

API “reqres” project

API Testing for <https://reqres.in/> – DEPI Graduation Project

As part of the DEPI Graduation Project, API requests were tested using the <https://reqres.in/> mock API service. The following test cases were considered to ensure the reliability and performance of the endpoints:

1. **Status Code Validation** – Verifying that each API request returns the expected HTTP status code (e.g., 200 OK, 201 Created, 400 Bad Request, etc.).
2. **Response Time** – Measuring and ensuring that the API responds within an acceptable time threshold to maintain performance standards.
3. **Action Verification** – Confirming that the intended action (e.g., user creation, update, deletion) was successfully executed as per the request.
4. **Response Body Validation** – Inspecting the structure and content of the response body to ensure it matches the expected schema and data.

Website API Requests

GET	LIST USERS
GET	SINGLE USER
GET	SINGLE USER NOT FOUND
GET	LIST <RESOURCE>
GET	SINGLE <RESOURCE>
GET	SINGLE <RESOURCE> NOT FOUND
POST	CREATE
PUT	UPDATE
PATCH	UPDATE
DELETE	DELETE
POST	REGISTER - SUCCESSFUL
POST	REGISTER - UNSUCCESSFUL
POST	LOGIN - SUCCESSFUL
POST	LOGIN - UNSUCCESSFUL
GET	DELAYED RESPONSE

