Step / Hook Details	Status	Duration
1	Given The userID is already deleted	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.395 s
3	Then Response body contains the same user_id as requested	PASSED	0.022 s

Verify that the user deletion operation with invalid userID (alphanumeric and special characters)

PASSED *DURATION - 0.302 s*

/ 1:07:26.522 PM // 1:07:26.824 PM /

User Deletion API - DELETE Request Scenarios



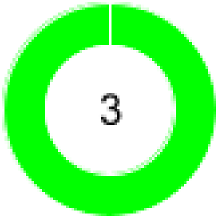
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



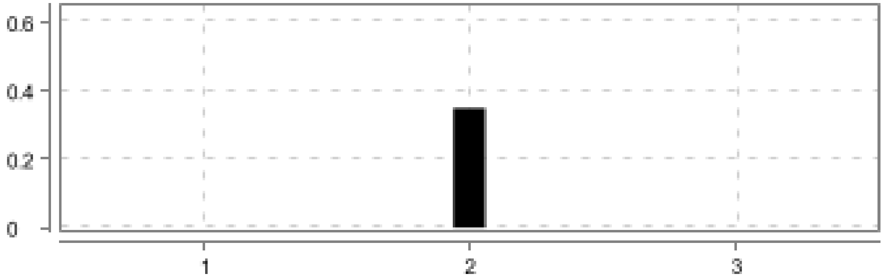
#	Step / Hook Details	Status	Duration
1	Given User enters userID with combination of number, alphabet and special characters	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.298 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Verify that the user deletion operation with invalid userID (only special characters)

PASSED *DURATION - 0.365 s*

/ 1:07:26.837 PM // 1:07:27.202 PM /

User Deletion API - DELETE Request Scenarios



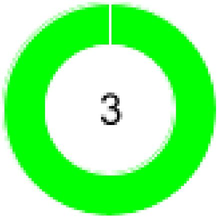
Steps

Total - 3

Pass - 3

Fail - 0

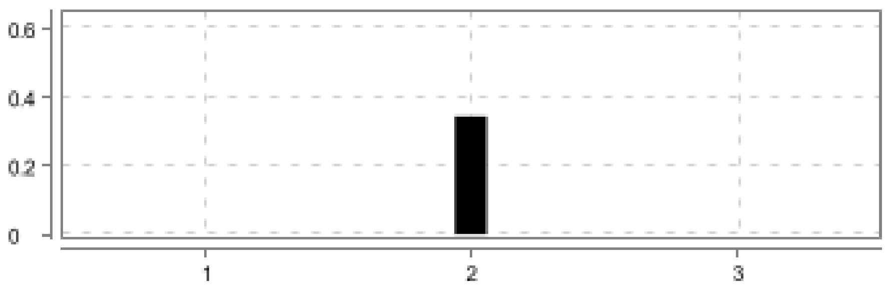
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User enters userID with only special characters	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.349 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Verify that the user deletion operation with non-existent userID

PASSED	*DURATION - 0.368 s*
/ 1:07:27.222 PM // 1:07:27.590 PM /	
User Deletion API - DELETE Request Scenarios	



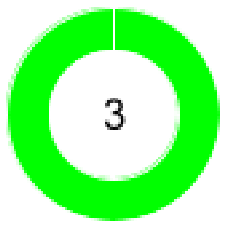
Steps

Total - 3

Pass - 3

Fail - 0

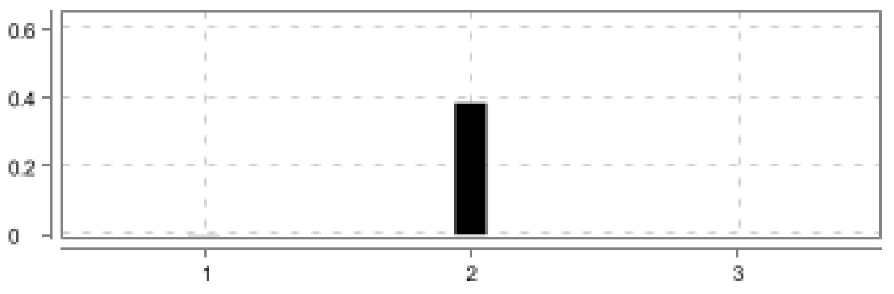
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User enters userID with 5 invalid digits	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.345 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Verify that the user deletion operation with blank userID

PASSED	*DURATION - 0.393 s*
/ 1:07:27.614 PM // 1:07:28.007 PM /	
User Deletion API - DELETE Request Scenarios	



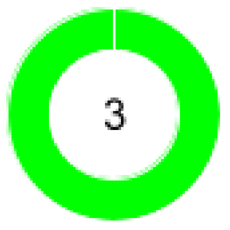
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



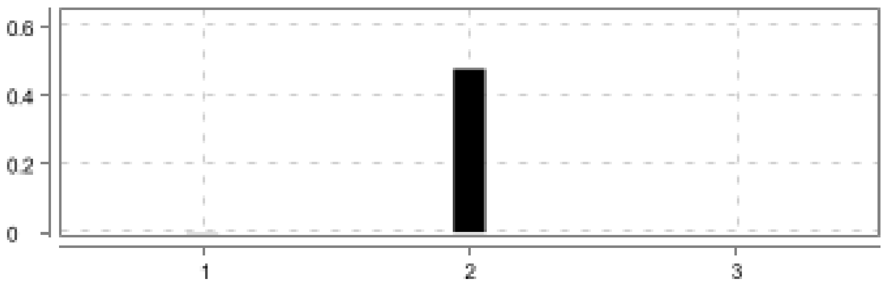
#	Step / Hook Details	Status	Duration
1	Given User enters userID with nothing (blank)	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.384 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Verify that the user deletion operation with zero value userID

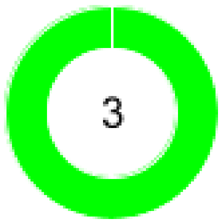
PASSED *DURATION - 0.502 s*

/ 1:07:28.025 PM // 1:07:28.527 PM /

User Deletion API - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



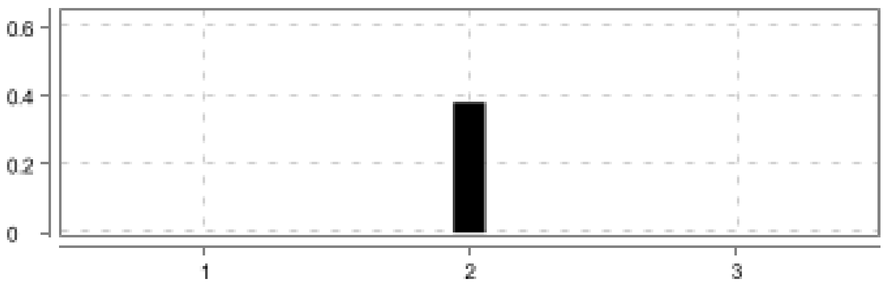
#	Step / Hook Details	Status	Duration
1	Given User enters userID with zero value	PASSED	0.001 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.477 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Verify that the user deletion operation with negative integer userID

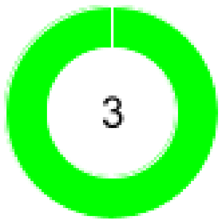
PASSED *DURATION - 0.380 s*

/ 1:07:28.543 PM // 1:07:28.923 PM /

User Deletion API - DELETE Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



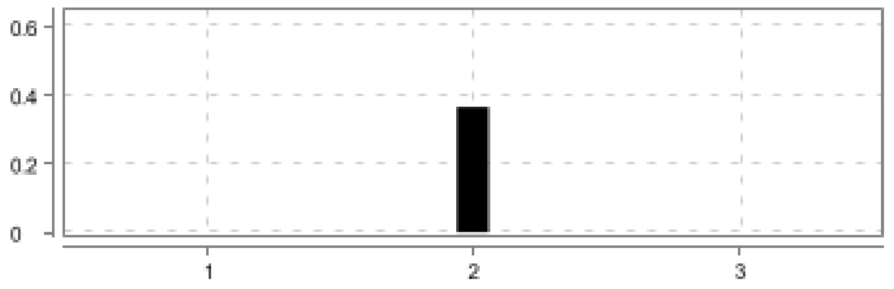
#	Step / Hook Details	Status	Duration
1	Given User enters userID with negative value	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.380 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Verify that the user deletion operation with out of range userID

PASSED *DURATION - 0.366 s*

/ 1:07:28.939 PM // 1:07:29.305 PM /

User Deletion API - DELETE Request Scenarios



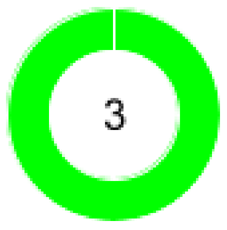
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



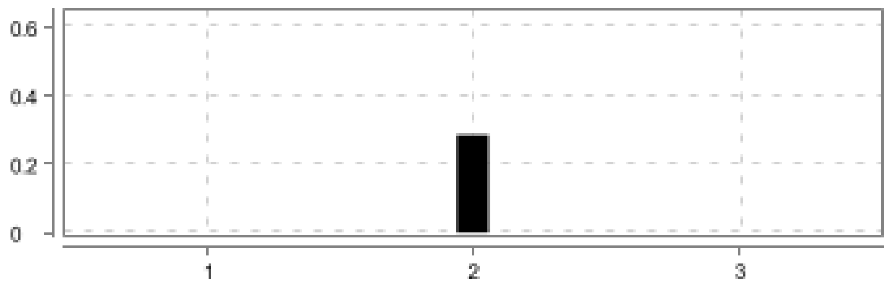
#	Step / Hook Details	Status	Duration
1	Given User enters userID with out of range value	PASSED	0.000 s
2	When User sends DELETE request to delete the respective userID	PASSED	0.366 s
3	Then User receives 400 Bad Request with the message	PASSED	0.000 s

Verify the user deletion operation with invalid BaseURL

PASSED *DURATION - 0.286 s*

/ 1:07:29.330 PM // 1:07:29.616 PM /

User Deletion API - DELETE Request Scenarios



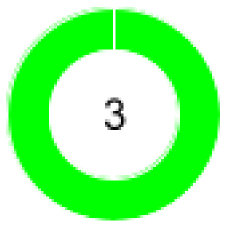
Steps

Total - 3

Pass - 3

Fail - 0

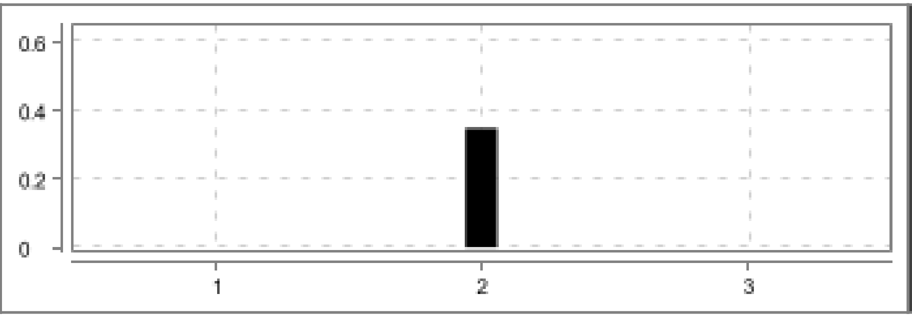
Skip - 0



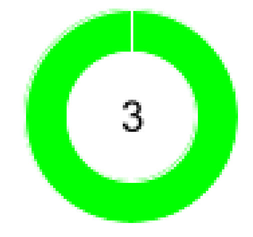
#	Step / Hook Details	Status	Duration
1	Given User enters userID with invalid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends DELETE request with the respective userID	PASSED	0.286 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Verify the user deletion operation with invalid Endpoint

PASSED	*DURATION - 0.349 s*
/ 1:07:29.632 PM // 1:07:29.981 PM /	
User Deletion API - DELETE Request Scenarios	



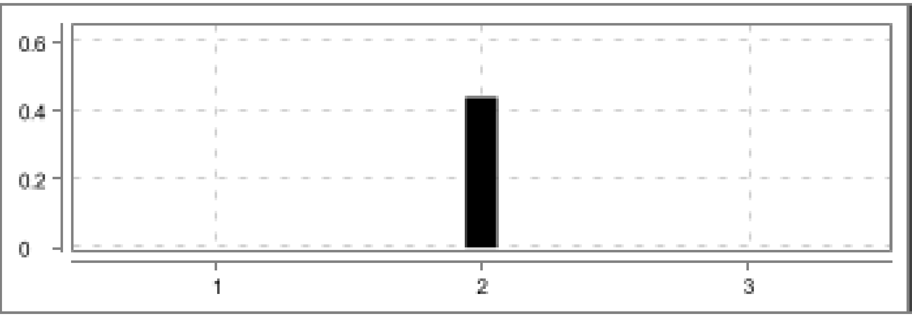
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



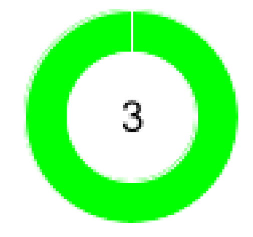
#	Step / Hook Details	Status	Duration
1	Given User enters userID with valid BaseURL and invalid Endpoint	PASSED	0.000 s
2	When User sends DELETE request with the respective userID	PASSED	0.349 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Verify the user deletion operation with empty Endpoint

PASSED	*DURATION - 0.449 s*
/ 1:07:29.995 PM // 1:07:30.444 PM /	
User Deletion API - DELETE Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



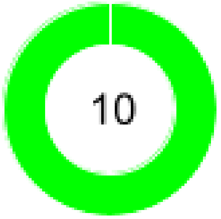
#	Step / Hook Details	Status	Duration
1	Given User enters userID with valid BaseURL and empty Endpoint	PASSED	0.000 s
2	When User sends DELETE request with the respective userID	PASSED	0.439 s
3	Then User receives 404 Not Found error with the message	PASSED	0.000 s

Get User by FirstName API - GET Request Scenarios

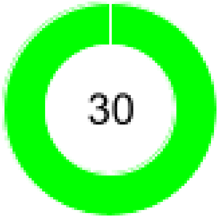
PASSED *DURATION - 4.409 s*

/ 1:07:30.459 PM // 1:07:34.868 PM /

Scenarios
Total - 10
Pass - 10
Fail - 0
Skip - 0



Steps
Total - 30
Pass - 30
Fail - 0
Skip - 0

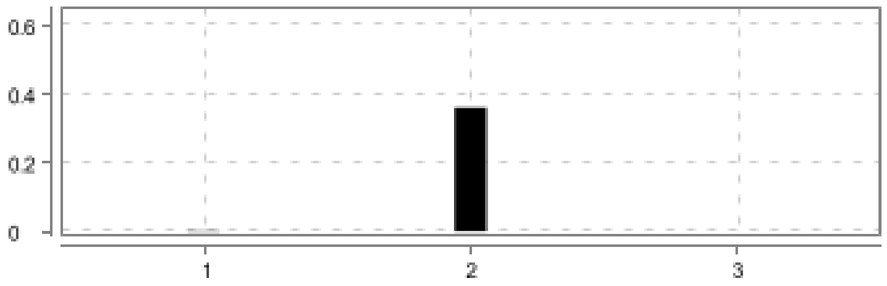


Get User by valid FirstName

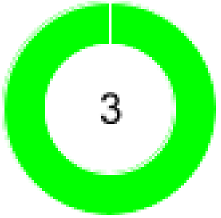
PASSED *DURATION - 0.365 s*

/ 1:07:30.459 PM // 1:07:30.824 PM /

Get User by FirstName API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



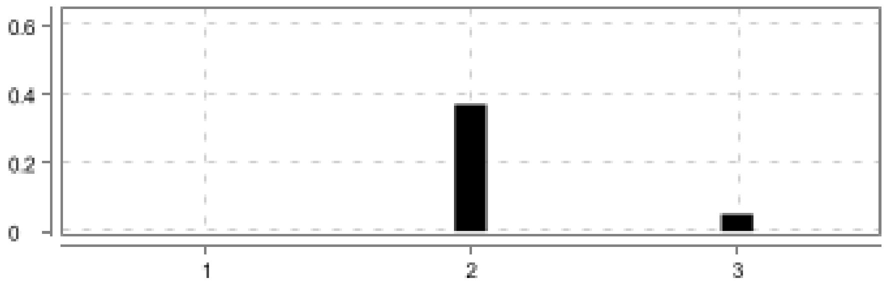
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and existing user FirstName	PASSED	0.002 s
2	When User sends GET request with existing user FirstName	PASSED	0.361 s
3	Then User receives 200 OK status with user details in response	PASSED	0.000 s

Validate FirstName in Response Matches Requested FirstName

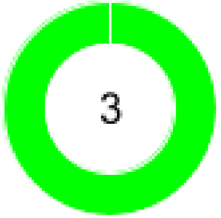
PASSED *DURATION - 0.424 s*

/ 1:07:30.840 PM // 1:07:31.264 PM /

Get User by FirstName API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and existing user FirstName	PASSED	0.000 s
2	When User sends GET request with existing user FirstName	PASSED	0.371 s

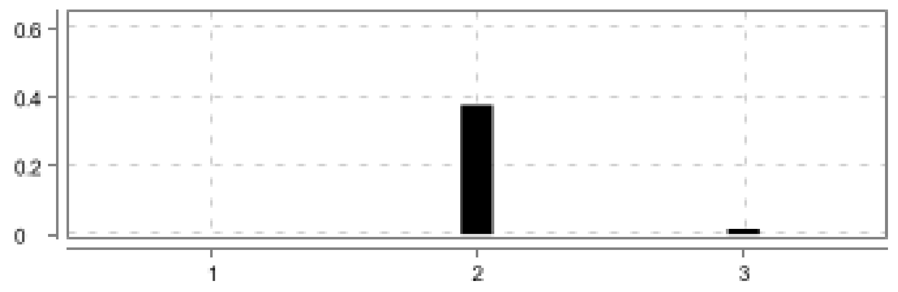
#	Step / Hook Details	Status	Duration
3	Then Response body contains same UserfirstName as requested	PASSED	0.051 s

Validate Response Body Content-Type is JSON

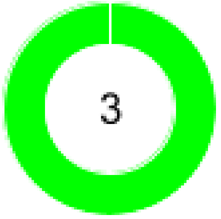
PASSED *DURATION - 0.409 s*

/ 1:07:31.271 PM // 1:07:31.680 PM /

Get User by FirstName API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



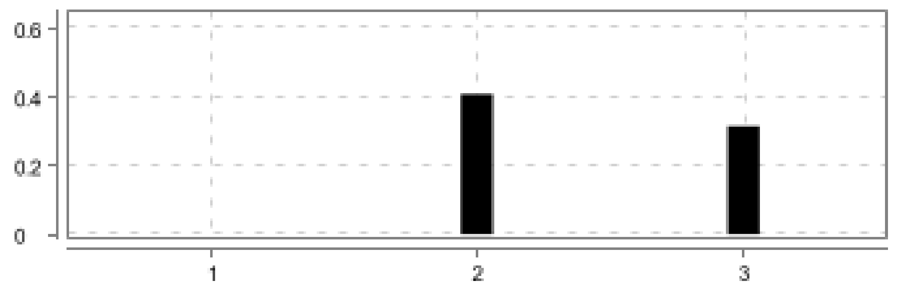
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and existing user FirstName	PASSED	0.000 s
2	When User sends GET request with existing user FirstName	PASSED	0.378 s
3	Then Response Content-Type is JSON with status code 200 OK	PASSED	0.016 s

Validate Required Fields in Response

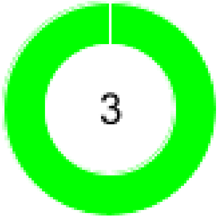
PASSED *DURATION - 0.728 s*

/ 1:07:31.724 PM // 1:07:32.452 PM /

Get User by FirstName API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



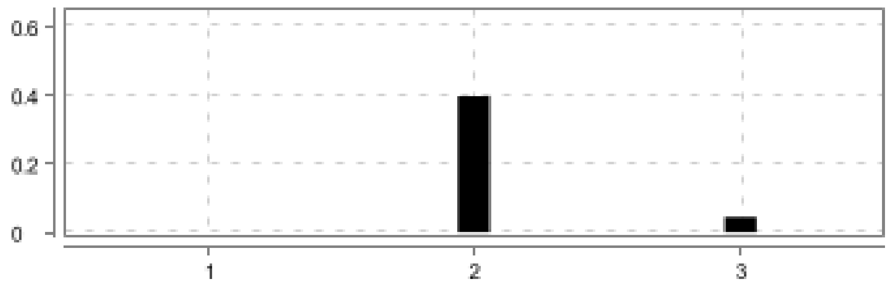
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and existing user FirstName	PASSED	0.000 s
2	When User sends GET request with existing user FirstName	PASSED	0.410 s
3	Then Response contains user details: user_id, UserfirstName, UserlastName, userContactNumber, userEmailId, creationTime, lastModTime, userAddress, addressId, plotNumber, street, state, country, zipCode	PASSED	0.315 s

Validate FirstName is String Type

PASSED	*DURATION - 0.447 s*

/ 1:07:32.465 PM // 1:07:32.912 PM /

Get User by FirstName API - GET Request Scenarios



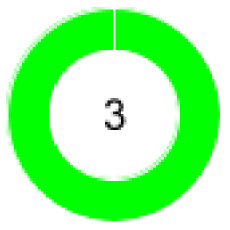
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



