GET 01 - Mendapatkan semua data post dengan status code 200

Memastikan bahwa permintaan mendapatkan semua data post dengan method GET berhasil dan mengembalikan status code 200

Severity **Priority Behavior** Type O Normal ↑ High Positive Integration Layer Is Flaky Milestone **Automation** API No Manual **Status**

Steps to reproduce

Actual

Action

Input data

Klik Send

Step	1
Action	Masukan base url
Input data	https://jsonplaceholder.typicode.com/posts
Expected resul	t
Step	2
Action	Pilih mehod 'GET'
Input data	
Expected resul	t
Step	3
Action	Pada bagian kolom menu pilih 'Test'
Input data	
Expected resul	t
Step	4
Action	Masukan code
Input data	pm.test('Response status code should have 200 value', () =>{ pm.response.to.have.status(200); });
Expected resul	t
Step	5

GET02 - Respons memiliki minimal satu data post

Severity	Priority	Behavior	Туре
O Normal	↑ High	Positive	Integration
Layer	Is Flaky	Milestone	Automation
API	No	Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Step	1
Action	Same as GET01
Input data	
Expected result	
Step	2
Action	Masukan code
Input data	pm.test('response should contain posts data', () => { const responseJson = pm.response.json(); pm.expect(re sponseJson).to.be.an('array').that.is.not.empty; });
Expected result	
Step	3
Action	Klik Send
Input data	
Expected result	

GET03 - Setiap data post memiliki atribut id dan title Memastikan bahwa semua data post memiliki atribut id dan title

Severity	Priority	Behavior	Туре
O Normal	↑ High	Positive	Integration
Layer	Is Flaky	Milestone	Automation
E2E	No Testing RESTFUL API TESTING 1.0	Manual	
		1.0	Status
			Actual

Step	1			
Action	Step Before same as GET01			
Input data				
Expecte result	d			
Step	2			
Action	Masukan code			
Input data	pm.test('each post should have id and title attributes', () => { const responseJson = pm.response.json(); const posts = responseJson; posts.forEach(post => { pm.expect(post).to.have.property('id'); pm.expect(post).to.have.property('title'); }); });			
Expecte result	d			
Step	3			
Action	Klik send			
Input data				
Expecte result	d			

GET04 - Respons berisi data kosong (null value)

Memeriksa apakah respons mengandung data kosong atau datanya kosong

Severity	Priority	Behavior	Туре
O Normal	O Medium	Negative	Integration
Layer	Is Flaky	Milestone	Automation
API	No	Testing RESTFUL API TESTING 1.0	Manual
		1.0	Status
			Actual

Step	1
Action	Step before same as GET01
Input data	
Expected result	
Step	2
Action	Masukan code
Input data	<pre>pm.test('response should be empty', () => { const responseJson = pm.response.json(); pm.expect(responseJson).to.be.an('array').that.is.empty; });</pre>
Expected result	
Step	3
Action	Klik send
Input data	
Expected result	

GET05 - Memastikan respons berformat salah Memeriksa kemungkinan format salah pada respons, misal: pdf

Severity	Priority	Behavior	Туре
O Normal	↑ High	Not set	Integration
Layer	Is Flaky	Milestone	Automation
Not set	No	Testing RESTFUL API TESTING 1.0	Manual
		1.0	Status
			Actual

Step	1
Action	Step before same as GET01
Input data	
Expected result	d
Step	2
Action	Masukan code
Input data	pm.test('response should be in PDF format', () => { const contentType = pm.response.headers.get('Content-Type'); pm.expect(contentType).to.include('application/pdf'); pm.expect(pm.response.text()).to.include('%PDF-'); });
Expected result	d
Step	3
Action	Klik send
Input data	
Expected result	d

POST01 - Membuat data post baru dengan status code 201

Memastikan bahwa permintaan POST berhasil membuat data post baru

SeverityPriorityBehaviorTypeO Normal↑ HighPositiveIntegrationLayerIs FlakyMilestoneAutomationE2ENo-Manual

Status

Actual

Steps to reproduce

5

Step

Step	1
Action	Input base url
Input data	https://jsonplaceholder.typicode.com/posts
Expected result	
Step	2
Action	Pilih method 'POST'
Input data	
Expected result	
Step	3
Action	Pada bagian kolom menu pilih, body dan inputkan request
Input data	{ "title": "foo", "body": "bar", "userId": 1 }
Expected result	
Step	4
Action	Pada bagian kolom menu pilih 'Test'
Input data	
Expected result	