Request

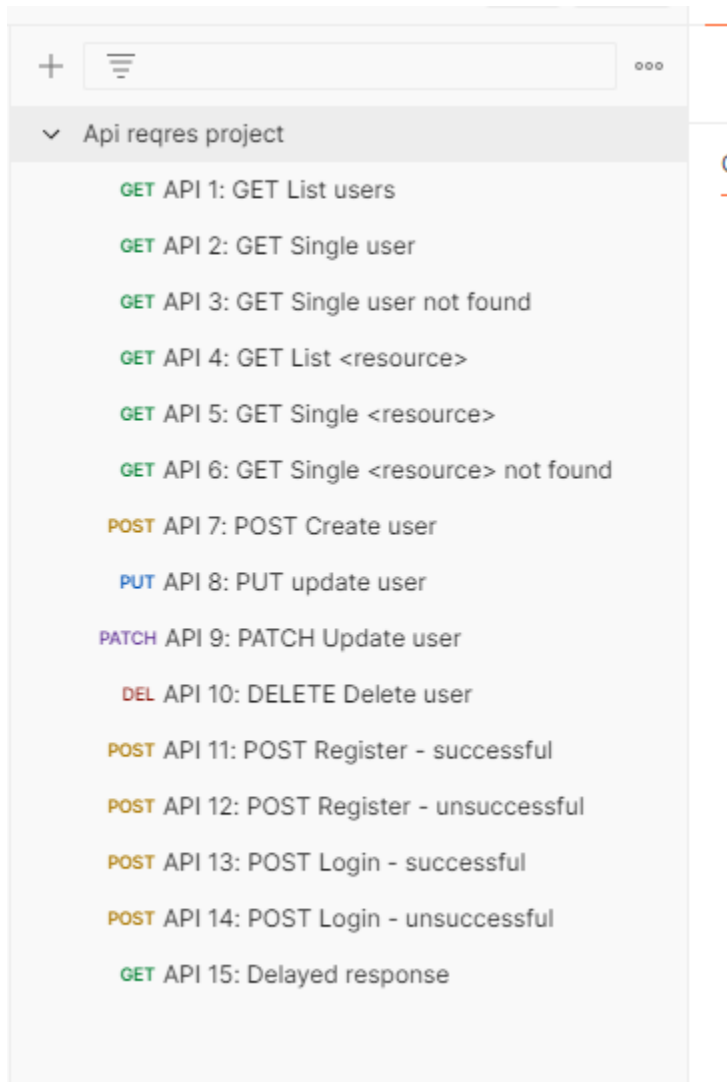
/api/users?page=2

Response

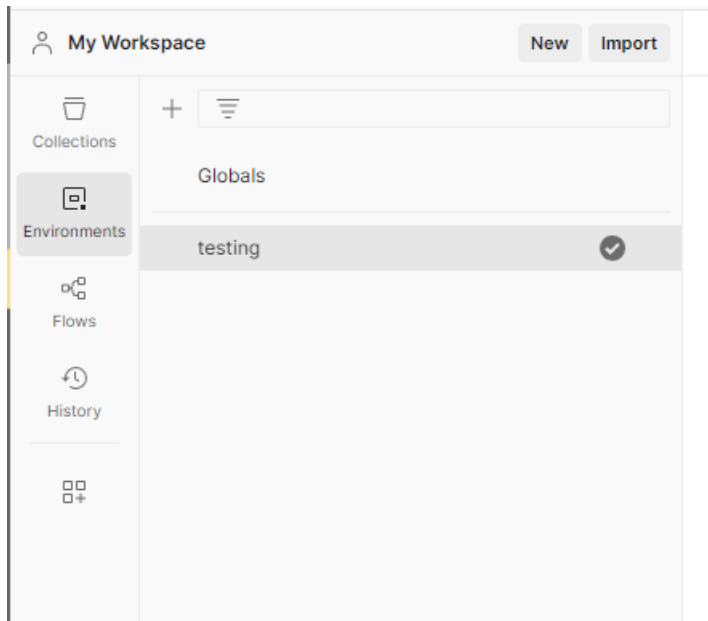
200

```
{
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://reqres.in/img"
    },
    {
      "id": 8,
      "email": "lindsay.ferguson@reqres.in",
      "first_name": "Lindsay",
      "last_name": "Ferguson",
      "avatar": "https://reqres.in/img"
    },
    {
      "id": 9,
      "email": "tobias.funke@reqres.in",
      "first_name": "Tobias",
      "last_name": "Funke",
      "avatar": "https://reqres.in/img"
    },
    {
      "id": 10,
      "email": "byron.fields@reqres.in",
      "first_name": "Byron",
      "last_name": "Fields",
      "avatar": "https://reqres.in/img"
    },
    {
      "id": 11,
      "email": "george.edwards@reqres.in",
      "first_name": "George",
      "last_name": "Edwards",
      "avatar": "https://reqres.in/img"
    },
    {
      "id": 12,
      "email": "rachel.howell@reqres.in",
      "first_name": "Rachel",
      "last_name": "Howell",
      "avatar": "https://reqres.in/img"
    }
  ],
  "support": {
    "url": "https://contentcaddy.io?utm_source=reqres",
    "text": "Tired of writing endless sc"
  }
}
```

Postman Collection



Postman Environment and its variables



testing Save

Filter variables

	Variable	Type		Initial value	Current value
<input checked="" type="checkbox"/>	baseUrl	default	▼	199595959	199595959
<input checked="" type="checkbox"/>	name	default	▼	omer	omer
<input checked="" type="checkbox"/>	job	default	▼	engineer	engineer
<input checked="" type="checkbox"/>	job2	default	▼	tester	tester
<input checked="" type="checkbox"/>	job3	default	▼	artist	artist
	Add new variable				

Run Collection

Run Sequence

Deselect All | Select All | Reset

1

☒

GET

API 1: GET List users

2

☒

GET

API 2: GET Single user

3

☒

GET

API 3: GET Single user not found

4

☒

GET

API 4: GET List <resource>

5

☒

GET

API 5: GET Single <resource>

6

☒

GET

API 6: GET Single <resource> not found

7

☒

POST

API 7: POST Create user

8

☒

PUT

API 8: PUT update user

9

☒

PATCH

API 9: PATCH Update user

10

☒

DEL

API 10: DELETE Delete user

11

☒

POST

API 11: POST Register - successful

12

☒

POST

API 12: POST Register - unsuccessful

13

☒

POST

API 13: POST Login - successful

14

☒

POST

API 14: POST Login - unsuccessful

15

☒

GET

API 15: Delayed response

Functional | Performance

Choose how to run your collection

☒ Run manually

Run this collection in the Collection Runner.