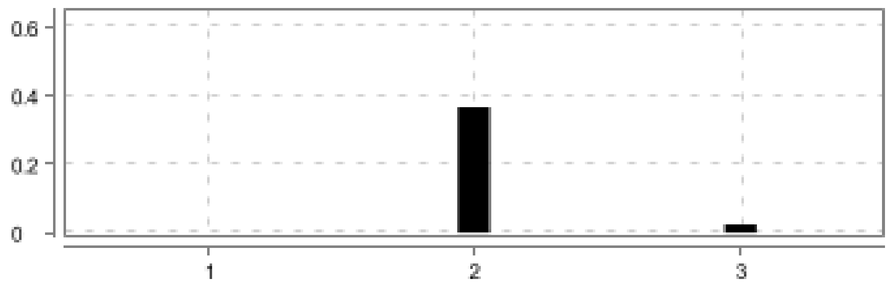
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and existing user FirstName	PASSED	0.000 s
2	When User sends GET request with existing user FirstName	PASSED	0.397 s
3	Then UserfirstName in response is of string type	PASSED	0.045 s

Validate address_id is integer type in response

PASSED	*DURATION - 0.393 s*

/ 1:07:32.928 PM // 1:07:33.321 PM /

Get User by FirstName API - GET Request Scenarios



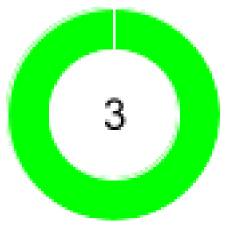
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



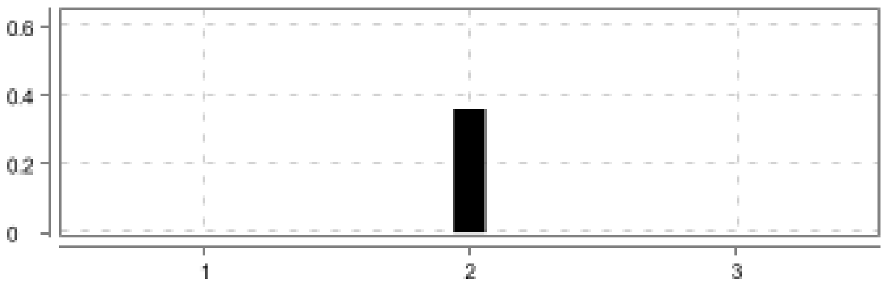
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and existing user FirstName	PASSED	0.000 s
2	When User sends GET request with existing user FirstName	PASSED	0.367 s
3	Then address_id is of integer type	PASSED	0.024 s

Get User with Non-Existing FirstName

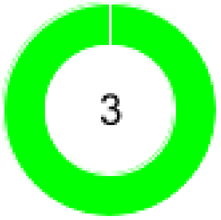
PASSED *DURATION - 0.359 s*

/ 1:07:33.327 PM // 1:07:33.686 PM /

Get User by FirstName API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



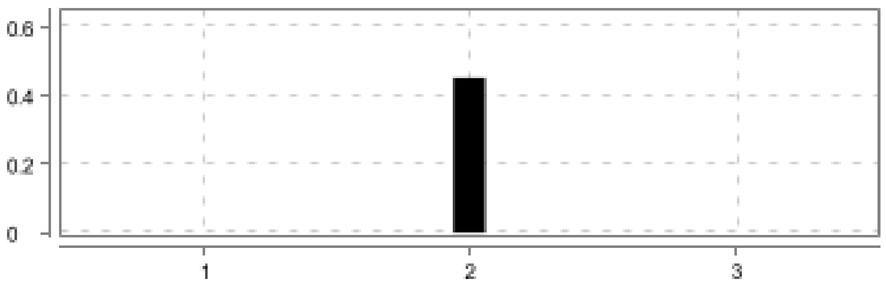
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and non-existing FirstName	PASSED	0.000 s
2	When User sends GET request with non-existent FirstName	PASSED	0.359 s
3	Then User receives 404 Not Found status with the error message	PASSED	0.000 s

Get User with Invalid FirstName Format (e.g. integer instead of string)

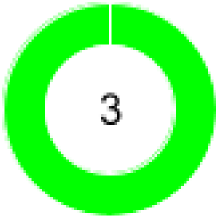
PASSED *DURATION - 0.454 s*

/ 1:07:33.701 PM // 1:07:34.155 PM /

Get User by FirstName API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



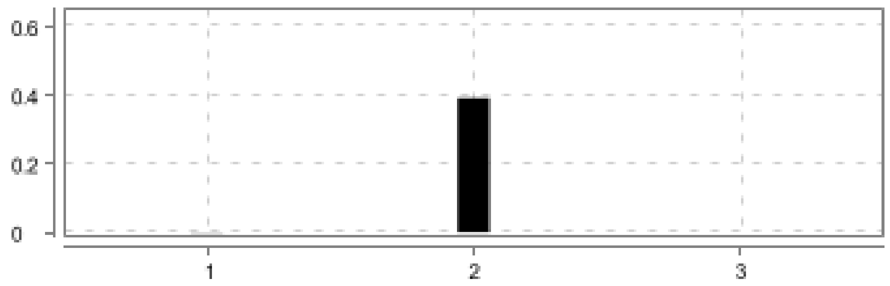
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and invalid FirstName as 123	PASSED	0.000 s
2	When User sends GET request with invalid FirstName 123	PASSED	0.452 s
3	Then User receives 404 Not Found status with the error message	PASSED	0.000 s

Get User with Invalid Endpoint

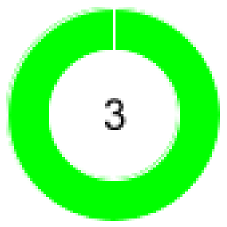
PASSED *DURATION - 0.394 s*

/ 1:07:34.163 PM // 1:07:34.557 PM /

Get User by FirstName API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



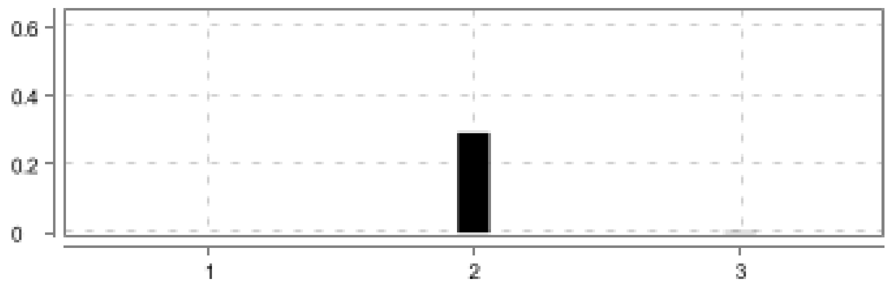
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL and invalid endpoint path	PASSED	0.001 s
2	When User sends GET request to wrong path	PASSED	0.392 s
3	Then User receives 404 Not Found status with the error message	PASSED	0.000 s

Get User without Authorization

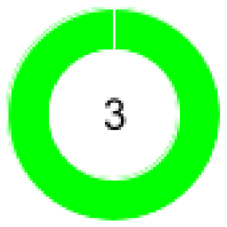
PASSED *DURATION - 0.294 s*

/ 1:07:34.574 PM // 1:07:34.868 PM /

Get User by FirstName API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



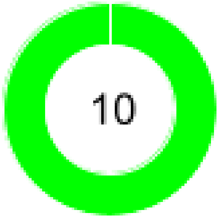
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and existing user FirstName but without Authorization header	PASSED	0.000 s
2	When User sends GET request with existing user FirstName without Auth	PASSED	0.292 s
3	Then User received 401 Unauthorized status	PASSED	0.002 s

Get User by ID API - GET Request Scenarios

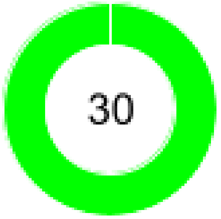
PASSED *DURATION - 4.358 s*

/ 1:07:34.882 PM // 1:07:39.240 PM /

Scenarios
Total - 10
Pass - 10
Fail - 0
Skip - 0



Steps
Total - 30
Pass - 30
Fail - 0
Skip - 0

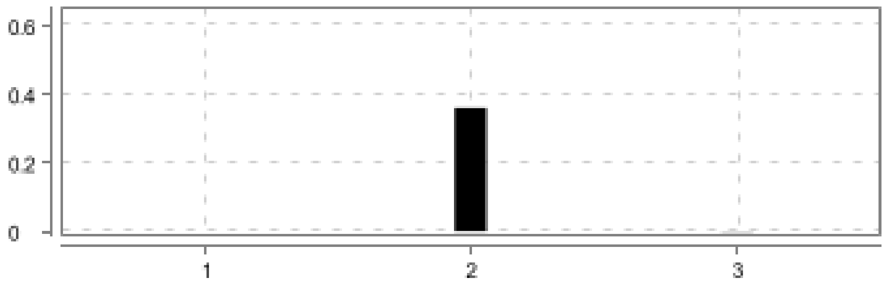


Get User by Valid ID

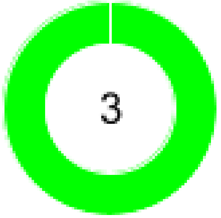
PASSED *DURATION - 0.361 s*

/ 1:07:34.882 PM // 1:07:35.243 PM /

Get User by ID API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



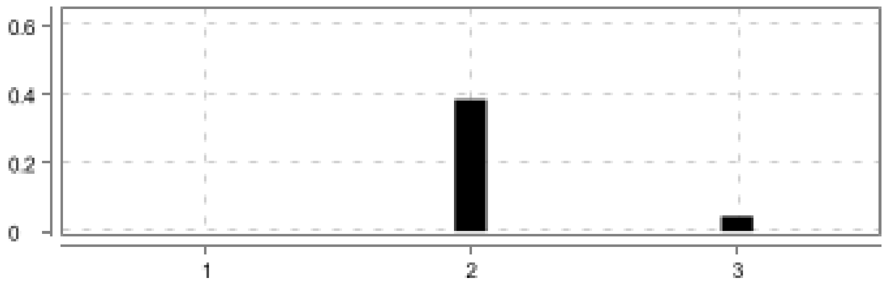
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and valid user ID	PASSED	0.000 s
2	When User sends GET request with existing user ID	PASSED	0.360 s
3	Then User receives 200 OK status with the user details in response	PASSED	0.001 s

Validate user_id in Response Matches Requested ID

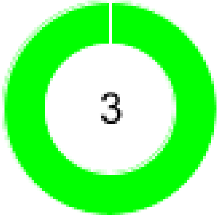
PASSED *DURATION - 0.434 s*

/ 1:07:35.257 PM // 1:07:35.691 PM /

Get User by ID API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0

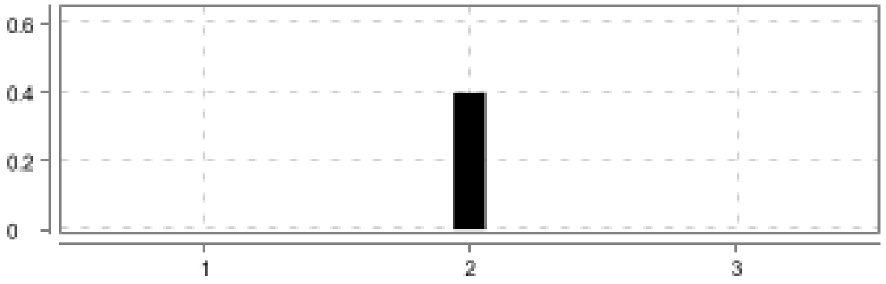


#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and valid user ID	PASSED	0.000 s
2	When User sends GET request with existing user ID	PASSED	0.385 s

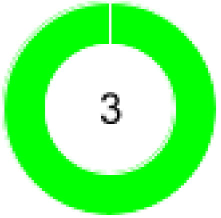
#	Step / Hook Details	Status	Duration
3	Then Response body contains same user_id as requested	PASSED	0.044 s

Validate Response Body Content-Type is JSON

PASSED	*DURATION - 0.401 s*
/ 1:07:35.702 PM // 1:07:36.103 PM /	
Get User by ID API - GET Request Scenarios	



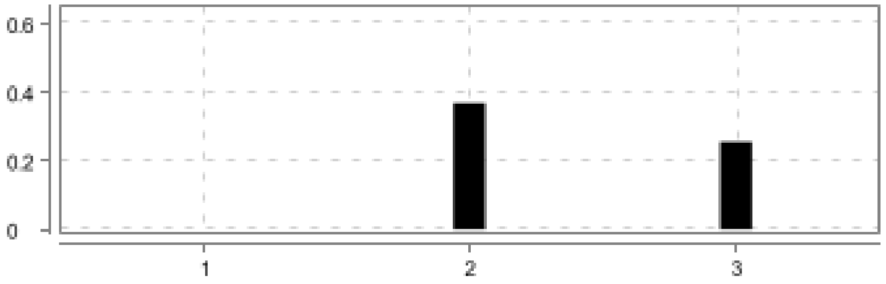
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



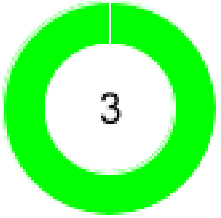
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and valid user ID	PASSED	0.000 s
2	When User sends GET request with existing user ID	PASSED	0.398 s
3	Then the Response Content-Type is JSON with status code 200 OK	PASSED	0.000 s

Validate Required Fields in Response

PASSED	*DURATION - 0.624 s*
/ 1:07:36.119 PM // 1:07:36.743 PM /	
Get User by ID API - GET Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



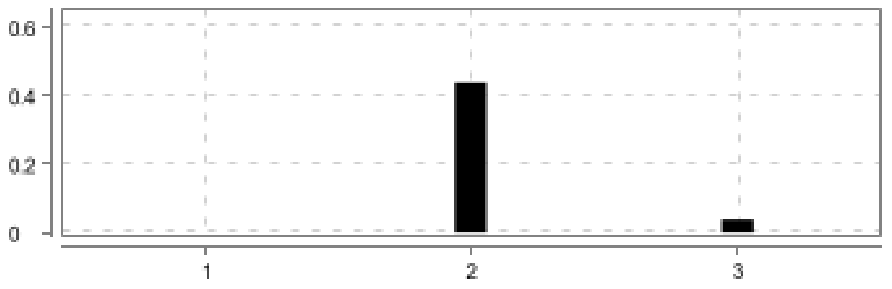
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and valid user ID	PASSED	0.000 s
2	When User sends GET request with existing user ID	PASSED	0.369 s
3	Then Response contains mandatory fields: user_id, UserfirstName, UserlastName, userContactNumber, userEmailId, creationTime, lastModTime, userAddress, addressId, plotNumber, street, state, country, zipCode	PASSED	0.255 s

Validate user_id is Integer Type

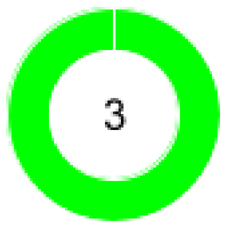
PASSED *DURATION - 0.475 s*

/ 1:07:36.759 PM // 1:07:37.234 PM /

Get User by ID API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



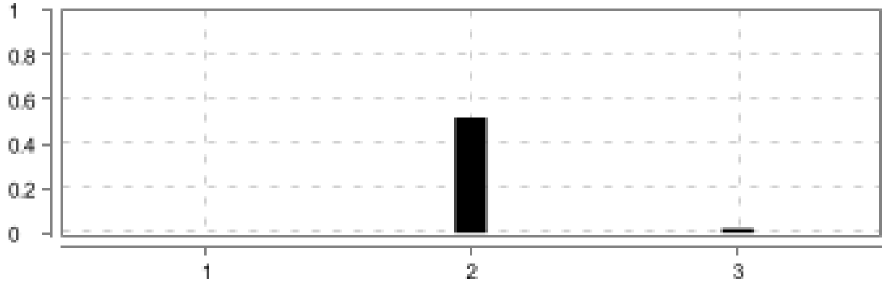
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and valid user ID	PASSED	0.000 s
2	When User sends GET request with existing user ID	PASSED	0.437 s
3	Then the user_id in response is of integer type	PASSED	0.038 s

Validate address_id is integer type in response

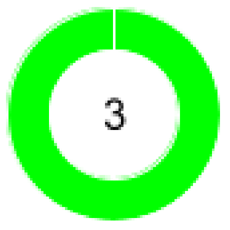
PASSED *DURATION - 0.538 s*

/ 1:07:37.249 PM // 1:07:37.787 PM /

Get User by ID API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



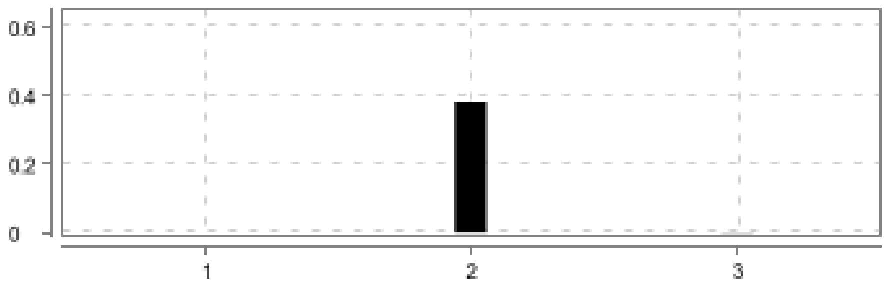
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and valid user ID	PASSED	0.000 s
2	When User sends GET request with existing user ID	PASSED	0.518 s
3	Then the address_id is of integer type	PASSED	0.017 s

Get User with Non-Existing ID

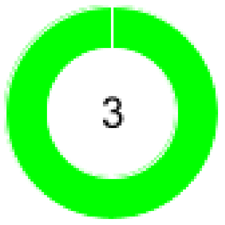
PASSED *DURATION - 0.384 s*

/ 1:07:37.797 PM // 1:07:38.181 PM /

Get User by ID API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



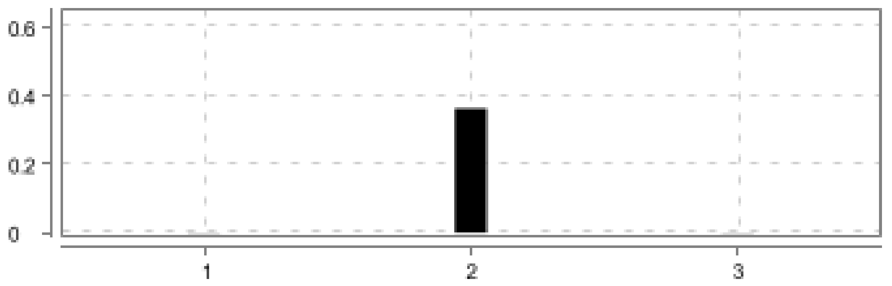
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and non-existing user ID	PASSED	0.000 s
2	When User sends GET request with non-existent ID	PASSED	0.382 s
3	Then User receives 404 Not Found status with an error message	PASSED	0.001 s

Get User with Invalid ID Format (e.g. string instead of integer)

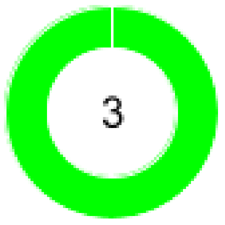
PASSED *DURATION - 0.371 s*

/ 1:07:38.195 PM // 1:07:38.566 PM /

Get User by ID API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



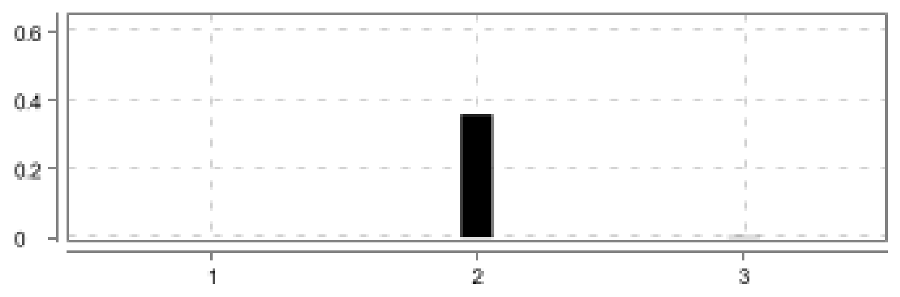
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with BaseURL, valid endpoint, and invalid user ID 'abc'	PASSED	0.001 s
2	When User sends GET request with invalid ID 'abc'	PASSED	0.363 s
3	Then User receives 400 Bad Request status with an error message	PASSED	0.001 s

