

API - Postman

1)GET

API_cv / User addition verification

Save

GET

https://gorest.co.in/public/v2/users/722716

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

Text

1

Body

Cookies

Headers (22)

Test Results (3/3)

200 OK

189 ms

1.09 KB

Save as Example

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

"id": 722716,

"name": "Bala Agarwal",

"email": "agarwal_bala@olson.biz",

"gender": "female",

"status": "active"

API_cv / User addition verification

Save

GET

https://gorest.co.in/public/v2/users/722716

Send

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Cookies

1

2

3

4

5

6

7

8

9

10

pm.test("Status code is 200", function () {

pm.response.to.have.status(200);

});

pm.test("Check name", function () {

pm.expect(pm.response.text()).to.include("Bala Agarwal");

});

pm.test("Check gender", function () {

var jsonData = pm.response.json();

pm.expect(jsonData.gender).to.eql("female");

});

Body

Cookies

Headers (22)

Test Results (3/3)

200 OK

189 ms

1.09 KB

Save as Example

All

Passed

Skipped

Failed

PASS

Status code is 200

PASS

Check name

PASS

Check gender

2)PATCH

API_cv / Verification of updating part of the post

PATCH Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify

```
1 {
2   ... "title": "Zmiana"
3 }
```

Body Cookies Headers (23) Test Results 200 OK 251 ms 1.18 KB Save as Example

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "userId": 1,
3   "id": 3,
4   "title": "Zmiana",
5   "body": "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odio et labore et velit aut"
6 }
```

3)PUT

PUT Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** Beautify



```
1 {
2   "userId": 1,
3   ... "id": 4,
4   ... "title": "1leum et est occaecati33",
5   ... "body": "1lullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit33"
6   ... }
```




Body Cookies Headers (23) Test Results (1/1) 200 OK 248 ms 1.23 KB Save as Example



Pretty Raw Preview Visualize **JSON**



```
1 {
2   "userId": 1,
3   "id": 4,
4   "title": "1leum et est occaecati33",
5   "body": "1lullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit33"
6 }
```



4)DELETE








API_cy / Verification of deleting post  

 Save  




 



DELETE  https://jsonplaceholder.typicode.com/posts/4 

Params Authorization Headers (8) **Body**  Pre-request Script Tests  Settings [Cookies](#)

 none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON**  [Beautify](#)



```
1 {
2   "userId": 1,
3   "id": 4,
4   "title": "eum et est occaecati33",
5   "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum\nculpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum\nillo velit"
```




Body Cookies Headers (22) Test Results (1/1)  200 OK 282 ms 953 B  Save as Example 



Pretty Raw Preview Visualize **JSON**  



```
1 {}
```



5)POST








API_cy / Verification of posting post  

 Save  




 



POST  https://jsonplaceholder.typicode.com/posts 

Params Authorization Headers (8) **Body**  Pre-request Script Tests  Settings [Cookies](#)

 none  form-data  x-www-form-urlencoded  raw  binary  GraphQL **JSON**  [Beautify](#)

```
1 {
2   "title": "{{randomFullName}}",
3   "body": "nowy post"
4 }
```

Body Cookies Headers (24) Test Results (1/1)  201 Created 339 ms 1.1 KB  Save as Example 

Pretty Raw Preview Visualize **JSON**  

```
1 {
2   "title": "Guy Yost",
3   "body": "nowy post",
4   "id": 101
5 }
```

6)GET - variable

The screenshot displays a REST client interface for a GET request. The URL is `https://jsonplaceholder.typicode.com/users/{{id}}`. The **Body** tab is selected, showing a single line with the variable `1`. A dropdown menu is open over the variable, showing its details: **E id**, **INITIAL 1**, **CURRENT 1**, and **SCOPE Environment**. The **Test Results (1/1)** tab is active, showing a **PASS** status with the message "Status code is 200". The response status is **200 OK** with a response time of **19 ms** and a size of **1.47 KB**. The interface includes tabs for **Params**, **Authorization**, **Headers (6)**, **Body**, and **Pre-request**. The **Body** tab has radio buttons for **none**, **form-data**, **x-www-form-urlencoded**, and **raw**. The **Test Results** tab has buttons for **All**, **Passed**, **Skipped**, and **Failed**. The **Test Results** tab also has a **PASS** button and a **Status code is 200** message.

7)POST - variable, tests

API_cv / Verification of posting post

POST http://{{host}}:3000/posts

Params Authorization host

none form-data

1 {
2 "title": "f",
3 "author": "f"

API_cv / Verification of posting post

POST http://{{host}}:3000/posts

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

1 {
2 "title": "{{title}}",
3 "author": "{{author}}"
4 }

Body Cookies Headers (15) Test Results (2/2)

201 Created 37 ms 564 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {  
2   "title": "tytul1",  
3   "author": "autor1",  
4   "id": 52  
5 }
```

API_cv / Verification of posting post

POST http://{{host}}:3000/posts

Params Authorization Headers (8) Body Pre-request Script Tests Settings

```
1 pm.test("Status code is 201", function () {  
2   pm.response.to.have.status(201);  
3 });  
4 pm.environment.get("title");  
5  
6 pm.test("Check title", function () {  
7   var jsonData = pm.response.json();  
8   pm.expect(jsonData.title).to.eql(pm.environment.get("title"));  
9 });
```

Body Cookies Headers (15) Test Results (2/2)

201 Created 37 ms 564 B Save as Example

All Passed Skipped Failed

PASS Status code is 201

PASS Check title

8)Authorization

Auth / basic auth

Save

GET <https://postman-echo.com/basic-auth> Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Type Basic A...
The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

Username postman
Password

Body Cookies (1) Headers (6) Test Results 200 OK 577 ms 338 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {  
2   "authenticated": true  
3 }
```

9)Authorization

Auth / Digest auth

Save

GET <https://postman-echo.com/digest-auth> Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Type Digest...
The authorization header will be automatically generated when you send the request. Learn more about [authorization](#)
By default, Postman will extract

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#)

Username postman
Password

Body Cookies (1) Headers (6) Test Results 200 OK 141 ms 340 B Save as Example

Pretty Raw Preview Visualize JSON

```
1 {  
2   "authenticated": true  
3 }
```