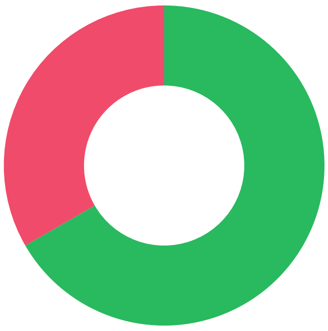
Test for updating a resource

Test for updating a resource

Completion chart	Completion stats	Completion rate
	● Untested (0) ● Passed (2) ● Failed (1) ● Blocked (0) ● Retest (0) ● Skipped (0) ● Invalid (0)	100%

Started by	Start time	Estimated	Time spent
Ni Wayan Amanda Putri Astawa	2024-03-24 06:26:43	00:00:00	00:01:25

Environment	Milestone
Development	Release 1.0.0

Update 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 06:27:01

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 PUT request and enter the BASE URL
Input data	https://jsonplaceholder.typicode.com/posts/1
Expected result	Method becomes PUT and base URL is added
Status	Untested

Step	3
Action	Enter JSON data in the "body" field
Input data	{ "id": 1, "title": "foo", "body": "bar", "userId": 1 }
Expected result	Request body added
Status	Untested

Step	4
Action	Click the save button
Expected result	Changes are saved
Status	Untested

Step	5
Action	Click the send button to send the PUT request to the endpoint
Expected result	JSON response displays the results of changes to post data based on ID
Status	Untested

Step	6
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	7
Action	Click the save button
Expected result	Changes are saved
Status	Untested

Step	8

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-9 / Updating a resource

Update post without inserting JSON data

Status	Time spent	Assignee
Failed	00:01:09	Ni Wayan Amanda Putri Astawa

Results

Result 1

Status	Time spent	User	Defects
Failed	00:01:09	Ni Wayan Amanda Putri Astawa	D-2 (Open)

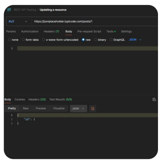
Finish time

2024-03-24 06:29:28

Comment

JSON response displays 200 (OK)

Attachments



REST API Testing / Updating a resource

PUT https://jsonplaceholder.typicode.com/posts/1

```
1
```

Body — Pretty — JSON — Test Results (5/5)

```
{
    "id": 1
}
```

failed_testing_upd..

Steps

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

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

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

Step	4

Action	Click the send button to send the PUT request to the endpoint
Expected result	JSON response displays 400 (Bad Request) or 422 (Unprocessable Entity)
Status	Failed

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

Update a non-existing post

Status	Time spent	Assignee
Passed	00:00:05	Ni Wayan Amanda Putri Astawa

Results

Result 1

Status	Time spent	User	Defects
Passed	00:00:05	Ni Wayan Amanda Putri Astawa	-

Finish time

2024-03-24 06:30:19

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 PUT request and enter its BASE URL with an ID that does not exist. (Example: ID = 999)
Input data	https://jsonplaceholder.typicode.com/posts/999
Expected result	Method becomes PUT and base URL is added
Status	Untested

Step	3
Action	Enter JSON data in the "body" field
Input data	{ "id": 1, "title": "foo", "body": "bar", "userId": 1 }
Expected result	Request body added
Status	Untested

Step	4
Action	Click the save button
Expected result	Changes are saved

Status	Untested

Step	5
Action	Click the send button to send the PUT request to the endpoint
Expected result	JSON response displays code status is 500 (Internal Server Error)
Status	Untested

Step	6
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	7
Action	Click the save button
Expected result	Changes are saved
Status	Untested

Step	8
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