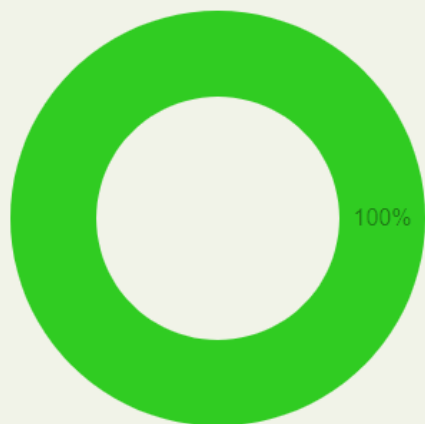# Serenity BDD

Test Results: All Tests

6 test scenarios (including 6 rows of test data)

🏠 Summary   🎛 Test Results

Scenario Results (including rows of test data)	Automated	
⊘ Passing	6	100%
◉ Pending	0	
⊘ Ignored	0	
⏭ Skipped	0	
✖ *Unsuccessful*		
⊗ Failed	0	
⚠ Broken	0	
⤬ Compromised	0	
Total	**6**	

100%

Functional Coverage Overview

Features

Feature	Scenarios	% Pass	Result	Coverage
Get a single character by id	6	100%	⊘	
Get a single character by id	6	100%	⊘	

List character info successfully

Given the user is in the Marvel developer portal
When the user sends a request with a valid character id "{id}"
Then the user gets a status code response OK and can see the character's name "{name}" and "{id}"

Examples:

id	name	
1011136	Air-Walker (Gabriel Lan)	
1011334	3-D Man	
1017100	A-Bomb (HAS)	

Failed request

Given a user is in the Marvel developer site
When the user sends an invalid request, with an id "{Id}", a token "{token}" and a hash "{hash}"
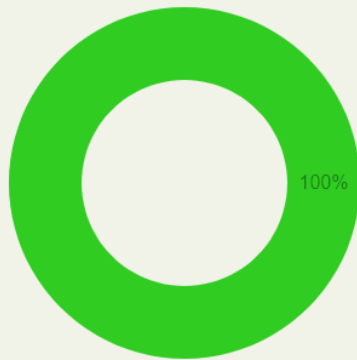**Then** the user gets a status code response {code}

Examples:

Id	token	hash	code	
1234	f0a55c0526a2a7495cce01466a651fa6	b7026387801d6606abe480e0dd879b2e	404	
1011136	aaa	b7026387801d6606abe480e0dd879b2e	401	
1234	aaa	myhash	401	

Test Results: All Tests

4 test scenarios (including 3 rows of test data)

🏠 Summary   🕐 Test Results

Scenario Results (including rows of test data)	Automated	
⊘ Passing	4	100%
◉ Pending	0	
🚫 Ignored	0	
⏭ Skipped	0	
✖ *Unsuccessful*		
❌ Failed	0	
⚠ Broken	0	
✗ Compromised	0	
Total	**4**	

Scenario details

Create user successfully ✅

When the user sends a request with the user's name "{name}" and job "{job}"
Then the user gets a status code response Created, and can see the user's id, name "{name}", job "{job}" and creation time

Examples:

name	job	
Andres	QA	
Juliana	Developer	
Mario	PO	

Empty user creation ✅

Given the user is in the reqres web page
When the user sends an empty request
Then the user gets a status code Created, an can see the user's id and creation time

Scenarios

2023-03-29 01:28:11 p. m.

[1.1:14] Create posts successfully

[1.2:15] Create posts successfully

[1.3:16] Create posts successfully

[1.4:17] Create posts successfully

[1.5:18] Create posts successfully

Summary | Tags | Feature: **src/test/java/features/createpost.feature** | **create post** *AS a json place holder user I WANT TO make a request to create the Post service SO THAT I CAN create a new post*

Scenario: [1.1:14] **Create posts successfully**	ms: 996
7 Given url 'https://jsonplaceholder.typicode.com/posts'	73
8 When request {"title":"my first post", "body": "hello, this is my first post ever", "userId": 1}	30
9 And method post	845
10 Then status 201	0
11 And match response == {"title": "my first post", "userId": 1,"body": "hello, this is my first post ever", "id": #ignore}	48