Action	Masukan code
Input data	pm.test('creating a new post should be successful', () => { const responseBody = pm.response.json(); pm.ex pect(pm.response.code).to.equal(201); });
Expected result	
Step	6
Action	Klik Send
Input data	
Expected result	

POST02 - Memeriksa post baru memiliki attribut valid

Memastikan bahwa post baru mengandung key atau properti yang valid

Severity	Priority	Behavior	Туре
O Normal	O Medium	Positive	Integration
Layer	Is Flaky	Milestone	Automation
Not set	No	Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Step	1
Action	Step before same as POST01
Input data	
Expected result	
Step	2
Action	Masukan code
Input data	pm.test('response should contain expected properties', () => { const responseBody = pm.response.json(); pm. expect(responseBody).to.have.all.keys('id', 'title', 'body', 'userld'); });
Expected result	
Step	3
Action	Klik send
Input data	
Expected result	

POST03 - Memastikan bahwa memiliki response 400 (Bad Request) atau 422 (Unprocessable Entity)

Memastikan bahwa POST memiliki response 400 dan 422

Severity	Priority	Behavior	Туре
O Normal	↑ High	Negative	Integration
Layer	Is Flaky	Milestone	Automation
API	No	Testing RESTFUL API TESTING 1.0	Manual
		1.0	Status
			Actual

Step	1
Action	Step before same as POST01
Input data	
Expected result	
Step	2
Action	Masukan code
Input data	pm.test('incomplete request body should fail', () => { pm.expect(pm.response.code).to.be.oneOf([400, 4 22]); });
Expected result	
Step	3
Action	Klik send
Input data	
Expected result	

POST04 - Mengirim dengan parameter tambahan yang tidak dikenali

mengirimkan permintaan POST dengan menambahkan parameter tambahan yang tidak dikenali seperti "extraParameter"

Severity	Priority	Behavior	Туре
O Normal	O Medium	Not set	Integration
Layer	Is Flaky	Milestone	Automation
API	No	Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Step	1
Action	Step before same as POST01
Input data	
Expecte result	d
Step	2
Action	Masukan code
Input data	pm.test('unrecognized parameters should fail', () => { const requestBody = { "title": "foo", "body": "bar", "userId": 1, "extraParameter": "value" }; pm.expect(pm.response.code).to.be.oneOf([400, 422]); });
Expecte result	d
Step	3
Action	Klik send
Input data	
Expecte result	d

POST05 - Memeriksa atribut "id" tidak ada dalam respon

Mengirim POST dengan asumsi attribut "id" tidak ada dalam response

Severity	Priority	Behavior	Туре
^ Major	↑ High	Negative	Integration
Layer	ls Flaky	Milestone	Automation
Not set	No	-	Manual
Status			
Actual			

Step	1
Action	Step before same as POST01
Input data	
Expected result	
Step	2
Action	Masukan code
Input data	<pre>pm.test('response should not have "id" attribute', () => { const responseBody = pm.response.json(); pm.exp ect(responseBody).to.not.have.property('id'); });</pre>
Expected result	
Step	3
Action	Klik send
Input data	
Expected result	

GET(2)01 - Memastikan respons berhasilMendapatkan post data dengan method GET dan data valid dan dengan status code 200

Severity	Priority	Behavior	Туре
O Normal	Not set	Positive	Integration
Layer	Is Flaky	Milestone	Automation
API	No	Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Step	1
Action	Masukan base url
Input data	https://jsonplaceholder.typicode.com/posts
Expected result	
Step	2
Action	Pilih method 'GET'
Input data	
Expected result	
Step	3
Action	Pada bagian kolom menu pilih 'Test'
Input data	
Expected result	
Step	4
Action	Masukan code
Input data	pm.test('Get Post by ID - Successful', () => { pm.response.to.have.status(200); });
Expected result	
Step	5
Action	Klik send
Input data	

GET(2)03 - Respon end-point adalah 404Mengecek apakah respons dari endpoint memiliki status code 404

Severity	Priority	Behavior	Туре
O Normal	Not set	Negative	Integration
Layer	Is Flaky	Milestone	Automation
Not set	No	Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Parameters

ID: 1

Step	1
Action	same as GET(2)01
Input data	
Expected result	

GET(2)02 - Respons mengandung data post dengan ID validMemastikan respons mengandung data post dan ID yang diharapkan