Get User with Invalid Endpoint

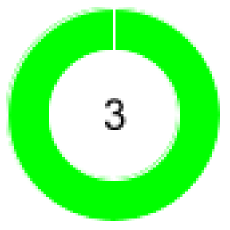
PASSED *DURATION - 0.364 s*

/ 1:07:38.577 PM // 1:07:38.941 PM /

Get User by ID API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



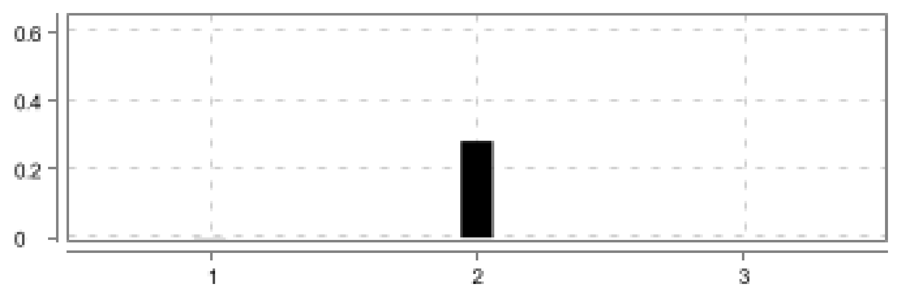
#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with the BaseURL and invalid endpoint path	PASSED	0.000 s
2	When User sends GET request to the wrong path	PASSED	0.358 s
3	Then User receives 404 Not Found status with an error message	PASSED	0.002 s

Get User without Authorization

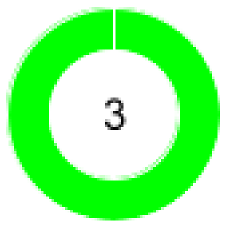
PASSED *DURATION - 0.287 s*

/ 1:07:38.953 PM // 1:07:39.240 PM /

Get User by ID API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0

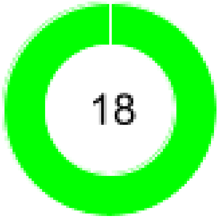


#	Step / Hook Details	Status	Duration
1	Given User prepares GET request with valid BaseURL, valid endpoint, and existing user ID but without Authorization header	PASSED	0.001 s
2	When User sends GET request with existing user ID without Authorization	PASSED	0.282 s
3	Then User receives the 401 Unauthorized status	PASSED	0.000 s

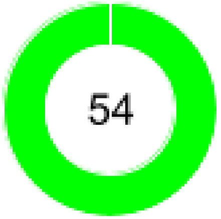
Update User Details API - PATCH Request Scenarios

PASSED	*DURATION - 7.556 s*
/ 1:07:39.264 PM // 1:07:46.820 PM /	

Scenarios
Total - 18
Pass - 18
Fail - 0
Skip - 0

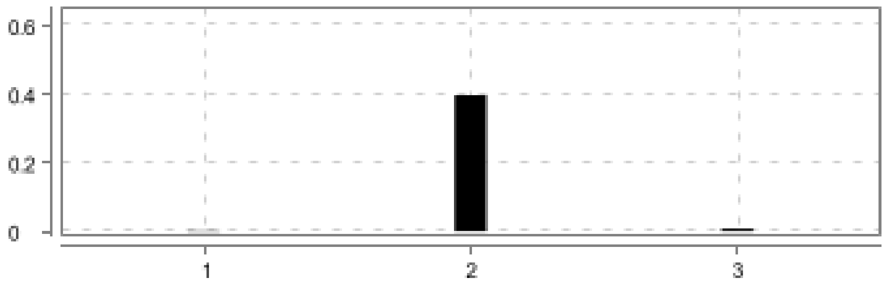


Steps
Total - 54
Pass - 54
Fail - 0
Skip - 0

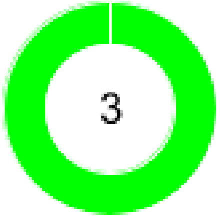


Verify that user is able to update FirstName

PASSED	*DURATION - 0.408 s*
/ 1:07:39.265 PM // 1:07:39.673 PM /	
Update User Details API - PATCH Request Scenarios	



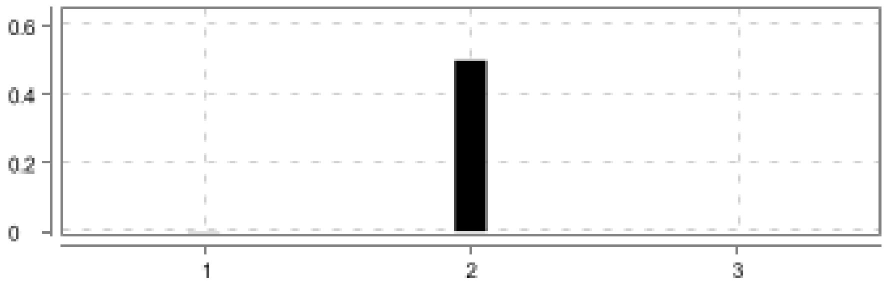
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



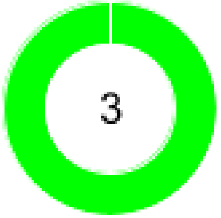
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.002 s
2	When User creates Patch request with file "patch_firstname.json"	PASSED	0.397 s
3	Then User receives 200 status code	PASSED	0.008 s

Verify that user is able to update LastName

PASSED	*DURATION - 0.501 s*
/ 1:07:39.679 PM // 1:07:40.180 PM /	
Update User Details API - PATCH Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_lastname.json"	PASSED	0.498 s

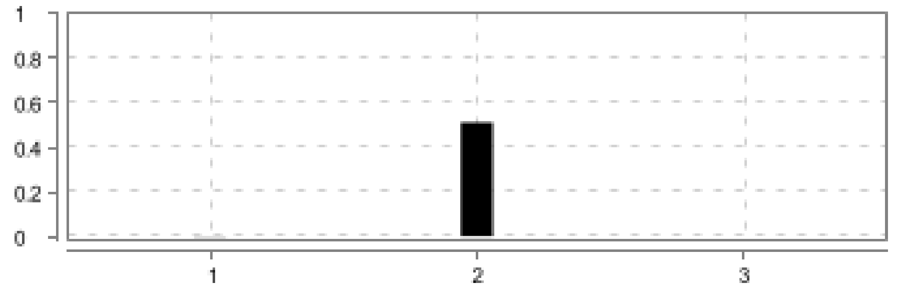
#	Step / Hook Details	Status	Duration
3	Then User receives 200 status code	PASSED	0.000 s

Verify that user is able to update contact_number

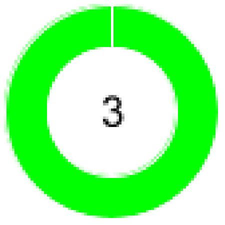
PASSED *DURATION - 0.516 s*

/ 1:07:40.193 PM // 1:07:40.709 PM /

Update User Details API - PATCH Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



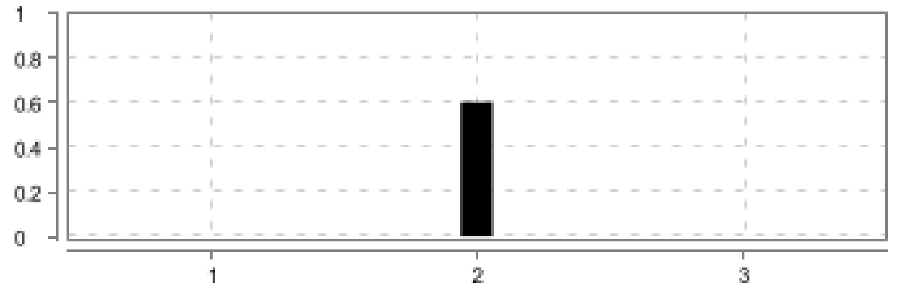
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_contactnumber.json"	PASSED	0.512 s
3	Then User receives 200 status code	PASSED	0.000 s

Verify that user is able to update email_id

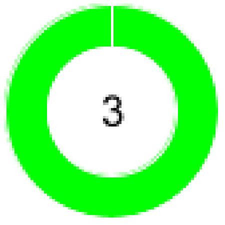
PASSED *DURATION - 0.605 s*

/ 1:07:40.734 PM // 1:07:41.339 PM /

Update User Details API - PATCH Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.000 s
2	When User creates Patch request with file "patch_emailid.json"	PASSED	0.602 s
3	Then User receives 200 status code	PASSED	0.000 s

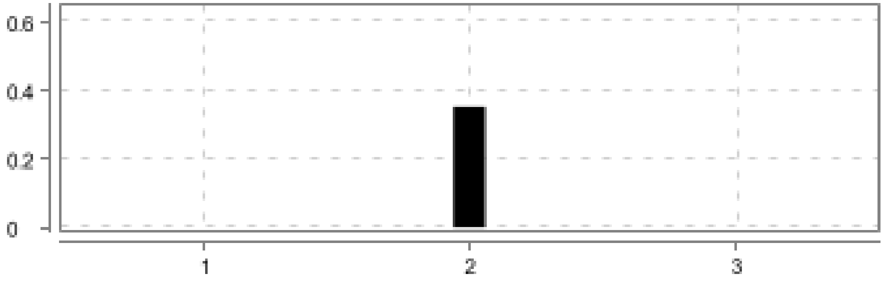
Verify that user is able to update plotNumber

PASSED

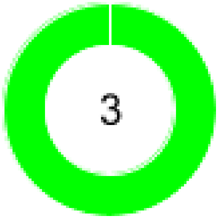
DURATION - 0.356 s

/ 1:07:41.346 PM // 1:07:41.702 PM /

Update User Details API - PATCH Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.000 s
2	When User creates Patch request with file "patch_plotnumber.json"	PASSED	0.353 s
3	Then User receives 200 status code	PASSED	0.000 s

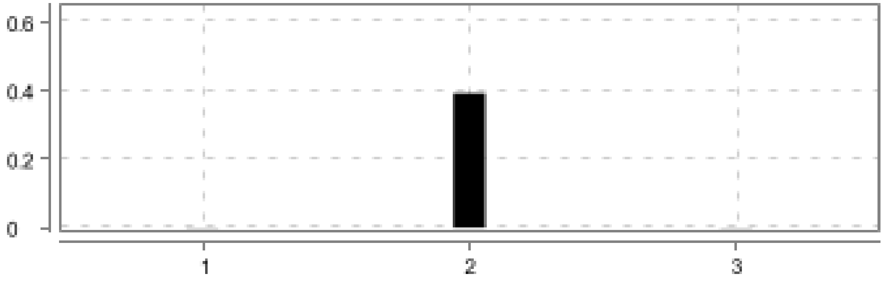
Verify that user is able to update Street

PASSED

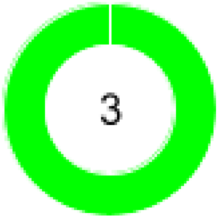
DURATION - 0.409 s

/ 1:07:41.713 PM // 1:07:42.122 PM /

Update User Details API - PATCH Request Scenarios



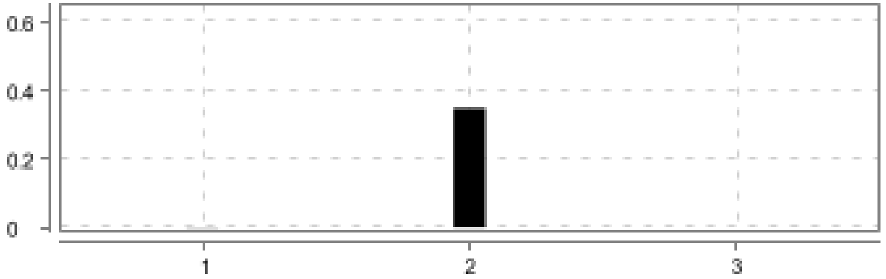
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



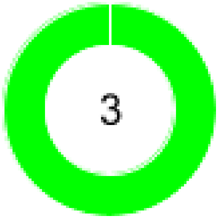
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_street.json"	PASSED	0.393 s
3	Then User receives 200 status code	PASSED	0.001 s

Verify that user is able to update state

PASSED	*DURATION - 0.352 s*
/ 1:07:42.134 PM // 1:07:42.486 PM /	
Update User Details API - PATCH Request Scenarios	



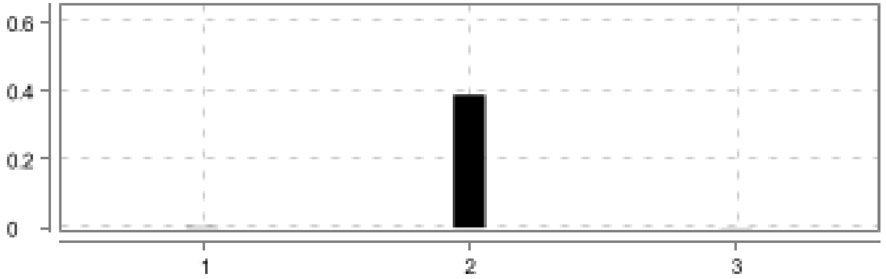
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



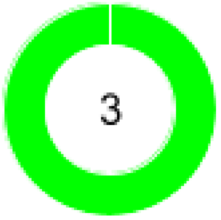
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_state.json"	PASSED	0.349 s
3	Then User receives 200 status code	PASSED	0.000 s

Verify that user is able to update Country

PASSED	*DURATION - 0.394 s*
/ 1:07:42.503 PM // 1:07:42.897 PM /	
Update User Details API - PATCH Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



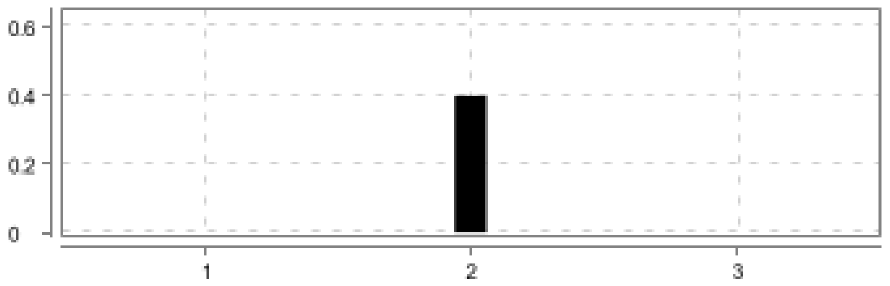
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.002 s
2	When User creates Patch request with file "patch_country.json"	PASSED	0.388 s
3	Then User receives 200 status code	PASSED	0.001 s

Verify that user is able to update zipCode

PASSED *DURATION - 0.401 s*

/ 1:07:42.914 PM // 1:07:43.315 PM /

Update User Details API - PATCH Request Scenarios



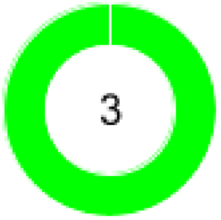
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



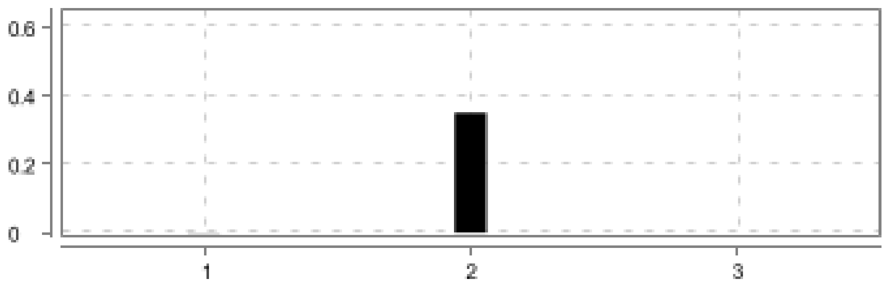
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.000 s
2	When User creates Patch request with file "patch_zipcode.json"	PASSED	0.398 s
3	Then User receives 200 status code	PASSED	0.000 s

Verify that user is not able to update contact_number with Characters

PASSED *DURATION - 0.354 s*

/ 1:07:43.327 PM // 1:07:43.681 PM /

Update User Details API - PATCH Request Scenarios



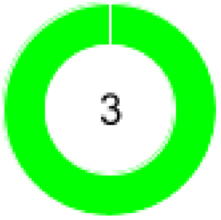
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



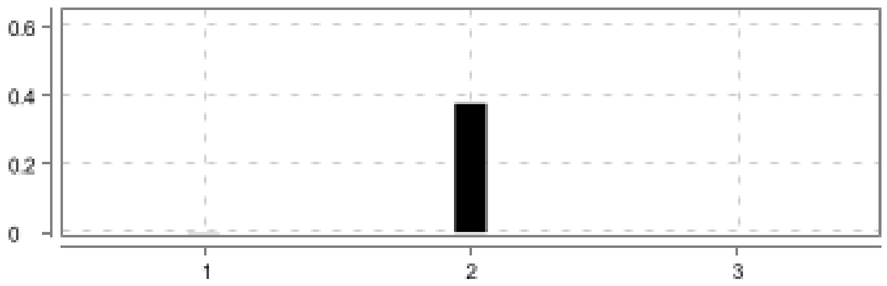
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_invalid_contactnumber.json"	PASSED	0.350 s
3	Then User receives 400 status code	PASSED	0.000 s

Verify that user is not able to update Invalid email_id

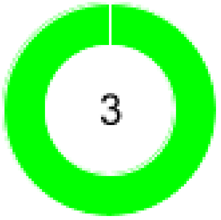
PASSED *DURATION - 0.383 s*

/ 1:07:43.692 PM // 1:07:44.075 PM /

Update User Details API - PATCH Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



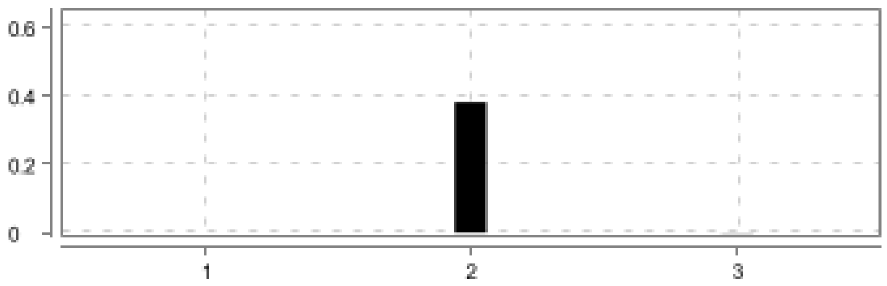
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_invalid_email.json"	PASSED	0.376 s
3	Then User receives 400 status code	PASSED	0.000 s

Verify that user is not able to update Invalid state with Numeric Data

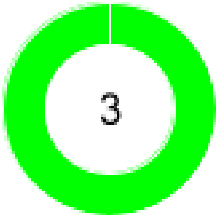
PASSED *DURATION - 0.384 s*

/ 1:07:44.083 PM // 1:07:44.467 PM /

Update User Details API - PATCH Request Scenarios



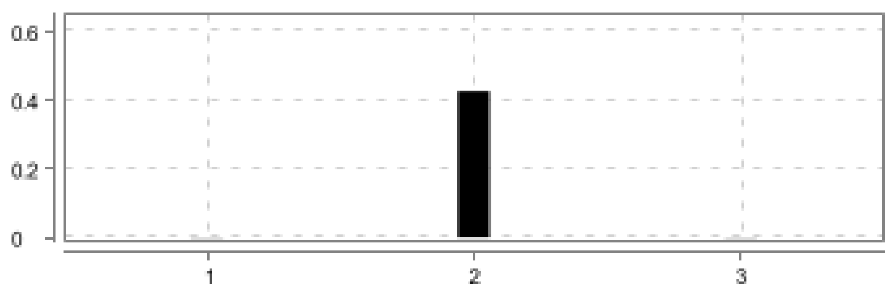
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



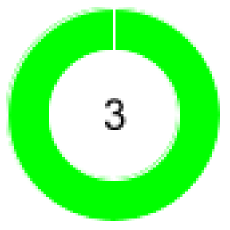
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.000 s
2	When User creates Patch request with file "patch_invalid_state.json"	PASSED	0.381 s
3	Then User receives 400 status code	PASSED	0.001 s

Verify that user is not able to update Invalid Country with Numeric Data

PASSED	*DURATION - 0.433 s*
/ 1:07:44.479 PM // 1:07:44.912 PM /	
Update User Details API - PATCH Request Scenarios	



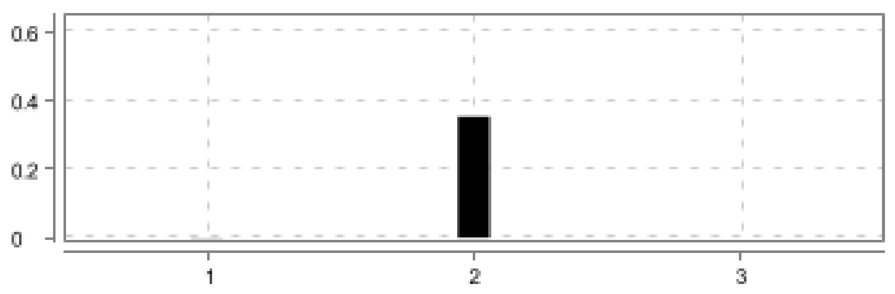
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



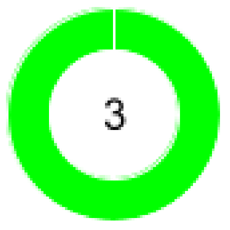
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_invalid_country.json"	PASSED	0.427 s
3	Then User receives 400 status code	PASSED	0.001 s

Verify that user is not able to update Invalid zipCode with Characters

PASSED	*DURATION - 0.357 s*
/ 1:07:44.920 PM // 1:07:45.277 PM /	
Update User Details API - PATCH Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_invalid_zipcode.json"	PASSED	0.354 s
3	Then User receives 400 status code	PASSED	0.000 s

