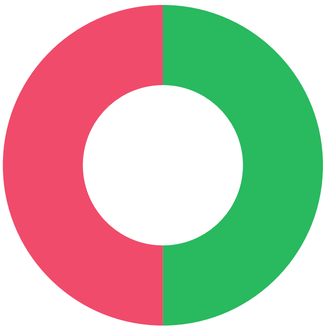
Test for deleting a resource

Test for deleting a resource

Completion chart



Completion stats

- ● Untested (0)
- ● Passed (1)
- ● Failed (1)
- ● Blocked (0)
- ● Retest (0)
- ● Skipped (0)
- ● Invalid (0)

Completion rate

100%

Started by	Start time	Estimated	Time spent
Ni Wayan Amanda Putri Astawa	2024-03-24 06:31:35	00:00:00	00:00:33

Environment	Milestone
Development	Release 1.0.0

RA-11 / Deleting a resource

Delete a specific post

Status	Time spent	Assignee
Passed	00:00:11	Ni Wayan Amanda Putri Astawa

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:11	Ni Wayan Amanda Putri Astawa	-

Finish time

2024-03-24 06:31:49

Steps

Step	1
Action	Click "Add request" in the Collection that has been created
Expected result	One new request added
Status	Untested

Step	2
Action	Add a DELETE request and enter the base URL
Input data	https://jsonplaceholder.typicode.com/posts/1
Expected result	Method becomes DELETE and base URL is added
Status	Untested

Step	3
Action	Click the save button
Expected result	Changes are saved
Status	Untested

Step	4
Action	Click the send button to send the DELETE request to the endpoint
Expected result	JSON response displays data post data based on ID has been deleted
Status	Untested

Step	5
Action	Enter the testing script in the "test" column

Input data

```
pm.test("Response status code is 200", function () {
  pm.response.to.have.status(200);
});



pm.test("Response has the correct Content-Type", function () {
  pm.expect(pm.response.headers.get("Content-Type")).to.include("application/json");
});



pm.test("Response body is empty", function () {
    const responseData = pm.response.json();

    pm.expect(responseData).to.be.an('object').that.is.empty;
});



pm.test("Verify that the post with ID 999 is deleted", function () {
    pm.expect(pm.response.code).to.equal(200);
    pm.expect(pm.response.json()).to.be.an('object').that.is.empty;
});



pm.test("Response time is within an acceptable range", function () {
  pm.expect(pm.response.responseTime).to.be.below(300);
});
```

Expected result	Script for testing added
Status	Untested

Step	6
Action	Click the save button
Expected result	Changes are saved
Status	Untested

Step	7
Action	Click the send button to view the test results and receive a response from the server
Expected result	Testing results are displayed
Status	Untested

RA-12 / Deleting a resource

Delete a non-existing post

Status	Time spent	Assignee
Failed	00:00:22	Ni Wayan Amanda Putri Astawa

Results

Result 1

Status	Time spent	User	Defects
Failed	00:00:22	Ni Wayan Amanda Putri Astawa	D-3 (Open)

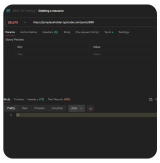
Finish time

2024-03-24 06:34:15

Comment

JSON response displays 200 (OK)

Attachments



DELETE https://jsonplaceholder.typicode.com/posts/999

```
1 {}
```

failed_testing_del...

Steps

Step	1
Action	Click "Add request" in the Collection that has been created
Expected result	One new request added
Status	Untested

Step	2
Action	Add a DELETE request and enter the base URL with an ID that does not exist. (Example: ID = 999)
Input data	https://jsonplaceholder.typicode.com/posts/999
Expected result	Method becomes DELETE and base URL is added
Status	Untested

Step	3
Action	Click the save button
Expected result	Changes are saved
Status	Untested

Step	4

Action	Click the send button to send the DELETE request to the endpoint

Input data	pm.test("Response status code is 200", function () {
	  pm.response.to.have.status(200);
	});
	
	pm.test("Response has the correct Content-Type", function () {
	  pm.expect(pm.response.headers.get("Content-Type")).to.include("application/json");
	});
	
	pm.test("Response body is empty", function () {
	    const responseData = pm.response.json();
	
	    pm.expect(responseData).to.be.an('object').that.is.empty;
	});
	
	pm.test("Verify that the post with ID 999 is deleted", function () {
	    pm.expect(pm.response.code).to.equal(200);
	    pm.expect(pm.response.json()).to.be.an('object').that.is.empty;
	});
	
	pm.test("Response time is within an acceptable range", function () {
	  pm.expect(pm.response.responseTime).to.be.below(300);
	});

Expected result	JSON response displays data post data based on ID has been deleted
Status	Failed

Step	5
Action	Enter the testing script in the "test" column
Expected result	Script for testing added
Status	Untested

Step	6
Action	Click the save button
Expected result	Changes are saved
Status	Untested

Step	7
Action	Click the send button to view the test results and receive a response from the server

Expected result	Testing results are displayed
Status	Untested