☐ Schedule runs

Periodically run collection at a specified time on the Postman Cloud.

☐ Automate runs via CLI

Configure CLI command to run on your build pipeline.

Run configuration

Iterations ⓘ

3

Delay ⓘ

100ms

Test data file ⓘ

Only JSON and CSV files are accepted.

Select File

> Advanced settings

Run Api reqres project

Run Result

Api reqres project - Run results

 Ran today at 05:24:11 · [View all runs](#)

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	testing	3	26s 215ms	159	372 ms

RUN SUMMARY

		1	2	3
▶ GET	API 1: GET List users	12 0	<div></div>	<div></div>
▶ GET	API 2: GET Single user	12 0	<div></div>	<div></div>
▶ GET	API 3: GET Single user not found	9 0	<div></div>	<div></div>
▶ GET	API 4: GET List <resource>	15 0	<div></div>	<div></div>
▶ GET	API 5: GET Single <resource>	9 0	<div></div>	<div></div>
▶ GET	API 6: GET Single <resource> not found	9 0	<div></div>	<div></div>
▶ POST	API 7: POST Create user	9 0	<div></div>	<div></div>
▶ PUT	API 8: PUT update user	9 0	<div></div>	<div></div>
▶ PATCH	API 9: PATCH Update user	9 0	<div></div>	<div></div>
▶ DELETE	API 10: DELETE Delete user	9 0	<div></div>	<div></div>
▶ POST	API 11: POST Register - successful	15 0	<div></div>	<div></div>
▶ POST	API 12: POST Register - unsuccessful	9 0	<div></div>	<div></div>
▶ POST	API 13: POST Login - successful	9 0	<div></div>	<div></div>
▶ POST	API 14: POST Login - unsuccessful	9 0	<div></div>	<div></div>
▶ GET	API 15: Delayed response	15 0	<div></div>	<div></div>

All Tests Passed (159) Failed (0) Skipped (0)


Iteration 1

GET API 1: GET List users

<https://reqres.in/api/users?page=2>

- PASS Status code is 200
- PASS Response time is less than 3000ms
- PASS response body has users list object
- PASS Each user has id, email, first name , last_name

API 1

 Api reqres project / **API 1: GET List users**

GET

▼

https://reqres.in/api/users?page=2

Params ●

Authorization

Headers (6)

Body

Scripts ●

Settings

Pre-request


Post-response ●

```
1  pm.test("Status code is 200", function () {
2    |    pm.response.to.have.status(200);
3  });
4  pm.test("Response time is less than 3000ms", function () {
5    |    pm.expect(pm.response.responseTime).to.be.below(3000);
6  });
7  pm.test("response body has users list object", function () {
8    |    pm.expect(pm.response.text()).to.include("data");
9  });
10 // Assert that each product has the necessary fields (id, name, price)
11 pm.test("Each user has id, email, first name , last_name", function () {
12   |    const responseJSON = pm.response.json();
13   |    pm.expect(responseJSON.data).to.be.an('array');
14   |    responseJSON.data.forEach(data => {
15   |      |    pm.expect(data).to.have.property('id');
16   |      |    pm.expect(data).to.have.property('email');
17   |      |    pm.expect(data).to.have.property('first_name');
18   |      |    pm.expect(data).to.have.property('last_name');
19   |    });
20 });
```

Body

Cookies

Headers (19)