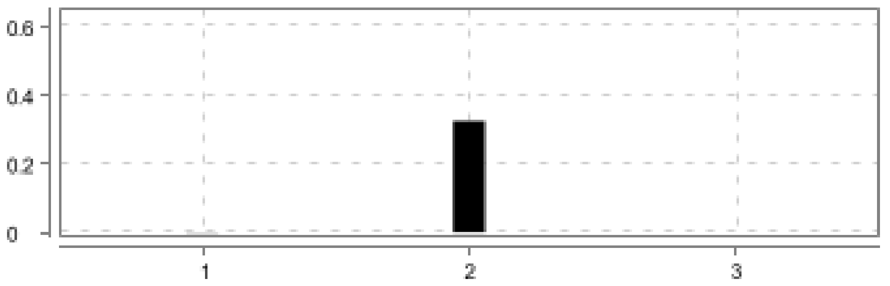
Verify that user is not able to update FirstName with invalid BaseURL

PASSED

DURATION - 0.327 s

/ 1:07:45.285 PM // 1:07:45.612 PM /

Update User Details API - PATCH Request Scenarios



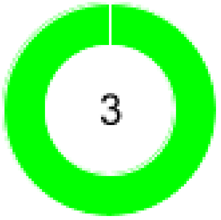
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, invalid BaseURL, valid Endpoint	PASSED	0.001 s
2	When User creates Patch request with file "patch_firstname.json"	PASSED	0.325 s
3	Then User receives 404 status code	PASSED	0.000 s

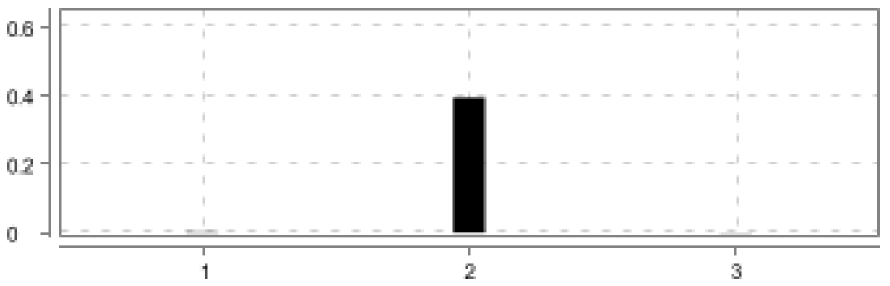
Verify that user is not able to update FirstName with valid BaseURL, Invalid EndPoint

PASSED

DURATION - 0.401 s

/ 1:07:45.629 PM // 1:07:46.030 PM /

Update User Details API - PATCH Request Scenarios



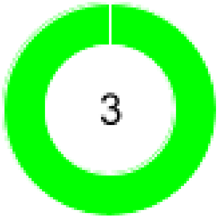
Steps

Total - 3

Pass - 3

Fail - 0

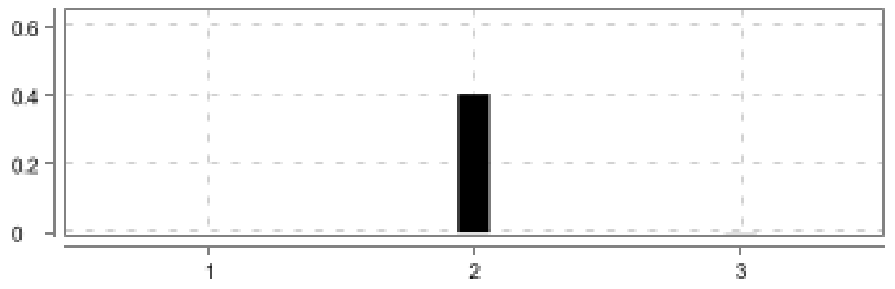
Skip - 0



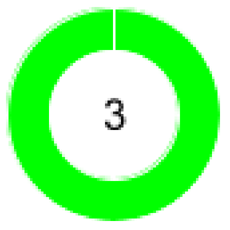
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, valid BaseURL, invalid Endpoint	PASSED	0.002 s
2	When User creates Patch request with file "patch_firstname.json"	PASSED	0.395 s
3	Then User receives 400 status code	PASSED	0.001 s

Verify that user is not able to update FirstName with valid BaseURL, empty EndPoint

PASSED	*DURATION - 0.410 s*
/ 1:07:46.039 PM // 1:07:46.449 PM /	
Update User Details API - PATCH Request Scenarios	



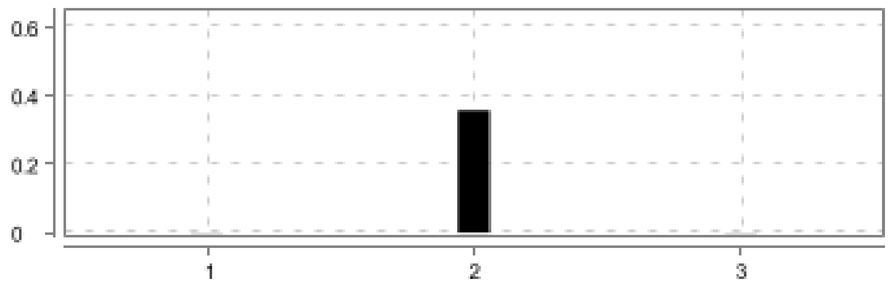
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



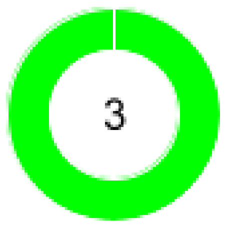
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, valid BaseURL, empty Endpoint	PASSED	0.000 s
2	When User creates Patch request with file "patch_firstname.json"	PASSED	0.404 s
3	Then User receives 404 status code	PASSED	0.001 s

Verify that user is not able to update FirstName with BaseURL, valid Endpoint and No Auth

PASSED	*DURATION - 0.362 s*
/ 1:07:46.458 PM // 1:07:46.820 PM /	
Update User Details API - PATCH Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



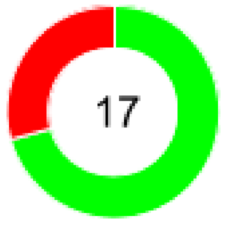
#	Step / Hook Details	Status	Duration
1	Given Patch request with new data, valid BaseURL, valid Endpoint and No Auth	PASSED	0.001 s
2	When User creates Patch request with file "patch_firstname.json"	PASSED	0.357 s
3	Then User receives 401 status code	PASSED	0.001 s

Add New User API - Negative Scenarios

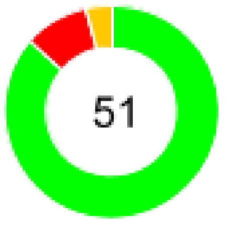
FAILED *DURATION - 6.119 s*

/ 1:07:46.835 PM // 1:07:52.954 PM /

Scenarios
Total - 17
Pass - 12
Fail - 5
Skip - 0



Steps
Total - 51
Pass - 44
Fail - 5
Skip - 2

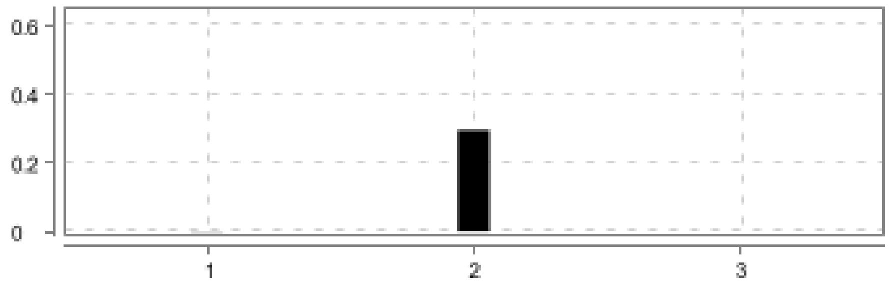


Add a new user without Authorization

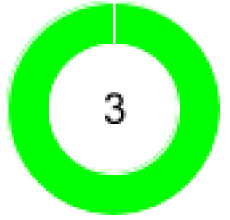
PASSED *DURATION - 0.298 s*

/ 1:07:46.835 PM // 1:07:47.133 PM /

Add New User API - Negative Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



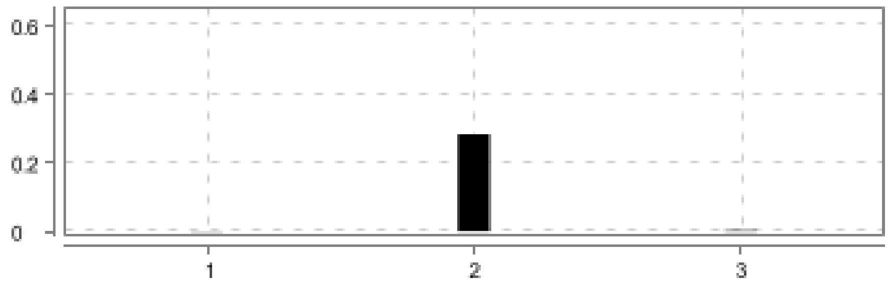
#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.001 s
2	When Users sends the POST request with all mandatory fields but without Authorization	PASSED	0.295 s
3	Then Users receives 401 Unauthorized status	PASSED	0.000 s

Add a new user with invalid BaseURL

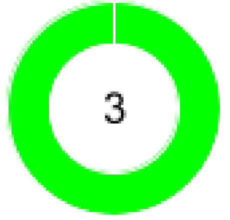
PASSED *DURATION - 0.289 s*

/ 1:07:47.139 PM // 1:07:47.428 PM /

Add New User API - Negative Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with invalid BaseURL and valid endpoint	PASSED	0.001 s
2	When User sends the POST request with all required user details	PASSED	0.283 s

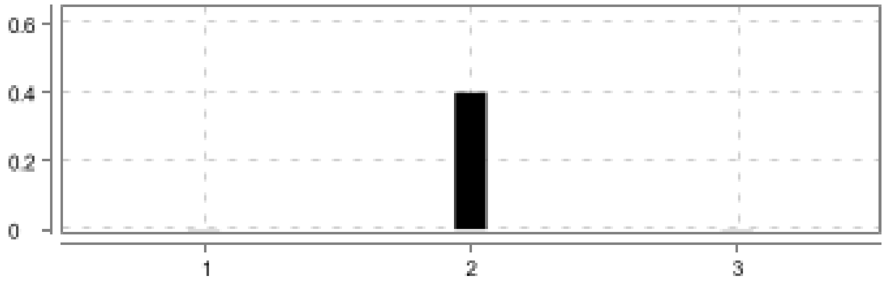
#	Step / Hook Details	Status	Duration
3	Then User receives 404 Not Found status with appropriate message	PASSED	0.003 s

Add a new user with invalid endpoint

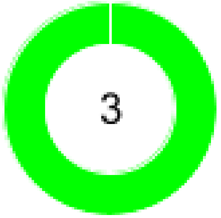
PASSED	*DURATION - 0.402 s*

/ 1:07:47.437 PM // 1:07:47.839 PM /

Add New User API - Negative Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



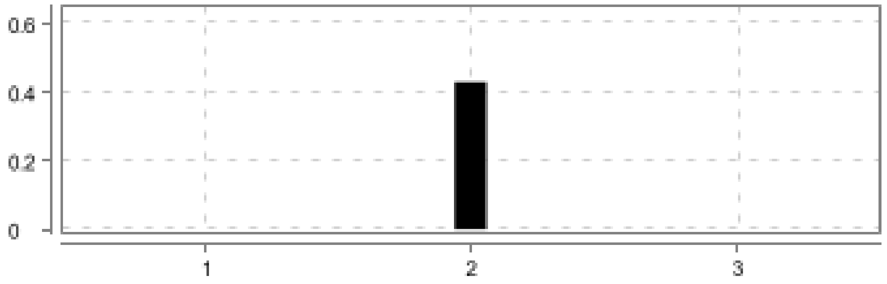
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and invalid endpoint	PASSED	0.001 s
2	When User sends the POST request with all required user details	PASSED	0.399 s
3	Then User receives 404 Not Found status with appropriate message	PASSED	0.001 s

Add a new user with empty endpoint

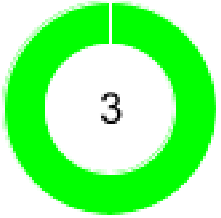
PASSED	*DURATION - 0.432 s*

/ 1:07:47.845 PM // 1:07:48.277 PM /

Add New User API - Negative Scenarios



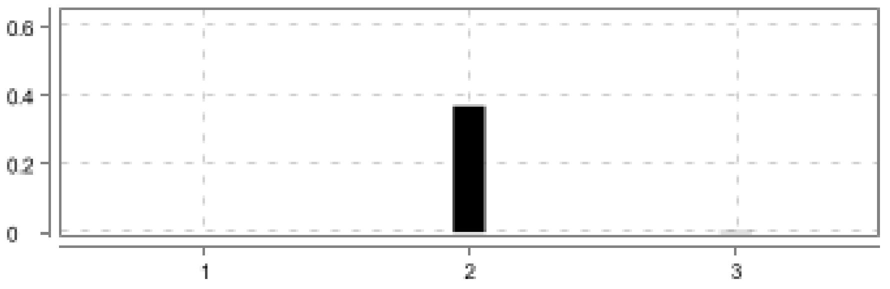
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



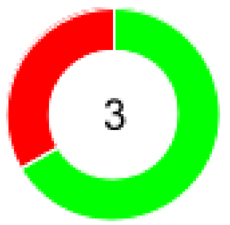
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and empty endpoint	PASSED	0.000 s
2	When User sends the POST request with all required user details	PASSED	0.430 s
3	Then User receives 404 Not Found status with appropriate message	PASSED	0.000 s

Add a new user with empty request body

FAILED	*DURATION - 0.376 s*
/ 1:07:48.281 PM // 1:07:48.657 PM /	
Add New User API - Negative Scenarios	



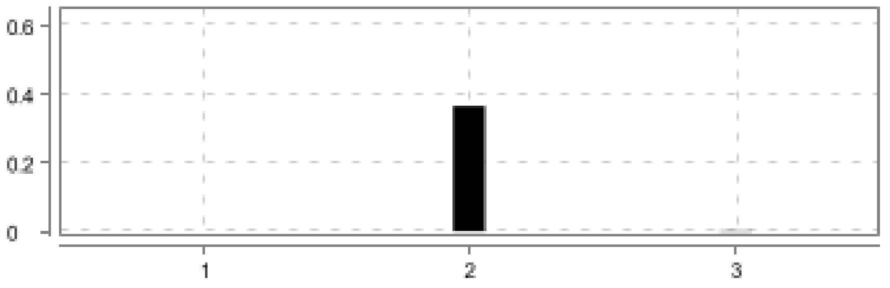
Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



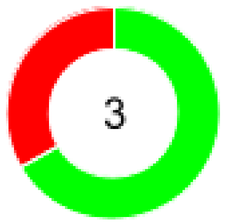
#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with an empty request body	PASSED	0.369 s
3	Then User receives 400 Bad Request status with appropriate message	FAILED	0.002 s
	java.lang.AssertionError: expected [400] but found [500] at org.testng.Assert.fail(Assert.java:111) at org.testng.Assert.failNotEquals(Assert.java:1590) at org.testng.Assert.assertEqualsImpl(Assert.java:150) at org.testng.Assert.assertEquals(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:1431) at org.testng.Assert.assertEquals(Assert.java:1395) at org.testng.Assert.assertEquals(Assert.java:1441) at StepDefinition.PostNegative_CreateUsers.user_receives_400_bad_request_status_with_appropriate_message(PostNegative_CreateUsers.java:186) at ?.User receives 400 Bad Request status with appropriate message(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/Post_Negative.feature:26) * Not displayable characters are replaced by '?'.		

Add a new user with only optional fields

FAILED	*DURATION - 0.370 s*
/ 1:07:48.664 PM // 1:07:49.034 PM /	
Add New User API - Negative Scenarios	



Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When Users sends the POST request without only mandatory fields	PASSED	0.366 s
3	Then User receives 400 Bad Request status with appropriate message	FAILED	0.002 s

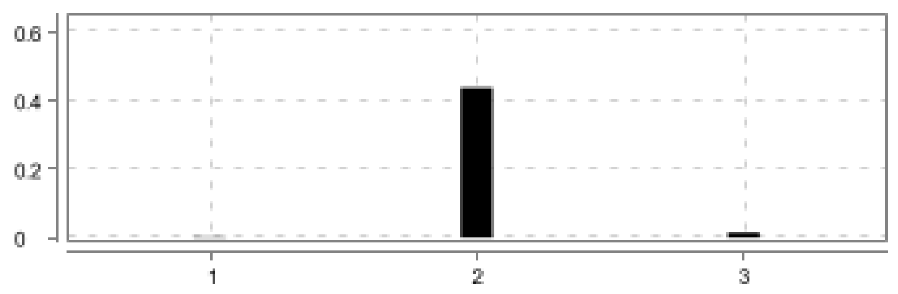
```
java.lang.AssertionError: expected [400] but found [500]
    at org.testng.Assert.fail(Assert.java:111)
    at org.testng.Assert.failNotEquals(Assert.java:1590)
    at org.testng.Assert.assertEqualsImpl(Assert.java:150)
    at org.testng.Assert.assertEquals(Assert.java:132)
    at org.testng.Assert.assertEquals(Assert.java:1431)
    at org.testng.Assert.assertEquals(Assert.java:1395)
    at org.testng.Assert.assertEquals(Assert.java:1441)
    at StepDefinition.PostNegative_CreateUsers.
user_receives_400_bad_request_status_with_appropriate_message(PostNegative_CreateUsers.java:186)
    at ?.User receives 400 Bad Request status with appropriate message(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/
test/resources/Features/Post_Negative.feature:31)
* Not displayable characters are replaced by '?'.
```

Add a new user with existing email and contact

PASSED *DURATION - 0.455 s*

/ 1:07:49.041 PM // 1:07:49.496 PM /

Add New User API - Negative Scenarios



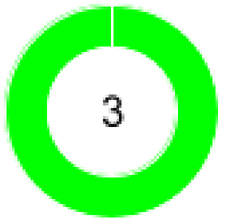
Steps

Total - 3

Pass - 3

Fail - 0

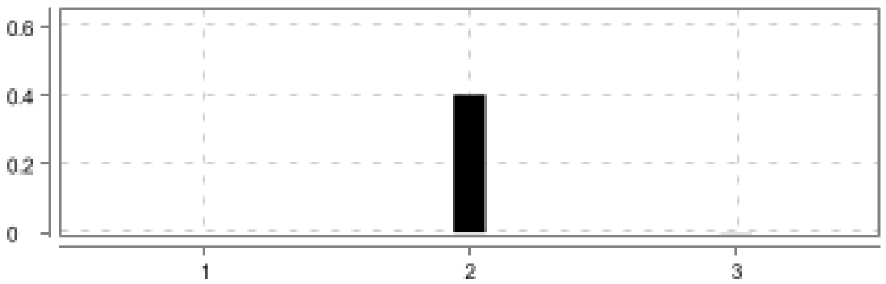
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.002 s
2	When Users sends the POST request with existing email and contact number	PASSED	0.438 s
3	Then User receives 409 Conflict status for existing entry	PASSED	0.014 s

Add a new user with invalid first name

PASSED	*DURATION - 0.405 s*
/ 1:07:49.503 PM // 1:07:49.908 PM /	
Add New User API - Negative Scenarios	



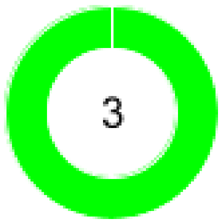
Steps

Total - 3

Pass - 3

Fail - 0

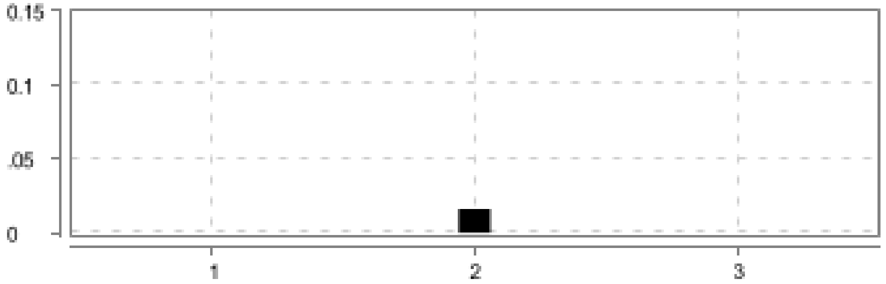
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with invalid first name	PASSED	0.402 s
3	Then User receives 400 Bad Request status with appropriate message	PASSED	0.001 s

Add a new user with invalid last name

FAILED	*DURATION - 0.017 s*
/ 1:07:49.917 PM // 1:07:49.934 PM /	
Add New User API - Negative Scenarios	