Severity	Priority	Behavior	Туре
O Normal	— Not set	Positive	Integration
Layer	Is Flaky	Milestone	Automation
API		Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Parameters

ID: 1

Step	1
Action	Step before same as GET(2)01
Input data	
Expected result	
Step	2
эсер	
Action	Masukan code
Input data	pm.test('Get Post by ID - Successful', () => { pm.response.to.have.status(200); });
Expected result	
Step	3
Action	Klik send
Input data	
Expected result	

GET(2)04 - Respon mengandung data yang sesuai dengan aslinya

Memastikan bahwa respons dari endpoint harus sama persis dengan data yang diharapkan

Severity	Priority	Behavior	Туре
O Normal	Not set	Positive	Integration
Layer	ls Flaky	Milestone	Automation
API	No	Testing RESTFUL API TESTING 1.0	Manual
			Status
			Actual

Parameters

ID: 1

Step	1
Action	Step before same as GET(2)01
Input data	
Expecte result	d
Step	2
Action	Masukan code
Input data	pm.test('Check Response Data', () => { const expectedData = { "userId": 1, "id": 1, "title": "sunt aut facere rep ellat provident occaecati excepturi optio reprehenderit", "body": "quia et suscipit suscipit recusandae consequu ntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem eveniet arc hitecto" }; const responseData = pm.response.json(); pm.expect(responseData).to.eql(expectedData); });
Expecte result	d
Step	3
Action	Klik send
Input data	
Expecte result	d

GET(2)05 - Mengecek respons dengan data yang unvalid

Memastikan respons dari endpoint jika berbeda dengan data yang valid

Severity	Priority	Behavior	Туре
O Normal	Not set	Negative	Integration
Layer	Is Flaky	Milestone	Automation
Not set	No	Testing RESTFUL API TESTING 1.0	Manual
			Status
			Actual

Step	1
Action	Step before same as GET(2)01
Input data	
Expected result	d
Step	2
Action	Masukan code
Input data	pm.test('Check Response Data', () => { const expectedData = { "userId": 100, "id": 100, "title": "sunt aut facer e repellat provident occaecati excepturi optio reprehenderit", "body": "quia et suscipit suscipit recusandae cons equuntur expedita et cum reprehenderit molestiae ut ut quas totam nostrum rerum est autem sunt rem evenie t architecto" }; const responseData = pm.response.json(); pm.expect(responseData).to.eql(expectedData); });
Expected result	d
Step	3
Action	Klik send
Input data	
Expected result	d

PUT01 - Memastikan update data post berdasarkan ID berhasil Memastikan response PUT untuk mengubah data post berhasil dengan status code 200

Severity	Priority	Behavior	Туре
O Normal	— Not set	Positive	Integration
Layer	Is Flaky	Milestone	Automation
Not set	No	Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Step	1
Action	Masukan base url
Input data	https://jsonplaceholder.typicode.com/posts
Expected result	
Step	2
Action	Pilih method 'PUT'
Input data	
Expected result	
Step	3
Action	Pada bagian kolom menu pilih 'body'
Input data	
Expected result	
Step	4
Action	Isikan body
Input data	{ id: 1, title: 'foo', body: 'bar', userld: 1, }
Expected result	
Class	_
Step	5
Action	Pada bagian kolom menu pilih 'Test'
Input data	

Expected result

Step	6
Action	Masukan code
Input data	pm.test('Update Post - Successful', () => { pm.response.to.have.status(200); });
Expected result	
Step	7
Action	Klik send
Input data	
Expected result	

PUT02 - Memastikan update data post berhasil sesuai data valid Memastikan bahwa data post telah berubah sesuai dengan payload yang diubah setelah melakukan method PUT.