Test Results (4/4) | 

200 OK

Filter Results ▼

PASSED

Status code is 200

PASSED

Response time is less than 3000ms

PASSED

response body has users list object

PASSED

Each user has id, email, first name , last_name

API 2

 Api reqres project / API 2: GET Single user

GET

▼

https://reqres.in/api/users/2

ParamsAuthorizationHeaders (6)BodyScripts ●Settings

Pre-request

Post-response *

```
1  ✓ pm.test("Status code is 200", function () {
2    |   pm.response.to.have.status(200);
3    | });
4  ✓ pm.test("Response time is less than 3000ms", function () {
5    |   pm.expect(pm.response.responseTime).to.be.below(3000);
6    | });
7  ✓ pm.test("response body has data parameter", function () {
8    |   pm.expect(pm.response.text()).to.include("data");
9    | });
10 ✓ pm.test("Only a single user is returned from the API response", function () {
11 |   var responseData = pm.response.json();
12 |   // Check if the 'data' property exists and is an object
13 |   pm.expect(responseData).to.have.property('data').that.is.an('object');
14 |   // Ensure that the 'data' object contains a single user with an 'id' field
15 |   pm.expect(responseData.data).to.have.property('id');
16 |   // Assert that there is no 'page' property in the 'data' object
17 |   pm.expect(responseData.data).to.not.have.property('page');
18 |   // Check if the response contains only one user (e.g., if 'data' is the user object and not an array)
19 |   pm.expect(responseData.data).to.not.be.an('array'); // This checks if 'data' is not an array
20 | });
```

BodyCookiesHeaders (19)Test Results (4/4) | 🕒

200 OK • 306 ms • 1.35 KB •

Filter Results ▼

PASSED

Status code is 200

PASSED

Response time is less than 3000ms


PASSED

response body has data parameter

PASSED

Only a single user is returned from the API response

API 3

 Api reqres project / **API 3: GET Single user not found**

GET

https://reqres.in/api/users/23

Params

Authorization

Headers (6)

Body


Scripts

Settings

Pre-request

Post-response

```
1 pm.test("Status code is 404", function () {
2   |   pm.response.to.have.status(404);
3   | });
4 pm.test("Response time is less than 3000ms", function () {
5   |   pm.expect(pm.response.responseTime).to.be.below(3000);
6   | });
7 pm.test("Single user not found - response is empty", function () {
8   |   var jsonData = pm.response.json();
9   |
10  |   // Check that the response is an empty object
11  |   pm.expect(Object.keys(jsonData).length).to.eql(0);
12  | });
13
```


 P

Body

Cookies

Headers (18)

Test Results (3/3)



404 Not Found • 157 ms • 1.14 KB

Filter Results ▾

PASSED

Status code is 404

PASSED

Response time is less than 3000ms

PASSED

Single user not found - response is empty

API 4

 Api reqres project / API 4: GET List <resource>

GET

▼

https://reqres.in/api/unknown

Params

Authorization

Headers (6)

Body

Scripts ●

Settings

Pre-request

Post-response ▾

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Response time is less than 3000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(3000);
6 });
7 pm.test("Response has pagination fields", function () {
8   const jsonData = pm.response.json();
9   pm.expect(jsonData).to.have.property("page");
10  pm.expect(jsonData).to.have.property("per_page");
11  pm.expect(jsonData).to.have.property("total");
12  pm.expect(jsonData).to.have.property("total_pages");
13 });
14 pm.test("Data array exists and has 6 items", function () {
15   const jsonData = pm.response.json();
16   pm.expect(jsonData).to.have.property("data").that.is.an("array");
17   pm.expect(jsonData.data.length).to.eql(6);
18 });
19 pm.test("Each item in data has expected fields", function () {
20   const jsonData = pm.response.json();
21   jsonData.data.forEach(item => {
22     pm.expect(item).to.have.property("id");
23     pm.expect(item).to.have.property("name");
24     pm.expect(item).to.have.property("year");
25     pm.expect(item).to.have.property("color");
26     pm.expect(item).to.have.property("pantone_value");
```

 Pa

Body

Cookies

Headers (19)

Test Results (5/5) | 🔊

200 OK • 94 ms • 1.5 KB •

Filter Results ▼

PASSED

Status code is 200

API 5

 Api reqres project / API 5: GET Single <resource>

GET

▼

https://reqres.in/api/unknown/2

Params

Authorization

Headers (6)

Body

Scripts ●

Settings

Pre-request

Post-response ●

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Response time is less than 3000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(3000);
6 });
7 pm.test("Response contains a single 'data' object, not an array", function () {
8   const jsonData = pm.response.json();
9
10  // Check that 'data' exists and is a single object (not array)
11  pm.expect(jsonData).to.have.property("data").that.is.an("object");
12  pm.expect(jsonData.data).to.not.be.an("array");
13
14  // Optional: check if it has expected keys
15  pm.expect(jsonData.data).to.include.all.keys("id", "name", "year", "color", "pantone_value");
16 });
```