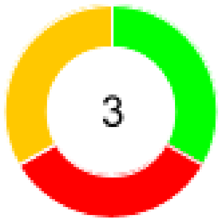
Steps

Total - 3

Pass - 1

Fail - 1

Skip - 1



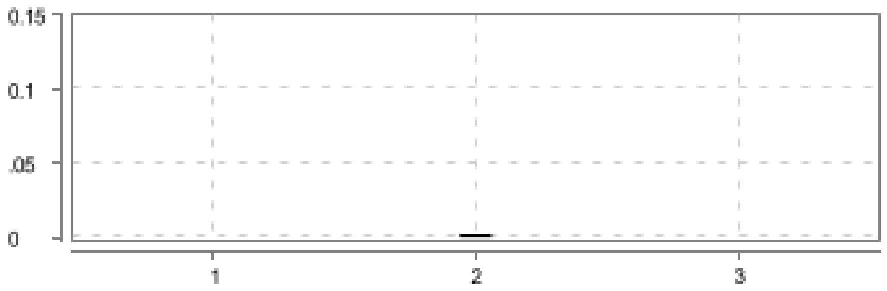
#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with invalid last name	FAILED	0.016 s
	java.lang.RuntimeException: JSON object is null. Check the file path or contents. at StepDefinition.PostNegative_CreateUsers.sendRequest(PostNegative_CreateUsers.java:193) at StepDefinition.PostNegative_CreateUsers.sendInvalidLastName(PostNegative_CreateUsers.java:92) at ?.User sends the POST request with invalid last name(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/Post_Negative.feature:45) * Not displayable characters are replaced by '?'.		
3	Then User receives 400 Bad Request status with appropriate message	SKIPPED	0.000 s

Add a new user with special characters in contact number

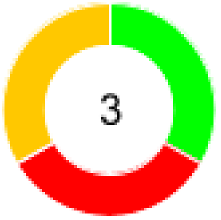
FAILED *DURATION - 0.004 s*

/ 1:07:49.945 PM // 1:07:49.949 PM /

Add New User API - Negative Scenarios



Steps
Total - 3
Pass - 1
Fail - 1
Skip - 1



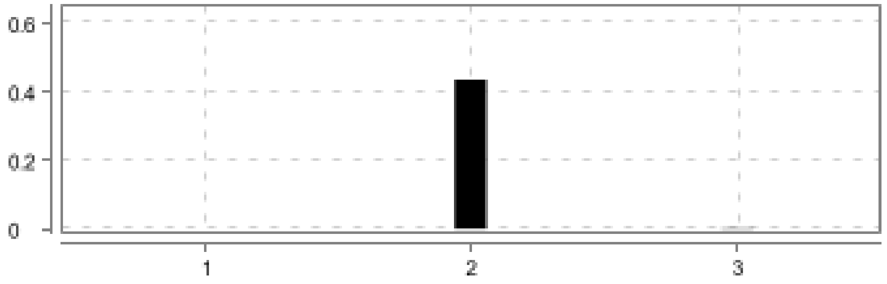
#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with special characters in contact number	FAILED	0.002 s
	java.lang.RuntimeException: JSON object is null. Check the file path or contents. at StepDefinition.PostNegative_CreateUsers.sendRequest(PostNegative_CreateUsers.java:193) at StepDefinition.PostNegative_CreateUsers.sendSpecialCharInContact(PostNegative_CreateUsers.java:100) at ?.User sends the POST request with special characters in contact number(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/Post_Negative.feature:50) * Not displayable characters are replaced by '?'.		
3	Then User receives 400 Bad Request status with appropriate message	SKIPPED	0.000 s

Add a new user with contact number less than 10 digits

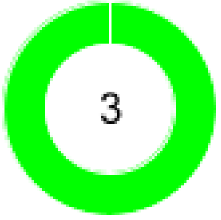
PASSED *DURATION - 0.439 s*

/ 1:07:49.952 PM // 1:07:50.391 PM /

Add New User API - Negative Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



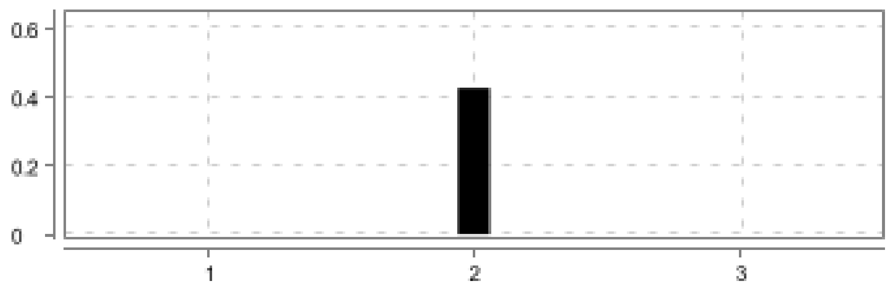
#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with less than 10 digits in contact number	PASSED	0.436 s
3	Then User receives 400 Bad Request status with appropriate message	PASSED	0.002 s

Add a new user with contact number more than 10 digits

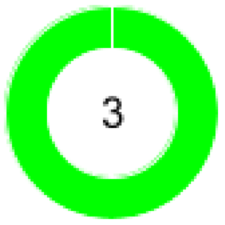
PASSED *DURATION - 0.429 s*

/ 1:07:50.396 PM // 1:07:50.825 PM /

Add New User API - Negative Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



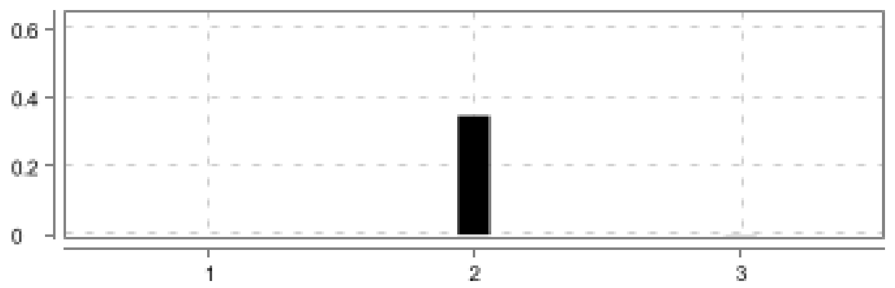
#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with more than 10 digits in contact number	PASSED	0.427 s
3	Then User receives 400 Bad Request status with appropriate message	PASSED	0.000 s

Add a new user with integer plot number

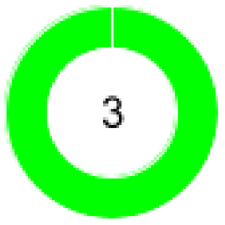
PASSED *DURATION - 0.349 s*

/ 1:07:50.829 PM // 1:07:51.178 PM /

Add New User API - Negative Scenarios



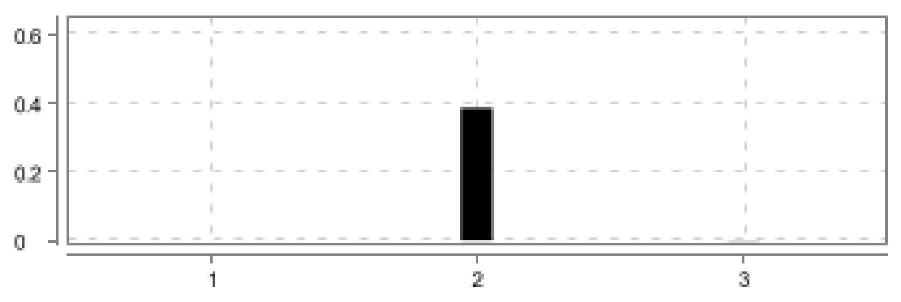
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with numeric plot number	PASSED	0.347 s
3	Then User receives 400 Bad Request status with appropriate message	PASSED	0.001 s

Add a new user with special characters in street name

PASSED	*DURATION - 0.392 s*
/ 1:07:51.187 PM // 1:07:51.579 PM /	
Add New User API - Negative Scenarios	



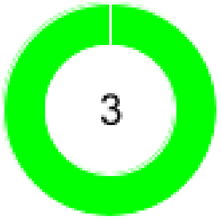
Steps

Total - 3

Pass - 3

Fail - 0

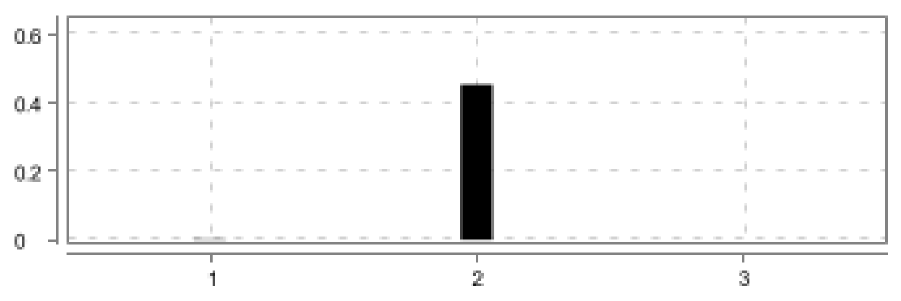
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with special characters in street name	PASSED	0.387 s
3	Then User receives 400 Bad Request status with appropriate message	PASSED	0.001 s

Add a new user with numeric state value

PASSED	*DURATION - 0.457 s*
/ 1:07:51.586 PM // 1:07:52.043 PM /	
Add New User API - Negative Scenarios	



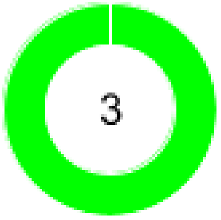
Steps

Total - 3

Pass - 3

Fail - 0

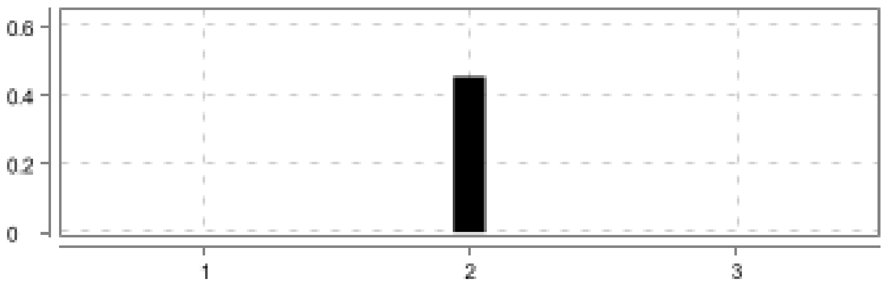
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.002 s
2	When User sends the POST request with numeric value in state	PASSED	0.454 s
3	Then User receives 400 Bad Request status with appropriate message	PASSED	0.000 s

Add a new user with numeric country value

PASSED	*DURATION - 0.456 s*
/ 1:07:52.047 PM // 1:07:52.503 PM /	
Add New User API - Negative Scenarios	



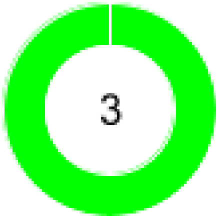
Steps

Total - 3

Pass - 3

Fail - 0

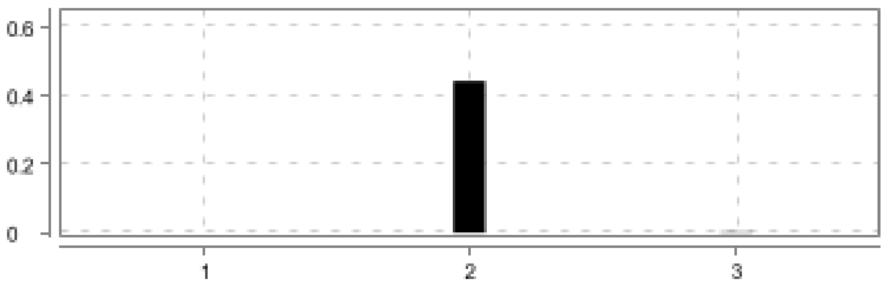
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with numeric value in country	PASSED	0.455 s
3	Then User receives 400 Bad Request status with appropriate message	PASSED	0.000 s

Add a new user with special characters in zip code

FAILED	*DURATION - 0.445 s*
/ 1:07:52.509 PM // 1:07:52.954 PM /	
Add New User API - Negative Scenarios	



Steps

Total - 3

Pass - 2

Fail - 1

Skip - 0

#	Step / Hook Details	Status	Duration
1	Given Users prepare the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with special characters in zip code	PASSED	0.442 s
3	Then User receives 400 Bad Request status with appropriate message	FAILED	0.002 s

```
java.lang.AssertionError: expected [true] but found [false]
    at org.testng.Assert.fail(Assert.java:111)
    at org.testng.Assert.failNotEquals(Assert.java:1590)
    at org.testng.Assert.assertTrue(Assert.java:57)
    at org.testng.Assert.assertTrue(Assert.java:67)
    at StepDefinition.PostNegative_CreateUsers.
user_receives_400_bad_request_status_with_appropriate_message(PostNegative_CreateUsers.java:187)
    at ?.User receives 400 Bad Request status with appropriate message(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/
test/resources/Features/Post_Negative.feature:86)
```

#	Step / Hook Details	Status	Duration
	* Not displayable characters are replaced by '?'.		

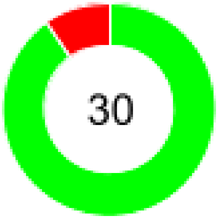
Update User Details API - PUT Request Scenarios

FAILED

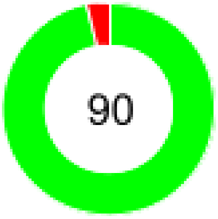
DURATION - 13.033 s

/ 1:07:52.977 PM // 1:08:06.010 PM /

Scenarios
Total - 30
Pass - 27
Fail - 3
Skip - 0



Steps
Total - 90
Pass - 87
Fail - 3
Skip - 0



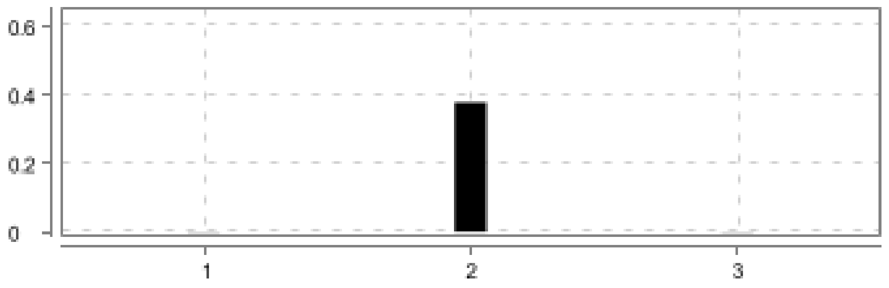
Update user details with valid BaseURL and valid Endpoints

PASSED

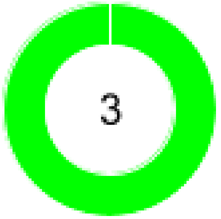
DURATION - 0.383 s

/ 1:07:52.977 PM // 1:07:53.360 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "update_valid_all_fields.json"	PASSED	0.378 s
3	Then User should receive status code 200	PASSED	0.001 s

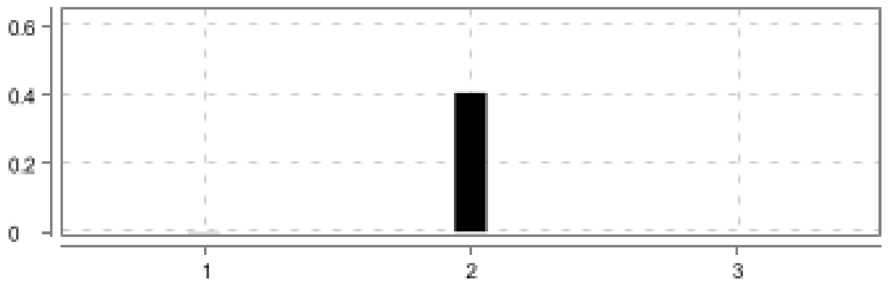
UPDATE user details with no request body

PASSED

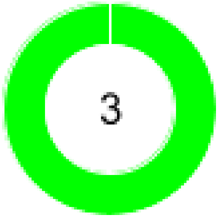
DURATION - 0.409 s

/ 1:07:53.365 PM // 1:07:53.774 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



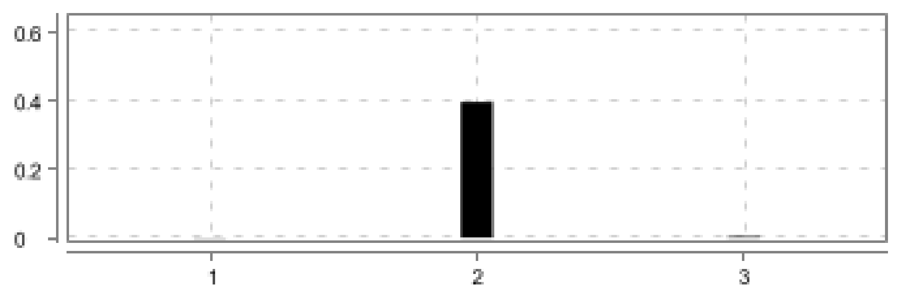
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request without request body	PASSED	0.406 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with duplicate contact number

FAILED *DURATION - 0.405 s*

/ 1:07:53.780 PM // 1:07:54.185 PM /

Update User Details API - PUT Request Scenarios



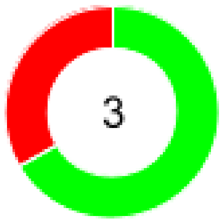
Steps

Total - 3

Pass - 2

Fail - 1

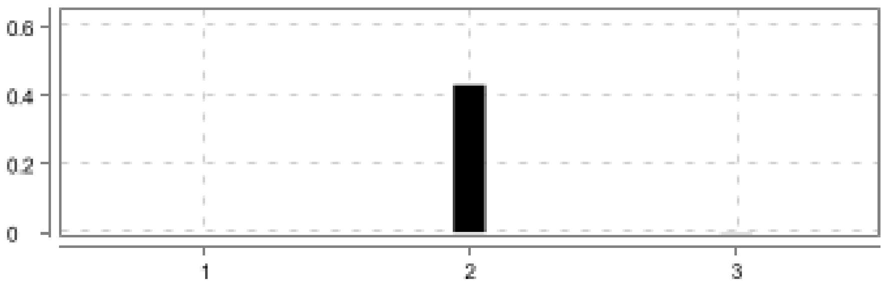
Skip - 0



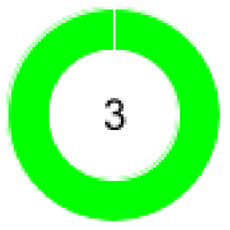
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "update_duplicate_contact.json"	PASSED	0.396 s
3	Then User should receive status code 409 java.lang.AssertionError: Expected status code 409 but got 200 expected [409] but found [200] at org.testng.Assert.fail(Assert.java:111) at org.testng.Assert.failNotEquals(Assert.java:1590) at org.testng.Assert.assertEqualsImpl(Assert.java:150) at org.testng.Assert.assertEquals(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:1431) at org.testng.Assert.assertEquals(Assert.java:1395) at StepDefinition.Put_UserId.validateStatusCode(Put_UserId.java:90) at ?.User should receive status code 409(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/Put_userid.feature:16) * Not displayable characters are replaced by '?'.	FAILED	0.004 s

Update user details with less than ten digits contact number

PASSED	*DURATION - 0.432 s*
/ 1:07:54.196 PM // 1:07:54.628 PM /	
Update User Details API - PUT Request Scenarios	



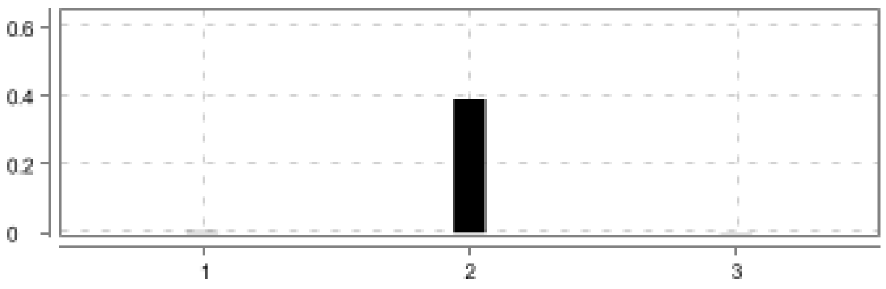
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



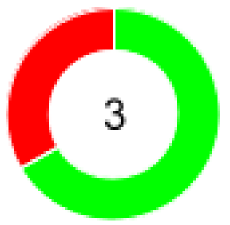
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "invalid_short_contact.json"	PASSED	0.430 s
3	Then User should receive status code 400	PASSED	0.001 s

UPDATE user details with duplicate email ID

FAILED	*DURATION - 0.398 s*
/ 1:07:54.640 PM // 1:07:55.038 PM /	
Update User Details API - PUT Request Scenarios	



Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.002 s
2	When User sends PUT request with JSON file "duplicate_email.json"	PASSED	0.390 s
3	Then User should receive status code 409	FAILED	0.001 s
	java.lang.AssertionError: Expected status code 409 but got 200 expected [409] but found [200] at org.testng.Assert.fail(Assert.java:111) at org.testng.Assert.failNotEquals(Assert.java:1590) at org.testng.Assert.assertEqualsImpl(Assert.java:150) at org.testng.Assert.assertEquals(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:1431) at org.testng.Assert.assertEquals(Assert.java:1395) at StepDefinition.Put_UserId.validateStatusCode(Put_UserId.java:90) at ?.User should receive status code 409(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/		

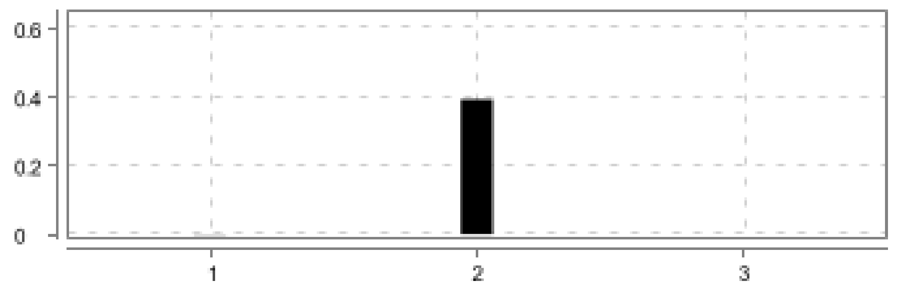
#	Step / Hook Details	Status	Duration
	Put_userid.feature:26) * Not displayable characters are replaced by '?'.		

