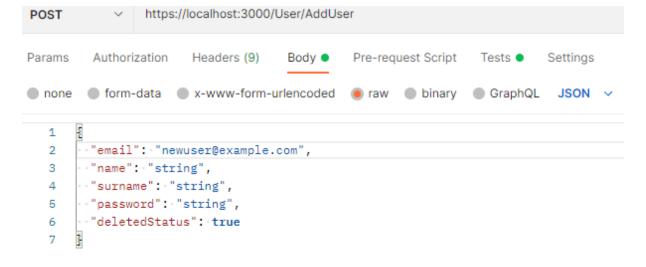
Testovi za Task 4

Task	Test Scenario #	Scenario ID	Test Scenario Description	Test Cases	
	1	S1.1	Testovi za rutu AddUser – ispravni podaci	 Provjera ispravnosti tipa odgovora Provjera postojanja content-type Provjera da li je content-type u JSON formatu Provjera da li response ima body Provjera ispravnog statusnog koda 	
	2	S1.2	estovi za rutu AddUser – eispravni podaci, prazno olje za unos imena	 Provjera ispravnosti tipa odgovora Provjera postojanja content-type Provjera da li je content-type u JSON formatu Provjera da li response ima body Provjera ispravnog statusnog koda 	
	3	S2.1	AddUserAdmin – ispravni podaci	 Provjera ispravnosti tipa odgovora Provjera postojanja content-type Provjera da li je content-type u JSON formatu Provjera da li response ima body Provjera ispravnog statusnog koda 	
	4	S2.2	Testovi za rutu AddUserAdmin – neispravni podaci, prazno polje	 Provjera ispravnosti tipa odgovora Provjera postojanja content-type Provjera da li je content-type u JSON formatu Provjera da li response ima body Provjera ispravnog statusnog koda 	

Testovi za rutu AddUser

1) Ispravni podaci

Body:



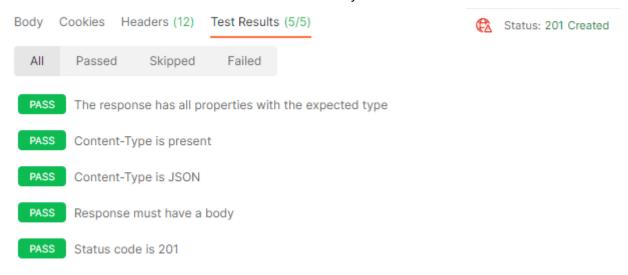
Response Body:

```
Body Cookies Headers (12) Test Results (5/5)
  Pretty
            Raw
                    Preview
                                Visualize
                                             JSON V
    1
            "id": 24,
    2
    3
            "email": "newuser@example.com",
            "name": "string",
    4
    5
            "surname": "string",
            "password": "string",
    7
            "deletedStatus": true
    8
```

Tests:

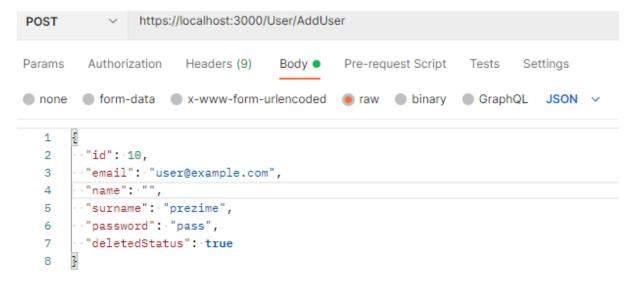
```
const response = pm.response.json();
 2 \times pm.test("The response has all properties with the expected type", () => {
         for(var i = 0; i < response.length; i++){</pre>
             pm.expect(response[i].id).to.be.an("number");
 4
             pm.expect(response[i].email).to.be.a("string");
 5
             pm.expect(response[i].name).to.be.a("string");
 6
 7
             pm.expect(response[i].surname).to.be.an("string");
             pm.expect(response[i].password).to.be.an("object");
 8
 9
             pm.expect(response[i].deletedStatus).to.be.an("boolean");
10
    });
11
12 \( pm.test("Content-Type is present", function () \{
        pm.response.to.have.header("Content-Type");
13
14
    {);
15
16 \rightarrow pm.test("Content-Type is JSON", function () {
17
         pm.response.to.be.json;
    {);
19 \( pm.test("Response must have a body", function () \{
20
          pm.response.to.be.withBody;
21
         pm.response.to.be.json;
22
    3);
23 \rightarrow pm.test("Status code is 201",() => {
        pm.response.to.have.status(201);
   3);
25
```

Rezultati testiranja:



2) Neispravni podaci, prazno polje za unos imena

Body:



Response Body:

```
Body Cookies Headers (11) Test Results
                               Visualize
                                           JSON V
  Pretty
           Raw
                   Preview
   1
            "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
    2
    3
            "title": "One or more validation errors occurred.",
            "status": 400,
   4
            "traceId": "00-598c9c0120701f3aa2e12c473c3dbdd9-4744af26f56fb9a5-00",
    5
            "errors": {
    6
   7
                "Name": [
   8
                    "The Name field is required."
   9
  10
  11
```