Severity	Priority	Behavior	Туре
O Normal	Not set	Positive	Integration
Layer	Is Flaky	Milestone	Automation
Not set	No	Testing RESTFUL API TESTING 1.0	Manual
			Status
			Actual

Step	1
Action	Step before same as PUT01
Input data	
Expecte result	d
Step	2
Action	Masukan code
Input data	<pre>pm.test('Check Updated Post Data', () => { const updatedData = { "id": 1, "title": "foo", "body": "bar", "userId": 1 }; const responseData = pm.response.json(); pm.expect(responseData).to.eql(updatedData); });</pre>
Expecte result	d
Step	3
Action	Klik send
Input data	
Expecte result	d

PUT03 - Update data post dengan ID unvalid

Memastikan bahwa data post fail dengan ID yang invalid pada body

SeverityPriorityBehaviorTypeO Normal- Not setNegativeIntegrationLayerIs FlakyMilestoneAutomationNot setNo-Manual

Status

Actual

Step	1
Action	Step before same as PUT01
Input data	
Expecte result	d
Step	2
Action	Masukan code
Input data	<pre>pm.test('Check Updated Post Data', () => { const updatedData = { "id": 999, "title": "foo", "body": "bar", "userl d": 1 }; const responseData = pm.response.json(); pm.expect(responseData).to.eql(updatedData); });</pre>
Expecte result	d
Step	3
Action	Klik send
Input data	
Expecte result	d

PUT04 - Memastikan update data post gagal dengan status 400 Memastikan response PUT untuk mengubah data post gagal dengan staus 400 (Bad Request)

	Severity	Priority	Behavior	Туре
	O Normal	— Not set	Negative	Integration
	Layer	Is Flaky	Milestone	Automation
N	Not set	No	Testing RESTFUL API TESTING 1.0	Manual
				Status
				Actual

Step	1
Action	Step before same as PUT01
Input data	
Expected resu	lt
Step	2
Action	Masukan code
Input data	pm.test('Update Post with Invalid Data - Bad Request', () => { pm.response.to.have.status(400); });
Expected resul	lt
Step	3
Action	Klik send
Input data	
Expected resul	lt

PUT05 - Update data post berhasil tanpa parameter ID dan userID

Memastikan test fail dengan data post ID dan userID tidak ada

Severity	Priority	Behavior	Туре
O Normal	Not set	Negative	Integration
Layer	Is Flaky	Milestone	Automation
Not set	No	Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Step	1
Action	Step before same as PUT01
Input data	
Expecte result	d
Step	2
Action	Masukan code
Input data	<pre>pm.test('Check Updated Post Data', () => { const updatedData = { "title": "foo", "body": "bar", }; const respons eData = pm.response.json(); pm.expect(responseData).to.eql(updatedData); });</pre>
Expecte result	d
Step	3
Action	Klik send
Input data	
Expecte result	d

DEL01 - Delete data post berdasarkan ID

Memastikan response dengan method DELETE untuk menghapus data post berdasarkan ID berhasil dengan status code 200

Severity	Priority	Behavior	Туре
O Normal	Not set	Positive	Integration
Layer	Is Flaky	Milestone	Automation
Not set	No	Testing RESTFUL API TESTING	Manual
		1.0	Status
			Actual

Parameters

ID: 1

Step	1
Action	Masukan base url
Input data	https://jsonplaceholder.typicode.com/posts
Expected result	
	_
Step	2
Action	Pilih method 'DELETE'
Input data	
Expected result	
Step	3
Action	Pada bagian kolom menu pilih 'TEST'
Input data	pm.test('Delete Post by ID - Success', () => { pm.response.to.have.status(200); });
Expected result	
Step	4
Action	Klik send
Input data	

DELO2 - Memastikan data post yang dihapus tidak tersedia lagi Memastikan data post tidak tersedia lagi dengan response 404 Not Found

Severity	Priority	Behavior	Туре
O Normal	— Not set	Positive	Integration
Layer	Is Flaky	Milestone	Automation
E2E	No	Testing RESTFUL API TESTING 1.0	Manual
		1.0	Status
			Actual

Step	1
Action	Hapus angka pada url atau param
Input data	
Expected result	
Step	2
Action	Masukan code pada test
Input data	pm.test('Check Deleted Post Availability', () => { pm.response.to.have.status(404); })
Expected result	
Step	3
Action	Klik send
Input data	
Expected result	

DELO3 - Memastikan response 500Memastian test dengan berhasil dengan response 500 atau internal server error

Severity	Priority	Behavior	Туре
O Normal	— Not set	Negative	Other
Layer	Is Flaky	Milestone	Automation
E2E	No	Testing RESTFUL API TESTING 1.0	Manual
		1.0	Status
			Actual

Step	1
Action	Step before same as PUT01
Input data	
Expected result	
Step	2
Action	Masukan code
Input data	pm.test('Delete Post by ID - Success', () => { pm.response.to.have.status(500); });
Expected result	
Step	3
Step	3 Klik send