UPDATE user detail with invalid emailID

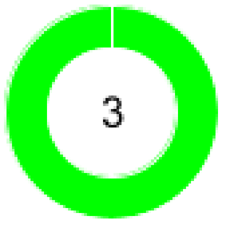
PASSED	*DURATION - 0.396 s*

/ 1:07:55.046 PM // 1:07:55.442 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



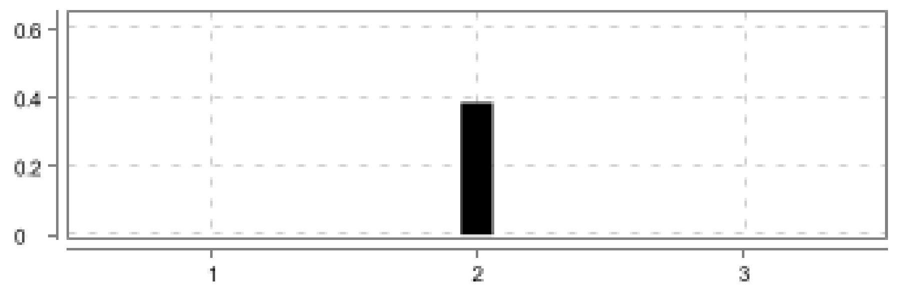
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "invalid_email.json"	PASSED	0.394 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with invalid plot number

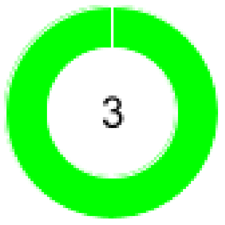
PASSED	*DURATION - 0.388 s*

/ 1:07:55.452 PM // 1:07:55.840 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



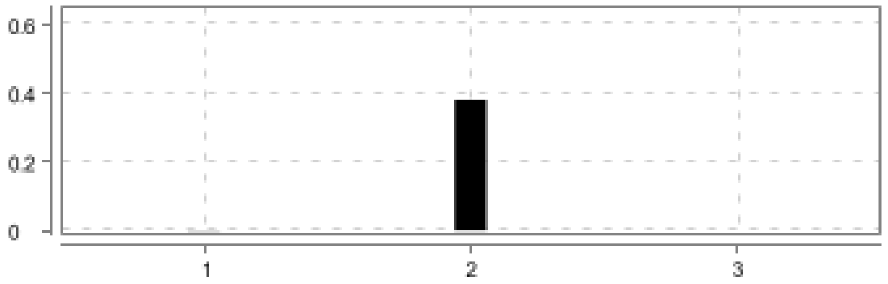
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "invalid_plotnumber.json"	PASSED	0.386 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with valid new plot number

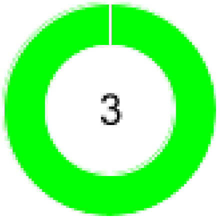
PASSED *DURATION - 0.386 s*

/ 1:07:55.848 PM // 1:07:56.234 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



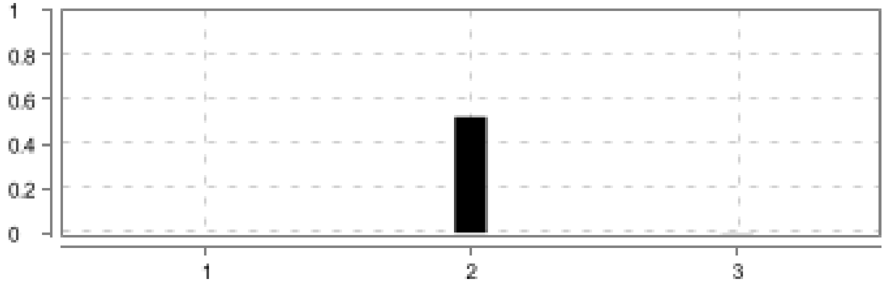
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "valid_plotnumber.json"	PASSED	0.382 s
3	Then User should receive status code 200	PASSED	0.000 s

UPDATE user details with invalid new street name

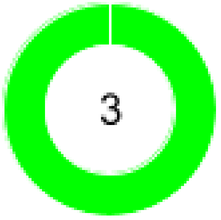
PASSED *DURATION - 0.524 s*

/ 1:07:56.244 PM // 1:07:56.768 PM /

Update User Details API - PUT Request Scenarios



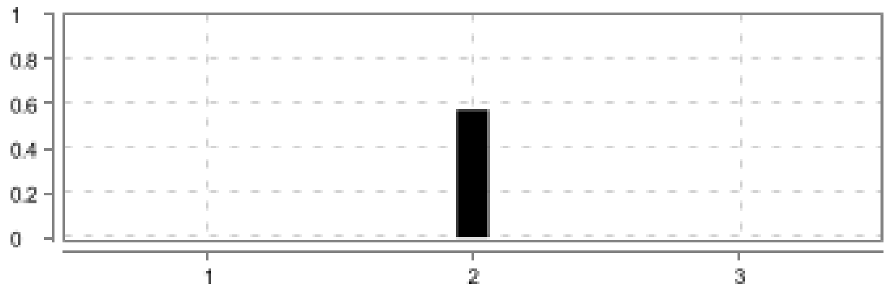
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



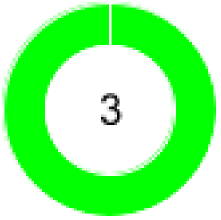
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "invalid_street.json"	PASSED	0.521 s
3	Then User should receive status code 400	PASSED	0.002 s

UPDATE user details with valid new street name

PASSED	DURATION - 0.574 s
/ 1:07:56.775 PM // 1:07:57.349 PM /	
Update User Details API - PUT Request Scenarios	



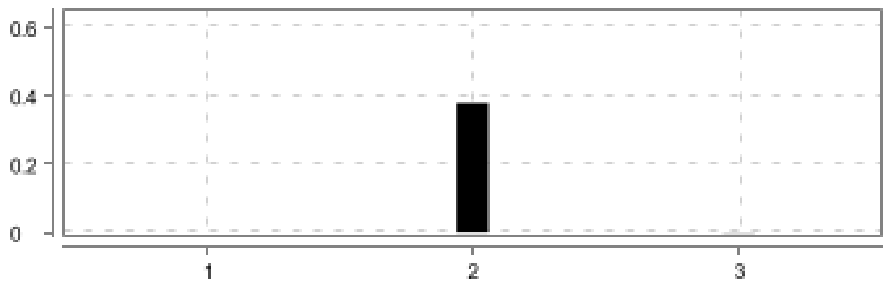
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



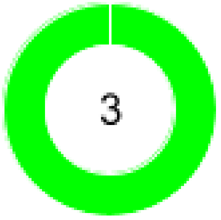
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "valid_street.json"	PASSED	0.573 s
3	Then User should receive status code 200	PASSED	0.000 s

UPDATE user details with invalid country name

PASSED	DURATION - 0.382 s
/ 1:07:57.355 PM // 1:07:57.737 PM /	
Update User Details API - PUT Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



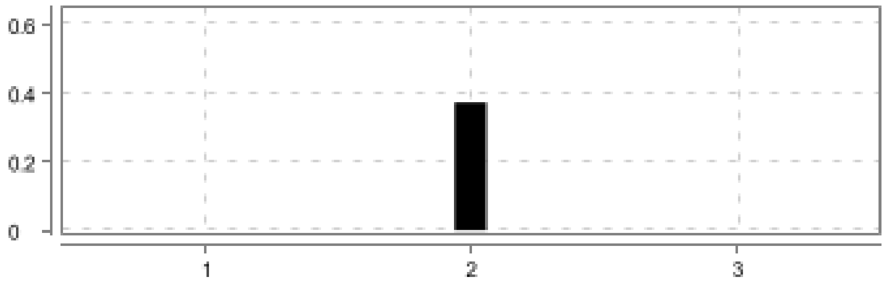
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "invalid_country.json"	PASSED	0.378 s
3	Then User should receive status code 400	PASSED	0.001 s

UPDATE user details with valid new country name

PASSED *DURATION - 0.376 s*

/ 1:07:57.742 PM // 1:07:58.118 PM /

Update User Details API - PUT Request Scenarios



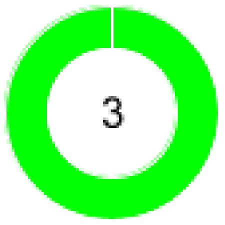
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



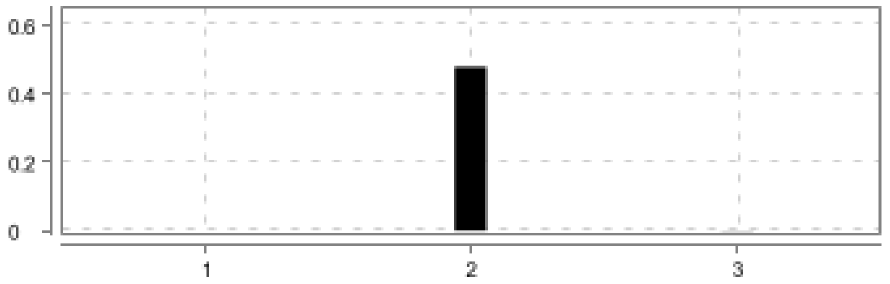
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "valid_country.json"	PASSED	0.374 s
3	Then User should receive status code 200	PASSED	0.000 s

UPDATE user details with invalid zipcode

PASSED *DURATION - 0.480 s*

/ 1:07:58.125 PM // 1:07:58.605 PM /

Update User Details API - PUT Request Scenarios



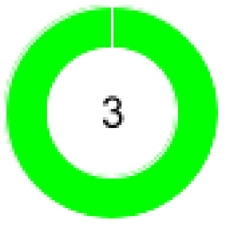
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



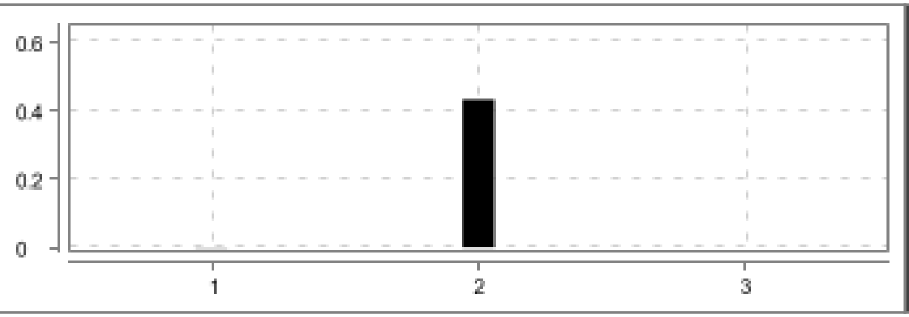
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "invalid_zipcode.json"	PASSED	0.478 s
3	Then User should receive status code 400	PASSED	0.001 s

UPDATE user details with valid new zipcode

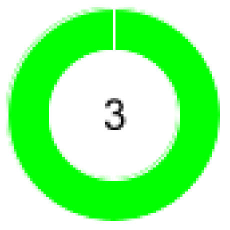
PASSED *DURATION - 0.435 s*

/ 1:07:58.612 PM // 1:07:59.047 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



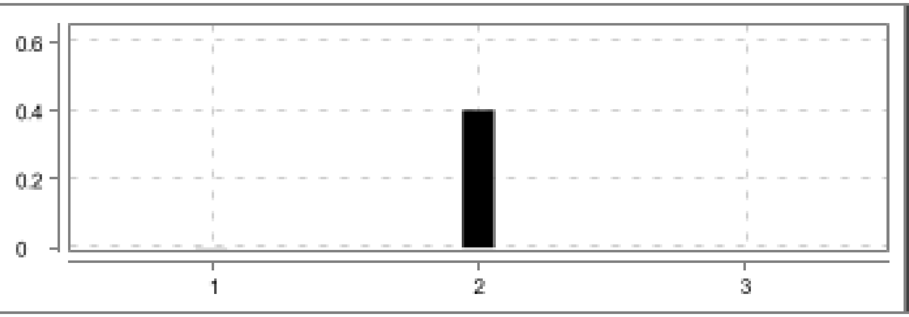
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "valid_zipcode.json"	PASSED	0.433 s
3	Then User should receive status code 200	PASSED	0.000 s

UPDATE user details with invalid firstname

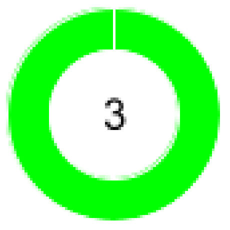
PASSED *DURATION - 0.404 s*

/ 1:07:59.058 PM // 1:07:59.462 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



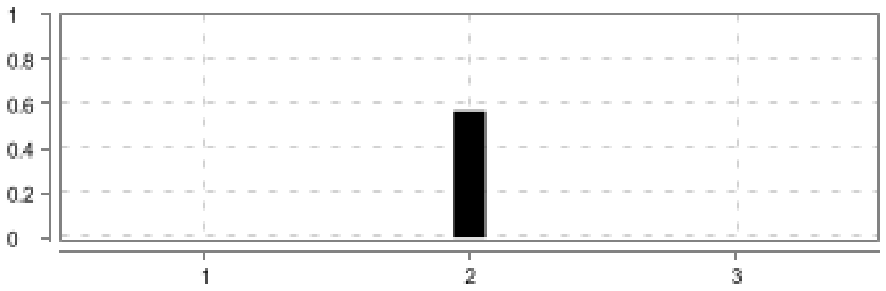
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "invalid_firstname.json"	PASSED	0.402 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with valid firstname

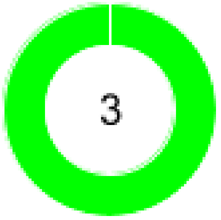
PASSED *DURATION - 0.569 s*

/ 1:07:59.465 PM // 1:08:00.034 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



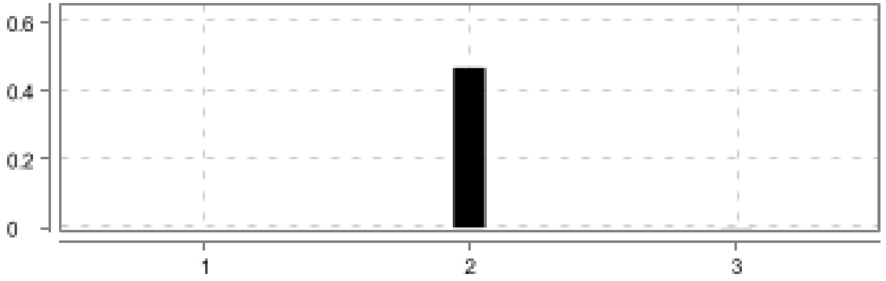
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "valid_firstname.json"	PASSED	0.568 s
3	Then User should receive status code 200	PASSED	0.000 s

UPDATE user details with invalid lastname

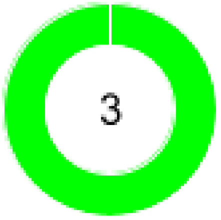
PASSED *DURATION - 0.469 s*

/ 1:08:00.039 PM // 1:08:00.508 PM /

Update User Details API - PUT Request Scenarios



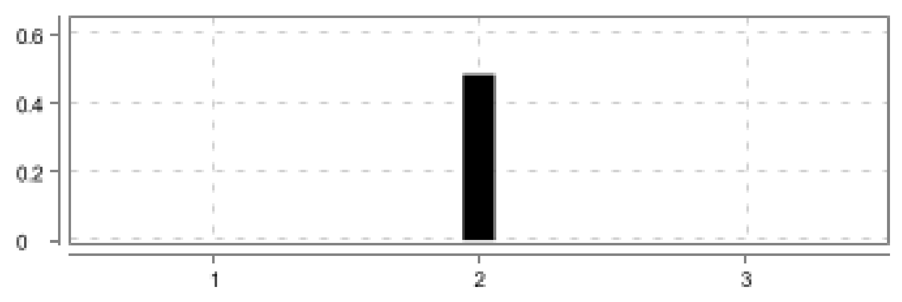
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



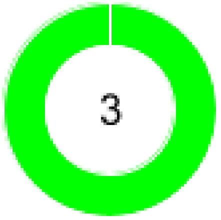
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "invalid_lastname.json"	PASSED	0.467 s
3	Then User should receive status code 400	PASSED	0.001 s

UPDATE user details with valid lastname

PASSED	DURATION - 0.486 s
/ 1:08:00.513 PM // 1:08:00.999 PM /	
Update User Details API - PUT Request Scenarios	



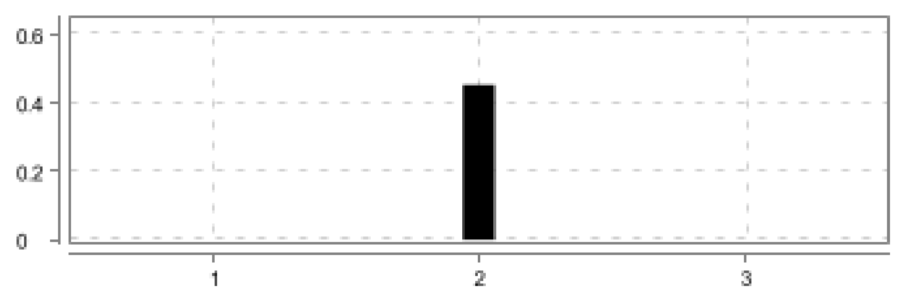
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



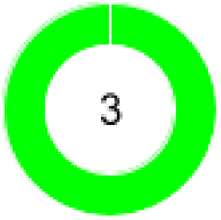
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "valid_lastname.json"	PASSED	0.484 s
3	Then User should receive status code 200	PASSED	0.000 s

UPDATE user details with empty contact number

PASSED	DURATION - 0.455 s
/ 1:08:01.003 PM // 1:08:01.458 PM /	
Update User Details API - PUT Request Scenarios	



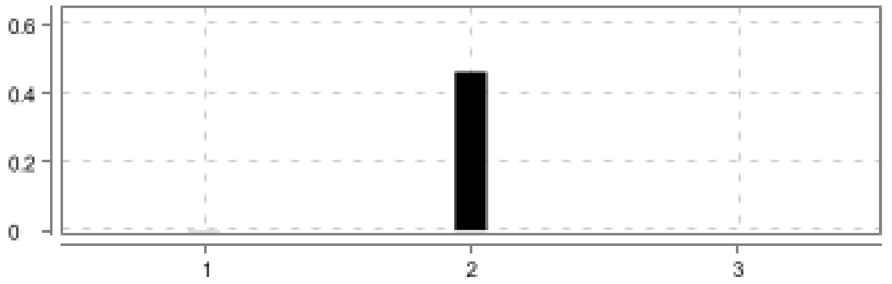
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



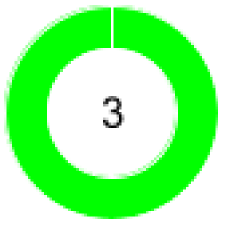
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "empty_contact.json"	PASSED	0.453 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with empty emailID

PASSED	*DURATION - 0.466 s*
/ 1:08:01.464 PM // 1:08:01.930 PM /	
Update User Details API - PUT Request Scenarios	



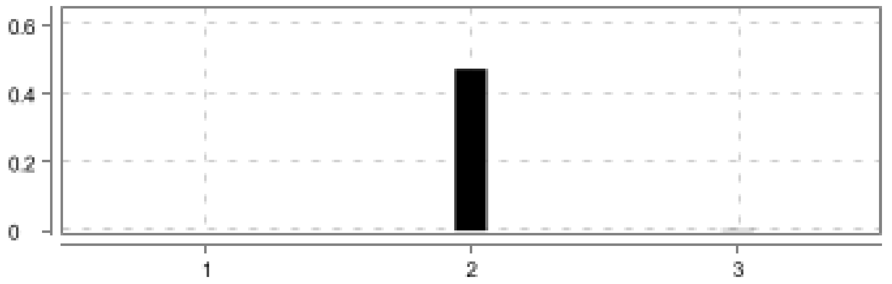
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



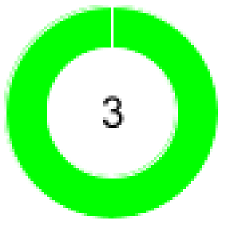
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "empty_email.json"	PASSED	0.462 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with empty firstname

PASSED	*DURATION - 0.478 s*
/ 1:08:01.935 PM // 1:08:02.413 PM /	
Update User Details API - PUT Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