Scenario: [1.2:15] **Create posts successfully**	ms: 252
7 Given url 'https://jsonplaceholder.typicode.com/posts'	1
8 When request {"title":"hello", "body": "I just want to say hello!!!", "userId": 3}	0
9 And method post	251
10 Then status 201	0
11 And match response == {"title": "hello", "userId": 3,"body": "I just want to say hello!!!", "id": #ignore}	0

Scenario: [1.3:16] **Create posts successfully**	ms: 264
7 Given url 'https://jsonplaceholder.typicode.com/posts'	0
8 When request {"title":"karate", "body": "I'm learning karate right now", "userId": 2}	0
9 And method post	264
10 Then status 201	0
11 And match response == {"title": "karate", "userId": 2,"body": "I'm learning karate right now", "id": #ignore}	0

Scenario: [1.4:17] **Create posts successfully**	ms: 292
7 Given url 'https://jsonplaceholder.typicode.com/posts'	0
8 When request {"title":"", "body": "", "userId": 2000}	0
9 And method post	291
10 Then status 201	0
11 And match response == {"title": "", "userId": 2000,"body": "", "id": #ignore}	1

Scenario: [1.5:18] **Create posts successfully**	ms: 268
7 Given url 'https://jsonplaceholder.typicode.com/posts'	0
8 When request {"title":"", "body": "", "userId": null}	0
9 And method post	267
10 Then status 201	0
11 And match response == {"title": "", "userId": null,"body": "", "id": #ignore}	1

6

0

Scenarios

2023-03-29 01:30:13 p. m.

[1.1:14] update a post

[1.2:15] update a post

[1.3:16] update a post

[2.1:25] invalid id

[2.2:26] invalid id

[2.3:27] invalid id

Summary | Tags | Feature: **src/test/java/features/putpost.feature** | **update post** *AS a json place holder user I WANT TO make a request to update a post SO THAT I CAN update a new post*

Scenario: [1.1:14] **update a post**	ms: 965
7 Given url 'https://jsonplaceholder.typicode.com/posts/1'	66
8 When request {"title":"editing a post", "body": "hello, this an updated post", "userId": 1, "id":1}	34
9 And method put	827
10 Then status 200	0
11 And match response == {"title": "editing a post", "userId": 1,"body": "hello, this an updated post", "id":1}	38

Scenario: [1.2:15] **update a post**	ms: 282
7 Given url 'https://jsonplaceholder.typicode.com/posts/2'	1
8 When request {"title":"updated post", "body": "updating this post", "userId": 3, "id":2}	0
9 And method put	281
10 Then status 200	0
11 And match response == {"title": "updated post", "userId": 3,"body": "updating this post", "id":2}	1

Scenario: [1.3:16] **update a post**	ms: 266
7 Given url 'https://jsonplaceholder.typicode.com/posts/3'	0
8 When request {"title":"still karate", "body": "I'm still learning karate", "userId": 2, "id":30}	0
9 And method put	265
10 Then status 200	0
11 And match response == {"title": "still karate", "userId": 2,"body": "I'm still learning karate", "id":3}	0

Scenario: [2.1:25] **invalid id**	ms: 272
19 Given url 'https://jsonplaceholder.typicode.com/posts/200'	1
20 When request {"title":"editing a post", "body": "", "userId": 500}	0
21 And method put	272
22 Then status 500	0

Scenario: [2.2:26] **invalid id**	ms: 291
19 Given url 'https://jsonplaceholder.typicode.com/posts/250'	1
20 When request {"title":"", "body": "updating this post", "userId": 400}	0
21 And method put	290
22 Then status 500	0

Scenario: [2.3:27] **invalid id**	ms: 249
19 Given url 'https://jsonplaceholder.typicode.com/posts/'	1
20 When request {"title":"still karate", "body": "I'm still learning karate", "userId": 1000}	0
21 And method put	248
22 Then status 404	0