Body

Cookies

Headers (19)

Test Results (3/3)

🕒

200 OK

93 ms

1.34 KB

⌵

Filter Results ▼

PASSED

Status code is 200

PASSED

Response time is less than 3000ms

PASSED

Response contains a single 'data' object, not an array

API 6

HTTP

Api reqres project / API 6: GET Single <resource> not found

GET

https://reqres.in/api/unknown/23

Params

Authorization

Headers (6)

Body

Scripts ●

Settings

Pre-request

Post-response ●

```
1 pm.test("Status code is 404 - resource not found", function () {
2   | pm.expect(pm.response.code).toEqual(404);
3   | });
4 pm.test("Response time is less than 3000ms", function () {
5   | pm.expect(pm.response.responseTime).toBeBelow(3000);
6   | });
7 pm.test("Response body is an empty object", function () {
8   |   const jsonData = pm.response.json();
9   |
10  |   // Make sure response is an object
11  |   pm.expect(jsonData).toBeAn("object");
12  |
13  |   // And it has no properties (i.e., it's empty)
14  |   pm.expect(Object.keys(jsonData).length).toEqual(0);
15  | });
16
17
```

Body

Cookies

Headers (17)

Test Results (3/3) | ↺

404 Not Found • 172 ms • 1.1 KB

Filter Results ▾

PASSED

Status code is 404 - resource not found

PASSED

Response time is less than 3000ms

PASSED

Response body is an empty object

API 7

HTTP

Api reqres project / API 7: POST Create user

POST

https://reqres.in/api/users

Params

Authorization

Headers (8)

Body

Scripts

Settings

Pre-request

Post-response

```
1 pm.test("Status code is 201", function () {
2   pm.response.to.have.status(201);
3 });
4 pm.test("Response time is less than 3000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(3000);
6 });
7 let nameVar = pm.environment.get("name");
8 let jobVar = pm.environment.get("job");
9 pm.test("user created successfully, Response contains name, job that was sended as parameters", function () {
10   const response = pm.response.json();
11   pm.expect(response).to.have.property('name', nameVar);
12   pm.expect(response).to.have.property('job', jobVar);
13   pm.expect(response).to.have.property('id').that.is.a('string');
14 });
15
```

Body

Cookies

Headers (15)

Test Results (3/3)

201 Created

171 ms

1.13 KB

Filter Results

PASSED

Status code is 201

PASSED

Response time is less than 3000ms

PASSED

user created successfully, Response contains name, job that was sended as parameters

Params

Authorization

Headers (8)

Body

Scripts

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

```
1 {
2   "name": "{{name}}",
3   "job": "{{job}}"
4 }
```

API 8

API reqres project / API 8: PUT update user

PUT

https://reqres.in/api/users/2

Params

Authorization

Headers (8)

Body

Scripts

Settings

Pre-request

Post-response

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Response time is less than 3000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(3000);
6 });
7 let nameVar = pm.environment.get("name");
8 let job2Var = pm.environment.get("job2");
9 pm.test("user updated successfully, Response contains correct name, job ", function () {
10   const response = pm.response.json();
11   pm.expect(response).to.have.property('name', nameVar);
12   pm.expect(response).to.have.property('job', job2Var);
13 });
```

Body

Cookies

Headers (17)

Test Results (3/3)

200 OK • 158 ms • 1.16 KB

Filter Results

PASSED

Status code is 200

PASSED

Response time is less than 3000ms

PASSED

user updated successfully, Response contains correct name, job

Params

Authorization

Headers (8)

Body

Scripts

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