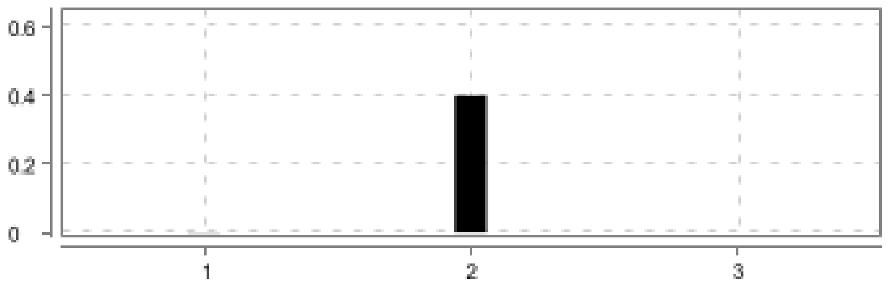
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "empty_firstname.json"	PASSED	0.473 s
3	Then User should receive status code 400	PASSED	0.002 s

UPDATE user details with empty lastname

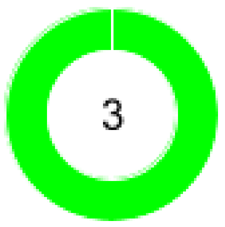
PASSED *DURATION - 0.403 s*

/ 1:08:02.417 PM // 1:08:02.820 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



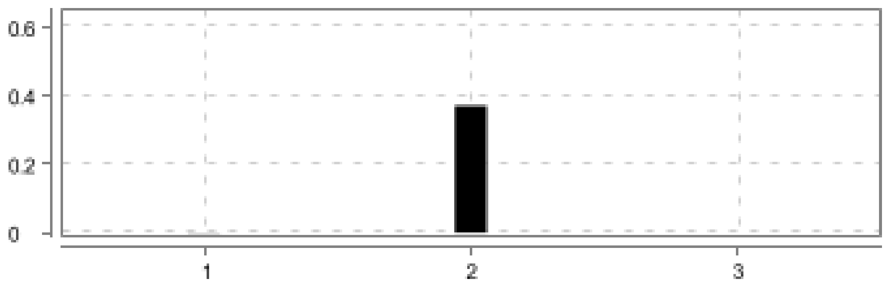
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "empty_lastname.json"	PASSED	0.399 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with empty plotnumber

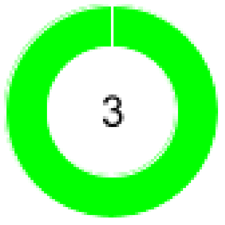
PASSED *DURATION - 0.379 s*

/ 1:08:02.830 PM // 1:08:03.209 PM /

Update User Details API - PUT Request Scenarios



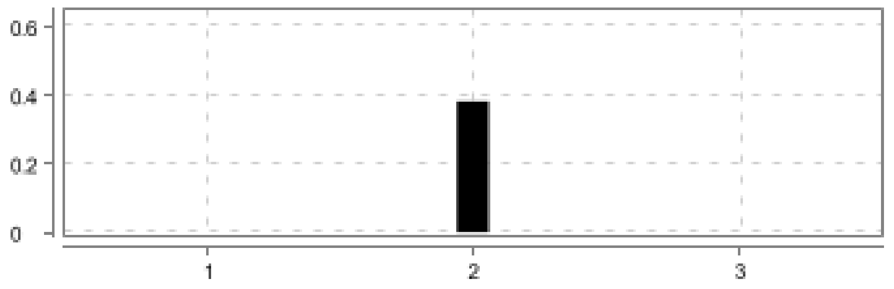
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



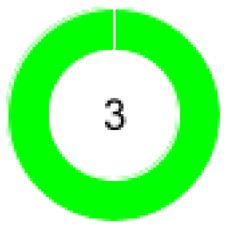
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "empty_plotnumber.json"	PASSED	0.371 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with empty street

PASSED	*DURATION - 0.385 s*
/ 1:08:03.217 PM // 1:08:03.602 PM /	
Update User Details API - PUT Request Scenarios	



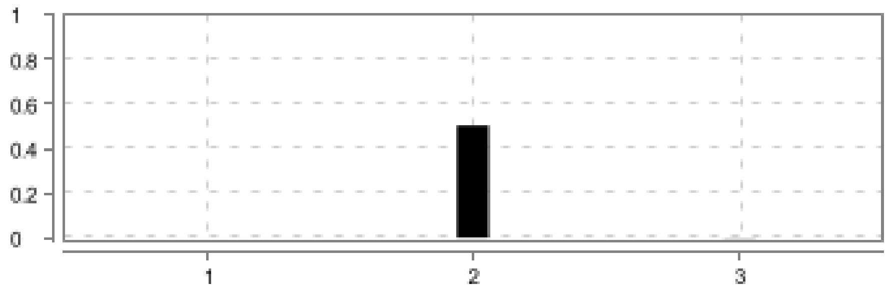
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



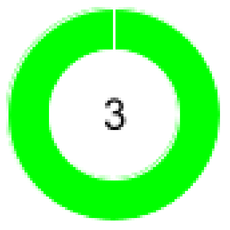
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "empty_street.json"	PASSED	0.383 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with empty country

PASSED	*DURATION - 0.510 s*
/ 1:08:03.609 PM // 1:08:04.119 PM /	
Update User Details API - PUT Request Scenarios	



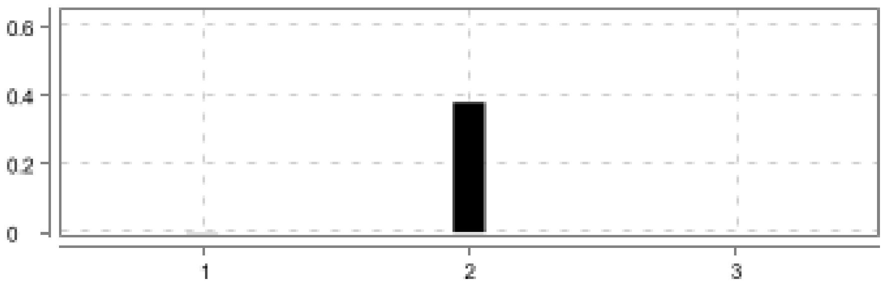
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



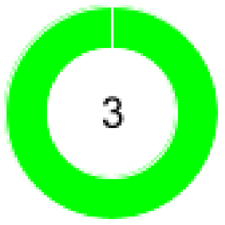
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.000 s
2	When User sends PUT request with JSON file "empty_country.json"	PASSED	0.505 s
3	Then User should receive status code 400	PASSED	0.001 s

UPDATE user details with empty zipcode

PASSED	*DURATION - 0.383 s*
/ 1:08:04.129 PM // 1:08:04.512 PM /	
Update User Details API - PUT Request Scenarios	



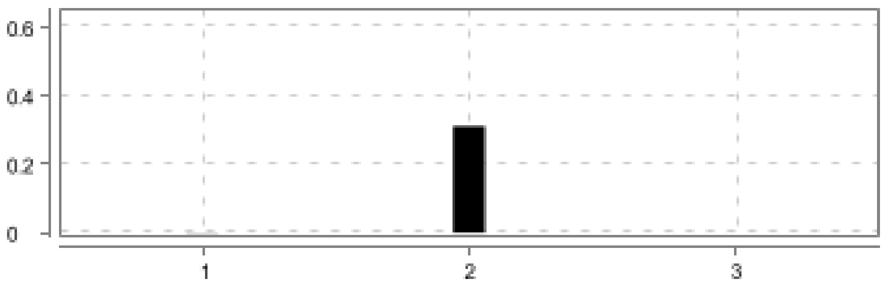
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



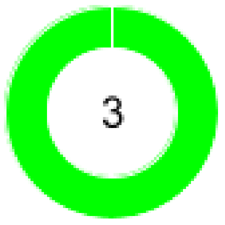
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "empty_zipcode.json"	PASSED	0.380 s
3	Then User should receive status code 400	PASSED	0.000 s

UPDATE user details with invalid BaseURL

PASSED	*DURATION - 0.312 s*
/ 1:08:04.519 PM // 1:08:04.831 PM /	
Update User Details API - PUT Request Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



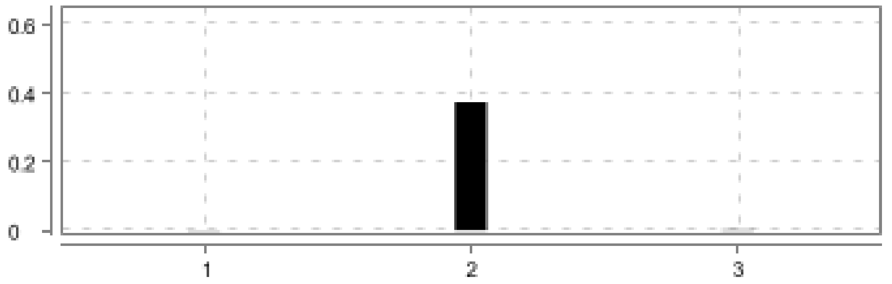
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and invalid BaseURL and valid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "update_valid_all_fields.json"	PASSED	0.311 s
3	Then User should receive status code 404	PASSED	0.000 s

UPDATE user details with invalid endpoint

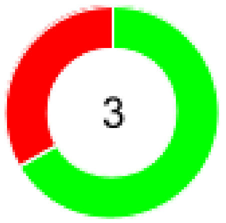
FAILED	*DURATION - 0.379 s*

/ 1:08:04.835 PM // 1:08:05.214 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



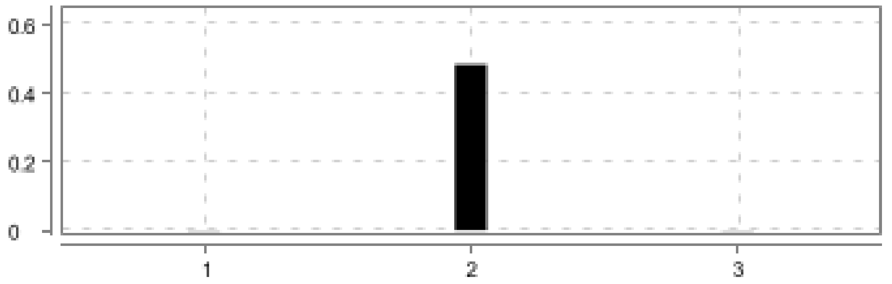
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and invalid Endpoint	PASSED	0.001 s
2	When User sends PUT request with JSON file "update_valid_all_fields.json"	PASSED	0.375 s
3	Then User should receive status code 404	FAILED	0.002 s
	java.lang.AssertionError: Expected status code 404 but got 400 expected [404] but found [400] at org.testng.Assert.fail(Assert.java:111) at org.testng.Assert.failNotEquals(Assert.java:1590) at org.testng.Assert.assertEqualsImpl(Assert.java:150) at org.testng.Assert.assertEquals(Assert.java:132) at org.testng.Assert.assertEquals(Assert.java:1431) at org.testng.Assert.assertEquals(Assert.java:1395) at StepDefinition.Put_UserId.validateStatusCode(Put_UserId.java:90) at ?.User should receive status code 404(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/Put_userid.feature:141) * Not displayable characters are replaced by '?'.		

UPDATE user details with empty endpoint

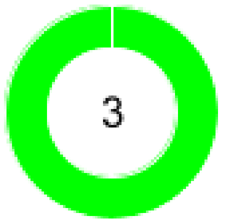
PASSED	*DURATION - 0.486 s*

/ 1:08:05.219 PM // 1:08:05.705 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with authorization and valid BaseURL and empty Endpoint	PASSED	0.001 s

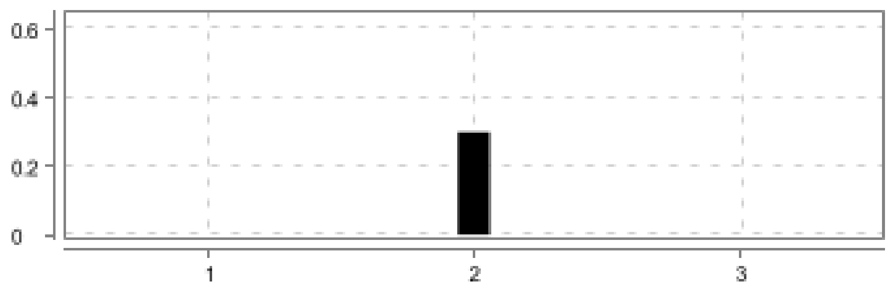
#	Step / Hook Details	Status	Duration
2	When User sends PUT request with JSON file "update_valid_all_fields.json"	PASSED	0.484 s
3	Then User should receive status code 404	PASSED	0.001 s

Add a New User with all fields without Authorization

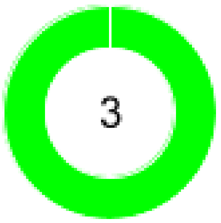
PASSED *DURATION - 0.301 s*

/ 1:08:05.709 PM // 1:08:06.010 PM /

Update User Details API - PUT Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



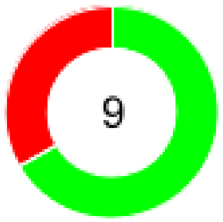
#	Step / Hook Details	Status	Duration
1	Given User prepares PUT request with valid BaseURL and valid Endpoint without Auth	PASSED	0.000 s
2	When User sends PUT request with JSON file "update_valid_all_fields.json"	PASSED	0.300 s
3	Then User should receive status code 401	PASSED	0.000 s

Get All Users API - GET Request Scenarios

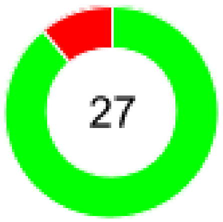
FAILED *DURATION - 10.337 s*

/ 1:08:06.026 PM // 1:08:16.363 PM /

Scenarios
Total - 9
Pass - 6
Fail - 3
Skip - 0



Steps
Total - 27
Pass - 24
Fail - 3
Skip - 0

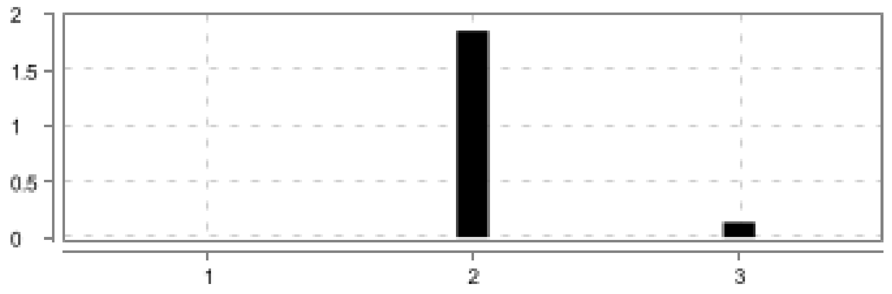


Get All Users with Valid Base URL and Endpoint

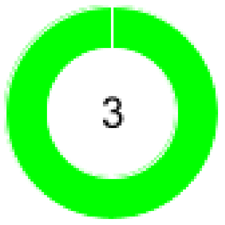
PASSED *DURATION - 1.988 s*

/ 1:08:06.026 PM // 1:08:08.014 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



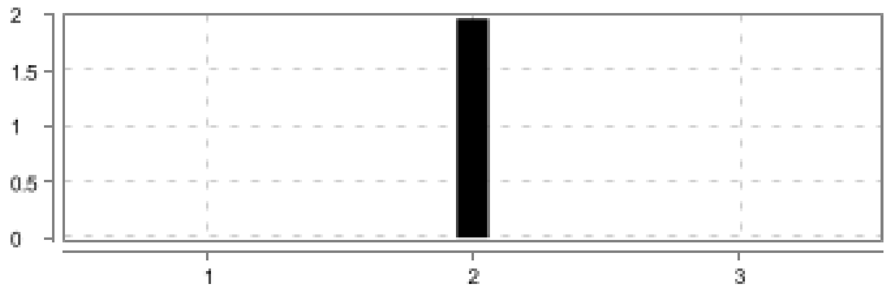
#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request with valid BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends GET request to fetch users	PASSED	1.854 s
3	Then User receives 200 OK status with list of users	PASSED	0.134 s

Validate Response Body Content-Type is JSON

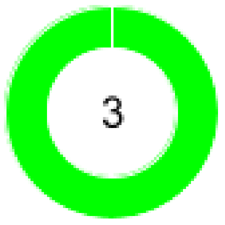
PASSED *DURATION - 1.970 s*

/ 1:08:08.017 PM // 1:08:09.987 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



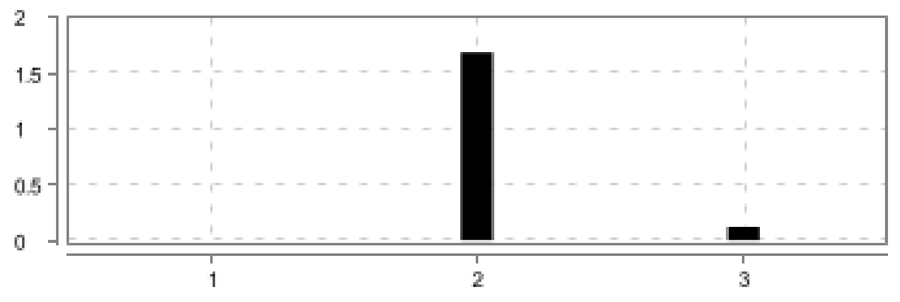
#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request with valid BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends GET request to fetch users	PASSED	1.969 s
3	Then Response Content-Type is JSON with the status code 200 OK	PASSED	0.000 s

Validate Response Contains Required Fields

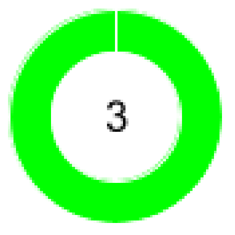
PASSED *DURATION - 1.814 s*

/ 1:08:09.989 PM // 1:08:11.803 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



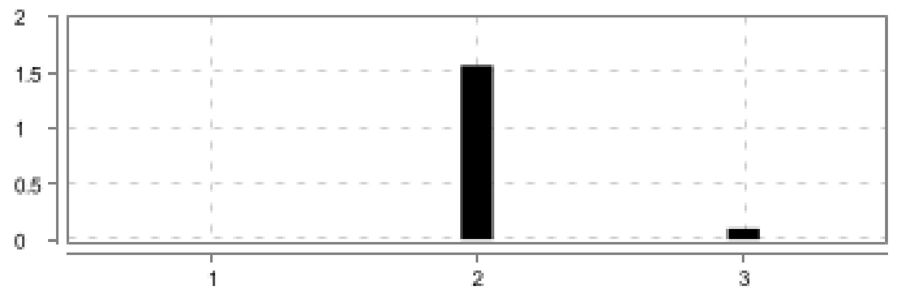
#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request with valid BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends GET request to fetch users	PASSED	1.690 s
3	Then Response contains user fields: user_id, UserfirstName, UserlastName, userContactNumber, userEmailId, creationTime, lastModTime, userAddress, addressId, plotNumber, street, state, country, zipCode	PASSED	0.121 s

Validate user_id is integer type in response

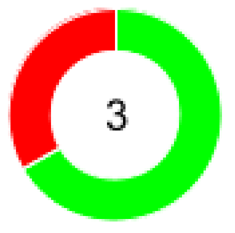
FAILED *DURATION - 1.668 s*

/ 1:08:11.805 PM // 1:08:13.473 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request with valid BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends GET request to fetch users	PASSED	1.566 s
3	Then user_id is of integer type with the status code 200 OK java.lang.AssertionError: expected [true] but found [false] at org.testng.Assert.fail(Assert.java:111) at org.testng.Assert.failNotEquals(Assert.java:1590) at org.testng.Assert.assertTrue(Assert.java:57) at org.testng.Assert.assertTrue(Assert.java:67) at StepDefinition.Get_Users.validate_user_id_type_integer(Get_Users.java:86) at ?.user_id is of integer type with the status code 200 OK(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/get.allusers.feature:21)	FAILED	0.100 s

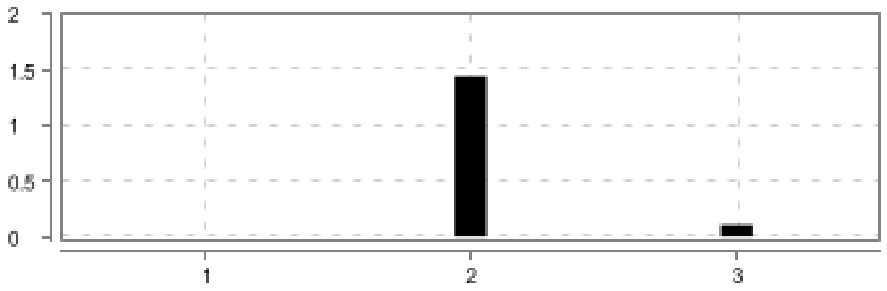
#	Step / Hook Details	Status	Duration
	* Not displayable characters are replaced by '?'.		