Tests:

```
1 const response = pm.response.json();
 2 ∨ pm.test("The response has all properties with the expected type", () => {
        for(var i = 0; i < response.length; i++){</pre>
3 ∨
4
            pm.expect(response[i].type).to.be.an("string");
5
            pm.expect(response[i].title).to.be.a("string");
            pm.expect(response[i].status).to.be.a("number");
6
            pm.expect(response[i].traceId).to.be.an("string");
8
            pm.expect(response[i].errors).to.be.an("object");
9
            pm.expect(response[i].errors.Name).to.be.an("string");
10
11 });
12 \( pm.test("Content-Type is present", function () \{
13
        pm.response.to.have.header("Content-Type");
   });
14
15
16 \( pm.test("Content-Type is JSON", function () \{
        pm.response.to.be.json;
17
18 });
19 \( pm.test("Response must have a body", function () \{
         pm.response.to.be.withBody;
20
         pm.response.to.be.json;
21
22
    });
23 \rightarrow pm.test("Status code is 400",() => {
24 pm.response.to.have.status(400);
25 });
```

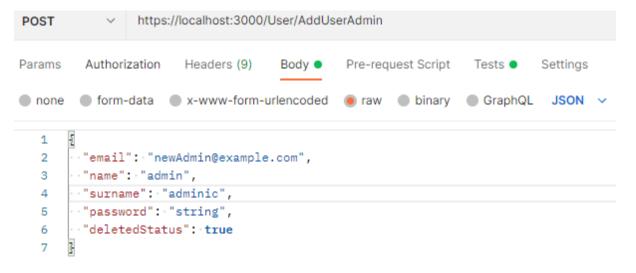
Rezultati testiranja :

Body	Cookies	Headers (11)	Test Results	(5/5)	®	Status: 400 Bad Request
All	Passed	l Skipped	Failed			
PASS	The res	ponse has all p	roperties with	the expected type		
PASS	Conten	t-Type is prese	nt			
PASS	Conten	t-Type is JSON				
PASS	Respon	se must have a	body			
PASS	Status	code is 400				

Testovi za rutu AddUserAdmin

3) Ispravni podaci

Body:



Response Body:

```
Body Cookies Headers (12) Test Results (5/5)
  Pretty
                    Preview
                                Visualize
                                            JSON
            Raw
    1
            "id": 25,
    2
    3
            "email": "newAdmin@example.com",
            "name": "admin",
    4
            "surname": "adminic",
    5
            "password": "string",
    6
            "deletedStatus": true
    7
    8
```

Tests:

```
const response = pm.response.json();
 2 Vpm.test("The response has all properties with the expected type", () => {
        for(var i = 0; i < response.length; i++){</pre>
3 🗸
            pm.expect(response[i].type).to.be.an("string");
 5
             pm.expect(response[i].title).to.be.a("string");
            pm.expect(response[i].status).to.be.a("number");
            pm.expect(response[i].traceId).to.be.an("string");
            pm.expect(response[i].errors).to.be.an("object");
8
9
            pm.expect(response[i].errors.Name).to.be.an("string");
10
11 });
12 \( pm.test("Content-Type is present", function () \{
13
        pm.response.to.have.header("Content-Type");
14
    });
15
16 \rightarrow pm.test("Content-Type is JSON", function () {
17
        pm.response.to.be.json;
18 });
19 \( pm.test("Response must have a body", function () \{
         pm.response.to.be.withBody;
20
21
         pm.response.to.be.json;
22
    });
23 \rightarrow pm.test("Status code is 400",() => {
24 pm.response.to.have.status(400);
25 });
```

Rezultati testiranja:

Status: 201 Created

All Passed Skipped Failed

PASS The response has all properties with the expected type

PASS Content-Type is present

PASS Content-Type is JSON

PASS Response must have a body

PASS Status code is 201

4) Neispravni podaci, prazno polje za password

Body:

```
POST
                https://localhost:3000/User/AddUserAdmin
       Authorization
                       Headers (9)
                                    Body 

                                              Pre-request Script
                                                                          Settings
Params
                                                                Tests •
none form-data x-www-form-urlencoded nraw binary GraphQL
                                                                           JSON ~
   1
       "email": "newAdmin@example.com",
   2
       ··"name": ·"admin",
   4
       ··"surname": "adminic",
       · "password": · "",
   5
       "deletedStatus": true
   6
   7
```

Tests:

```
const response = pm.response.json();
1
2
    pm.test("The response has all properties with the expected type", () => {
         for(var i = 0; i < response.length; i++){</pre>
3
4
            pm.expect(response[i].id).to.be.an("number");
            pm.expect(response[i].email).to.be.a("string");
5
6
            pm.expect(response[i].name).to.be.a("string");
7
            pm.expect(response[i].surname).to.be.an("string");
8
            pm.expect(response[i].password).to.be.an("object");
9
            pm.expect(response[i].deletedStatus).to.be.an("boolean");
10
    });
11
12
    pm.test("Content-Type is present", function () {
13
        pm.response.to.have.header("Content-Type");
14
    });
15
    pm.test("Content-Type is JSON", function () {
16
17
        pm.response.to.be.json;
18
    {);
    pm.test("Response must have a body", function () {
19
20
         pm.response.to.be.withBody;
21
         pm.response.to.be.json;
22
    {);
    pm.test("Status code is 400",() => {
23
        pm.response.to.have.status(400);
24
25
   });
```

Rezultati testiranja :

Body	Cookies	Headers (11)	Test Results	(5/5)	®	Status: 400 Bad Request
All	Passed	l Skipped	Failed			
PASS	The res	ponse has all p	roperties with	the expected type		
PASS	Conten	t-Type is prese	nt			
PASS	Conten	t-Type is JSON				
PASS	Respon	se must have a	body			
PASS	Status	code is 400				