```
1 {
2   "name": "{{name}}",
3   "job": "{{job2}}"
4 }
```

API 9

Api reqres project / API 9: PATCH Update user

PATCH

https://reqres.in/api/users/2

Params

Authorization

Headers (8)

Body

Scripts

Settings

Pre-request

Post-response

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Response time is less than 3000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(3000);
6 });
7 let nameVar = pm.environment.get("name");
8 let job3Var = pm.environment.get("job3");
9 pm.test("user updated successfully, Response contains correct name, job ", function () {
10   const response = pm.response.json();
11   pm.expect(response).to.have.property('name', nameVar);
12   pm.expect(response).to.have.property('job', job3Var);
13 });
```

Body

Cookies

Headers (16)

Test Results (3/3)

200 OK

171 ms

1.14 KB

Filter Results

PASSED

Status code is 200

PASSED

Response time is less than 3000ms

PASSED

user updated successfully, Response contains correct name, job

Params

Authorization

Headers (8)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "name": "{{name}}",
3   "job": "{{job3}}"
4 }
```


API 10

HTTP

Api reqres project / API 10: DELETE Delete user

DELETE

https://reqres.in/api/users/2

Params

Authorization

Headers (6)

Body

Scripts ●

Settings

Pre-request

Post-response ▾

1

2

3

4

5

6

7

8

9

10

11

12

pm.test("Status code is 204", function () {

pm.response.to.have.status(204);

});

pm.test("Response time is less than 3000ms", function () {

pm.expect(pm.response.responseTime).to.be.below(3000);

});

pm.test("Response body is empty", function () {

pm.response.to.have.body("");

});

⏏

Pac

Body

Cookies

Headers (14)

Test Results (3/3) | ⌵

204 No Content

147 ms

1 KB


Filter Results ▾

PASSED Status code is 204

PASSED Response time is less than 3000ms

PASSED Response body is empty

API 11

 Api reqres project / API 11: POST Register - successful

POST

https://reqres.in/api/register

Params

Authorization

Headers (8)

Body

Scripts

Settings

Pre-request

Post-response

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4 pm.test("Response time is less than 3000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(3000);
6 });
7 pm.test("Response contains 'id' and 'token'", function () {
8   const response = pm.response.json();
9   pm.expect(response).to.have.property('id');
10  pm.expect(response).to.have.property('token');
11 });
12 pm.test("The 'id' should be a number", function () {
13   const response = pm.response.json();
14   pm.expect(response.id).to.be.a('number');
15 });
16 pm.test("The 'token' should be a string", function () {
17   const response = pm.response.json();
18   pm.expect(response.token).to.be.a('string');
19 });
20
```

Body

Cookies

Headers (16)

Test Results (5/5)

200 OK

165 ms

1.13 KB

Filter Results

PASSED

Status code is 200

PASSED

Response time is less than 3000ms

PASSED

Response contains 'id' and 'token'

PASSED

The 'id' should be a number

Params

Authorization

Headers (8)

Body

Scripts

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

```
1 {
2   "email": "eve.holt@reqres.in",
3   "password": "pistol"
4 }
```

API 12

Api reqres project / API 12: POST Register - unsuccessful

POSThttps://reqres.in/api/register

ParamsAuthorizationHeaders (8)Body●Scripts●Settings

Pre-request

Post-response*

```
1 pm.test("Status code is 400 for missing password field", function () {
2   pm.response.to.have.status(400);
3 });
4 pm.test("Response time is less than 3000ms", function () {
5   pm.expect(pm.response.responseTime).to.be.below(3000);
6 });
7
8 pm.test("Error message indicates missing password", function () {
9   const response = pm.response.json();
10  pm.expect(response.error).to.include('Missing password');
11 });
12
```