Validate address_id is integer type in response

FAILED	*DURATION - 1.553 s*

/ 1:08:13.477 PM // 1:08:15.030 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0

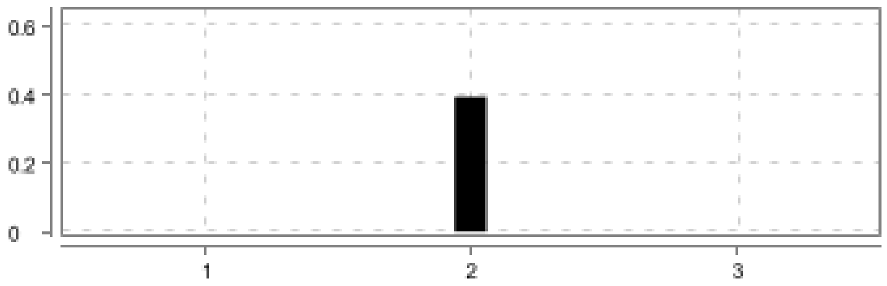
#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request with valid BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends GET request to fetch users	PASSED	1.444 s
3	Then address_id is of integer type with the status code 200 OK	FAILED	0.106 s
	java.lang.AssertionError: addressId should be a number expected [true] but found [false] at org.testng.Assert.fail(Assert.java:111) at org.testng.Assert.failNotEquals(Assert.java:1590) at org.testng.Assert.assertTrue(Assert.java:57) at StepDefinition.Get_Users.validate_address_id_type_integer(Get_Users.java:93) at ?.address_id is of integer type with the status code 200 OK(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/get.allusers.feature:26) * Not displayable characters are replaced by '?'.		

Get Users with Invalid Endpoint

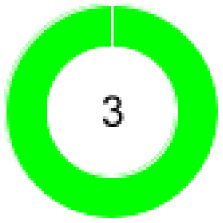
PASSED	*DURATION - 0.396 s*

/ 1:08:15.034 PM // 1:08:15.430 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



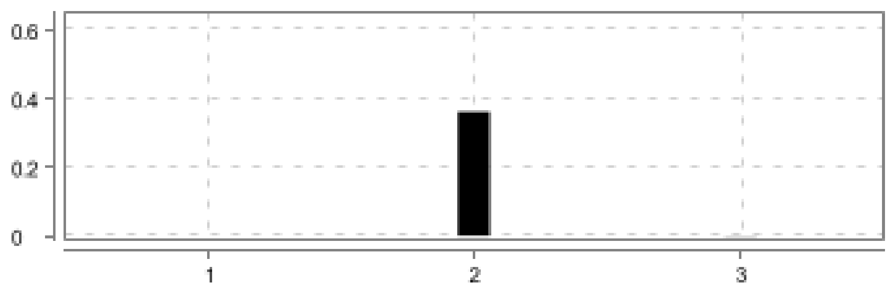
#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request with valid BaseURL and invalid endpoint	PASSED	0.000 s
2	When User sends GET request to fetch users	PASSED	0.394 s
3	Then User receives 404 Not Found status	PASSED	0.000 s

Get Users with Empty Endpoint

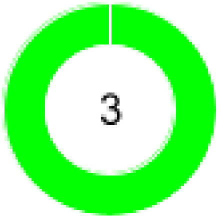
PASSED	*DURATION - 0.366 s*

/ 1:08:15.433 PM // 1:08:15.799 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



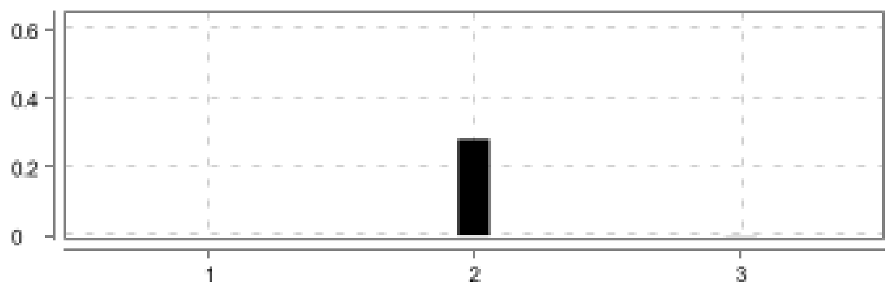
#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request with valid BaseURL and empty endpoint	PASSED	0.000 s
2	When User sends GET request to fetch users	PASSED	0.362 s
3	Then User receives 404 Not Found status	PASSED	0.001 s

Get USERS without Authorization and receives 401

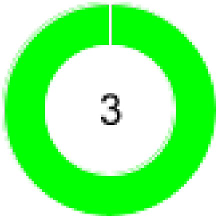
PASSED	*DURATION - 0.282 s*

/ 1:08:15.803 PM // 1:08:16.085 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



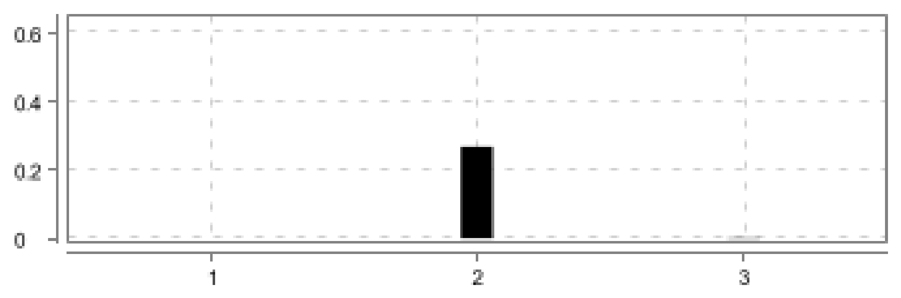
#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request without Authorization header	PASSED	0.000 s
2	When User sends GET request to fetch users without Auth	PASSED	0.280 s
3	Then User receives 401 Unauthorized status	PASSED	0.001 s

Get Users with Invalid Base URL

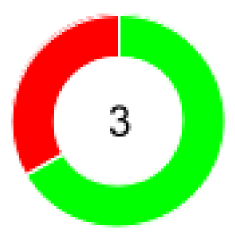
FAILED *DURATION - 0.273 s*

/ 1:08:16.090 PM // 1:08:16.363 PM /

Get All Users API - GET Request Scenarios



Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



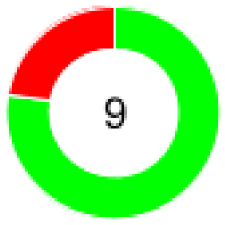
#	Step / Hook Details	Status	Duration
1	Given User prepares the GET request with invalid BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends GET request to fetch users	PASSED	0.269 s
3	Then User receives 404 Not Found status with error message	FAILED	0.002 s
	java.lang.AssertionError: expected [true] but found [false] at org.testng.Assert.fail(Assert.java:111) at org.testng.Assert.failNotEquals(Assert.java:1590) at org.testng.Assert.assertTrue(Assert.java:57) at org.testng.Assert.assertTrue(Assert.java:67) at StepDefinition.Get_Users.validate_404_with_invalid_baseurl(Get_Users.java:142) at ?.User receives 404 Not Found status with error message(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/get.allusers.feature:46) * Not displayable characters are replaced by '?'.		

Add New User API - Positive Scenarios

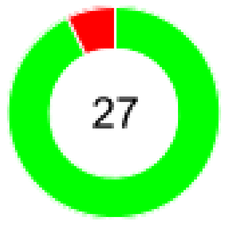
FAILED *DURATION - 6.980 s*

/ 1:08:16.372 PM // 1:08:23.352 PM /

Scenarios
Total - 9
Pass - 7
Fail - 2
Skip - 0



Steps
Total - 27
Pass - 25
Fail - 2
Skip - 0

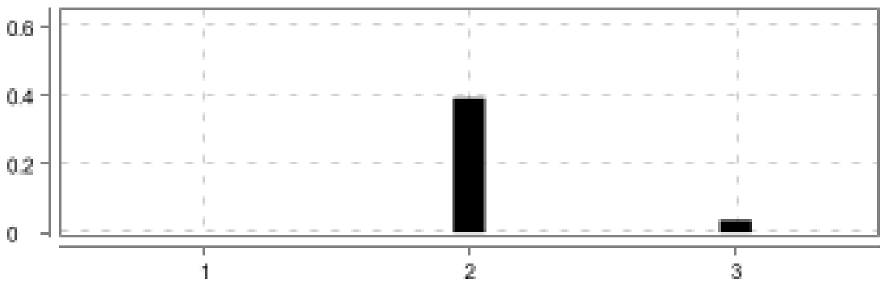


Add a new user with all mandatory and optional fields

PASSED *DURATION - 0.792 s*

/ 1:08:16.372 PM // 1:08:17.164 PM /

Add New User API - Positive Scenarios



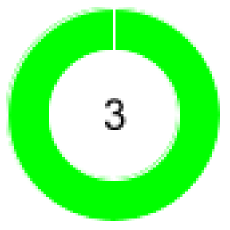
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



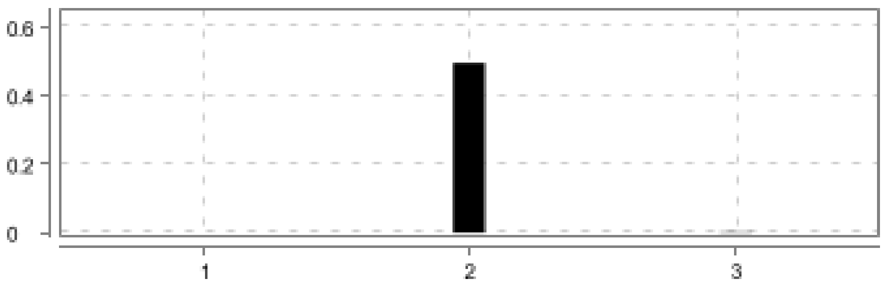
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with all required and optional fields	PASSED	0.392 s
3	Then User receives 201 Created status with matching fields	PASSED	0.036 s

Add a new user with only mandatory fields

FAILED *DURATION - 0.498 s*

/ 1:08:17.170 PM // 1:08:17.668 PM /

Add New User API - Positive Scenarios



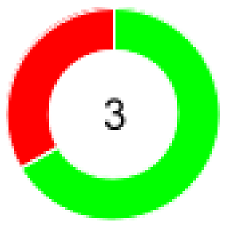
Steps

Total - 3

Pass - 2

Fail - 1

Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with only mandatory fields	PASSED	0.496 s
3	Then User receives 201 Created status	FAILED	0.002 s

```
java.lang.AssertionError: Expected status code 201 but got 500 expected [201] but found [500]
    at org.testng.Assert.fail(Assert.java:111)
    at org.testng.Assert.failNotEquals(Assert.java:1590)
    at org.testng.Assert.assertEqualsImpl(Assert.java:150)
    at org.testng.Assert.assertEquals(Assert.java:132)
    at org.testng.Assert.assertEquals(Assert.java:1431)
    at org.testng.Assert.assertEquals(Assert.java:1395)
    at StepDefinition.PostPositive_CreateUsers.validateResponse(PostPositive_CreateUsers.java:112)
    at ?.User receives 201 Created status(file:///C:/Users/chand/eclipse-workspace/API_RESTASSURED/src/test/resources/Features/
```

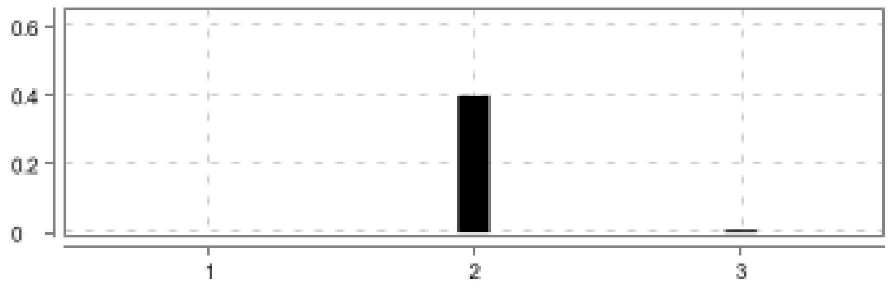
#	Step / Hook Details	Status	Duration
	post_positive.feature:11) * Not displayable characters are replaced by '?'.		

Validate autogenerated user_id in response

PASSED	*DURATION - 0.796 s*

/ 1:08:17.671 PM // 1:08:18.467 PM /

Add New User API - Positive Scenarios



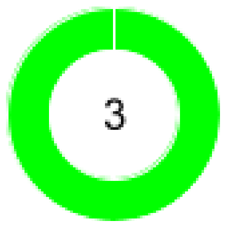
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



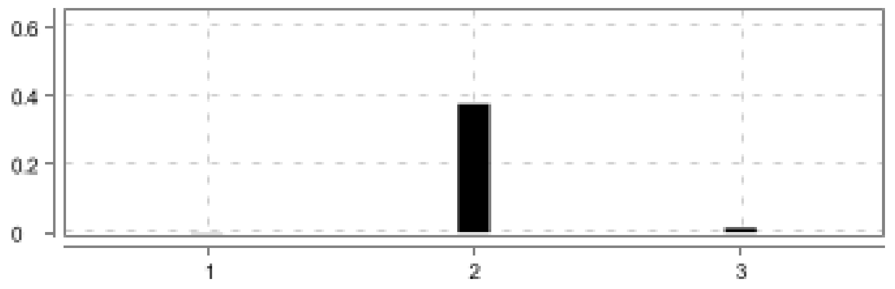
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with all required and optional fields	PASSED	0.398 s
3	Then User sees autogenerated user_id in the response body	PASSED	0.007 s

Validate autogenerated addressID in response

PASSED	*DURATION - 0.731 s*

/ 1:08:18.473 PM // 1:08:19.204 PM /

Add New User API - Positive Scenarios



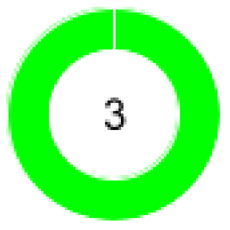
Steps

Total - 3

Pass - 3

Fail - 0

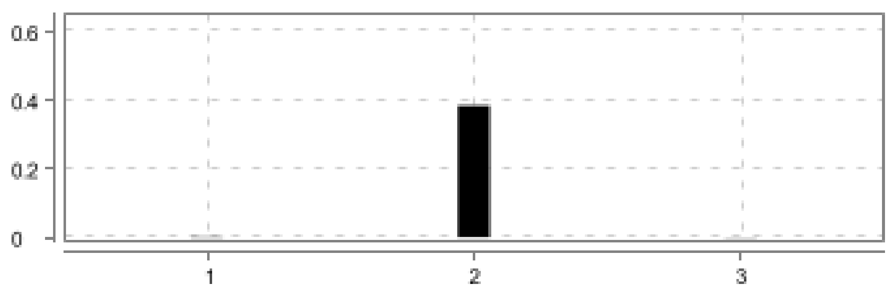
Skip - 0



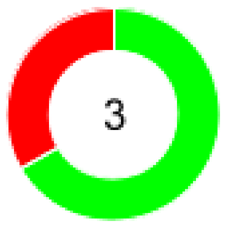
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.001 s
2	When User sends the POST request with all required and optional fields	PASSED	0.376 s
3	Then User sees autogenerated addressID in the response body	PASSED	0.013 s

Validate Content-Type is application/json in response

FAILED	*DURATION - 1.072 s*
/ 1:08:19.262 PM // 1:08:20.334 PM /	
Add New User API - Positive Scenarios	



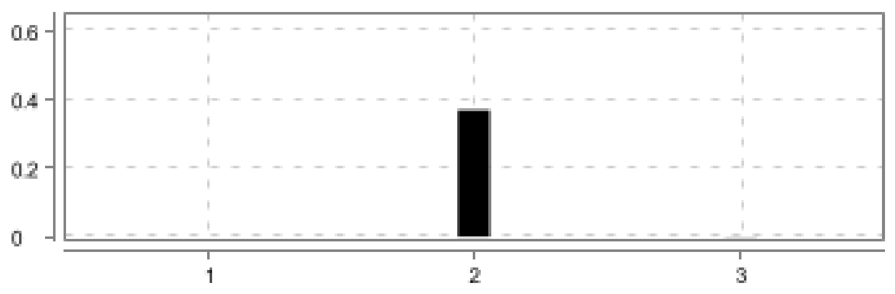
Steps
Total - 3
Pass - 2
Fail - 1
Skip - 0



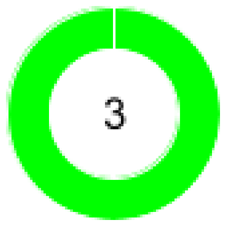
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.002 s
2	When User sends the POST request with all required and optional fields	PASSED	0.385 s
3	Then User receives Content-Type as application/json with status code 201 Step undefined	FAILED	0.001 s

Validate response status code is 201

PASSED	*DURATION - 0.742 s*
/ 1:08:20.351 PM // 1:08:21.093 PM /	
Add New User API - Positive Scenarios	



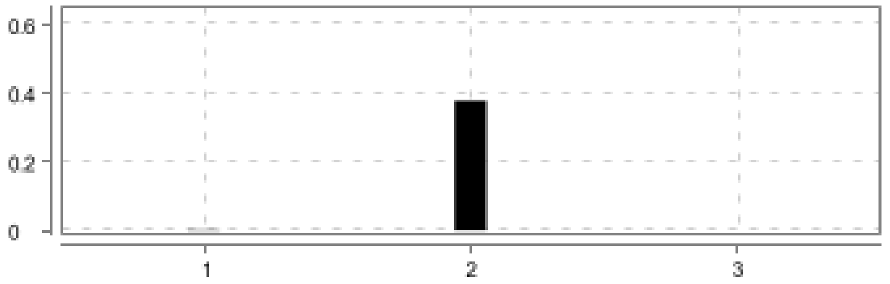
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



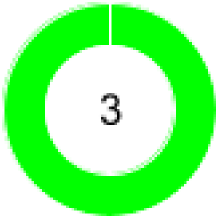
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with all required and optional fields	PASSED	0.370 s
3	Then User receives status code 201	PASSED	0.001 s

Validate response status text is Created

PASSED	*DURATION - 0.735 s*
/ 1:08:21.118 PM // 1:08:21.853 PM /	
Add New User API - Positive Scenarios	



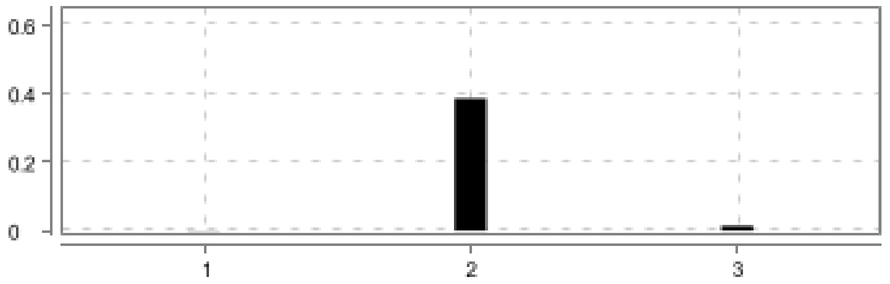
Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



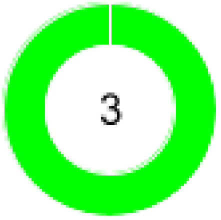
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.002 s
2	When User sends the POST request with all required and optional fields	PASSED	0.379 s
3	Then User receives status text as Created	PASSED	0.000 s

Validate user_id is integer in response

PASSED	*DURATION - 0.748 s*
/ 1:08:21.873 PM // 1:08:22.621 PM /	
Add New User API - Positive Scenarios	



Steps
Total - 3
Pass - 3
Fail - 0
Skip - 0



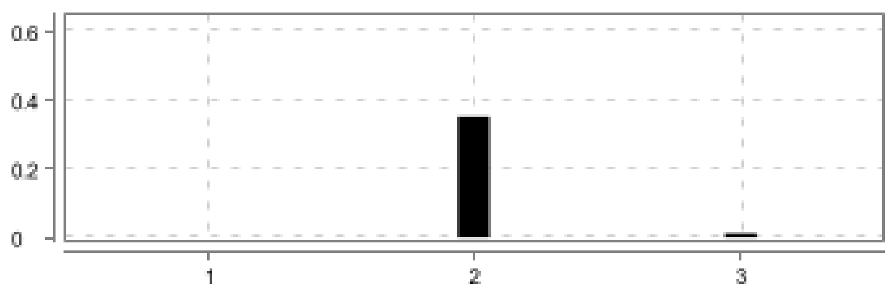
#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.001 s
2	When User sends the POST request with all required and optional fields	PASSED	0.386 s
3	Then User sees user_id is of integer type with status 201	PASSED	0.013 s

Validate addressID is integer in response

PASSED	*DURATION - 0.720 s*

/ 1:08:22.632 PM // 1:08:23.352 PM /

Add New User API - Positive Scenarios



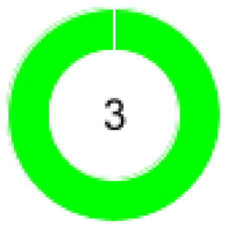
Steps

Total - 3

Pass - 3

Fail - 0

Skip - 0



#	Step / Hook Details	Status	Duration
1	Given User prepares the POST request with BaseURL and valid endpoint	PASSED	0.000 s
2	When User sends the POST request with all required and optional fields	PASSED	0.353 s
3	Then User sees addressID is of integer type with status 201	PASSED	0.011 s