Pe

BodyCookiesHeaders (15)Test Results (3/3)🕒

400 Bad Request • 179 ms • 1.08 KB •

Filter Results ▾

PASSED

Status code is 400 for missing password field

PASSED

Response time is less than 3000ms

PASSED

Error message indicates missing password

☐ none☐ form-data☐ x-www-form-urlencoded☒ raw☐ binary☐ GraphQLJSON ▾

1

2

3

```
"email": "sydney@fife"
```

API 13

HTTP

Api reqres project / API 13: POST Login - successful

POST

https://reqres.in/api/login

Params

Authorization

Headers (8)

Body

Scripts

Settings

Pre-request

Post-response

```
1 pm.test("Status code is 200 for successful login", function () {
2   |   pm.response.to.have.status(200);
3   | });
4
5 pm.test("Response time is less than 3000ms", function () {
6   |   pm.expect(pm.response.responseTime).to.be.below(3000);
7   | });
8
9 pm.test("Response contains 'token'", function () {
10  |   const response = pm.response.json();
11  |   pm.expect(response).to.have.property('token');
12  | });
```

Body

Cookies

Headers (15)

Test Results (3/3)

200 OK

151 ms

1.07 KB

Filter Results

PASSED

Status code is 200 for successful login

PASSED

Response time is less than 3000ms

PASSED

Response contains 'token'

Params

Authorization

Headers (8)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1 {
2   "email": "eve.holt@reqres.in",
3   "password": "cityslicka"
4 }
```

API 14

HTTP

Api reqres project / API 14: POST Login - unsuccessful

POST

https://reqres.in/api/login

Params

Authorization

Headers (8)

Body ●

Scripts ●

Settings

Pre-request

Post-response *

```
1 pm.test("Status code is 400 for missing password field", function () {
2   | pm.response.to.have.status(400);
3   | });
4
5 pm.test("Response time is less than 3000ms", function () {
6   | pm.expect(pm.response.responseTime).to.be.below(3000);
7   | });
8
9 pm.test("Error message indicates missing password", function () {
10  |   const response = pm.response.json();
11  |   pm.expect(response.error).to.include('Missing password');
12  | });
13
```

Body

Cookies

Headers (15)

Test Results (3/3)

400 Bad Request

158 ms

1.07 KB

Filter Results ▾

PASSED

Status code is 400 for missing password field

PASSED

Response time is less than 3000ms

PASSED

Error message indicates missing password

Params

Authorization

Headers (8)

Body ●

Scripts ●

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▾

1 ▾

2 "email": "peter@klaven"

3

API 15

[HTTP](#) Api reqres project / API 15: Delayed response

GET ▼ https://reqres.in/api/users?delay=3

Params ● Authorization Headers (6) Body Scripts ● Settings

Pre-request

Post-response ●

```
1 pm.test("Status code is 200", function () {
2   pm.response.to.have.status(200);
3 });
4
5 pm.test("Response contains correct data structure", function () {
6   const response = pm.response.json();
7   pm.expect(response).to.have.property("page");
8   pm.expect(response).to.have.property("per_page");
9   pm.expect(response).to.have.property("total");
10  pm.expect(response).to.have.property("data").that.is.an('array');
11 });
12
13 pm.test("Data contains correct number of users", function () {
14   const response = pm.response.json();
15   pm.expect(response.data.length).to.eql(6);
16 });
17
18 pm.test("Response time is at least 3000ms", function () {
19   pm.expect(pm.response.responseTime).to.be.above(3000);
20 });
21
22 pm.test("Response time is under 5000ms", function () {
23   pm.expect(pm.response.responseTime).to.be.below(5000);
24 });
25
```

Body Cookies Headers (16) Test Results (5/5) ↻ 200 OK • 3.16 s • 1.46 KB • 📄 Pac

Filter Results ▼

PASSED

Status code is 200

PASSED

Response contains correct data structure