

Testovi za Task 4

Task	Test Scenario #	Scenario ID	Test Scenario Description	Test Cases
	1	S1.1	Testovi za rutu AddUser – ispravni podaci	1. Provjera ispravnosti tipa odgovora 2. Provjera postojanja content-type 3. Provjera da li je content-type u JSON formatu 4. Provjera da li response ima body 5. Provjera ispravnog statusnog koda
	2	S1.2	Testovi za rutu AddUser – neispravni podaci, prazno polje za unos imena	1. Provjera ispravnosti tipa odgovora 2. Provjera postojanja content-type 3. Provjera da li je content-type u JSON formatu 4. Provjera da li response ima body 5. Provjera ispravnog statusnog koda
	3	S2.1	AddUserAdmin – ispravni podaci	1. Provjera ispravnosti tipa odgovora 2. Provjera postojanja content-type 3. Provjera da li je content-type u JSON formatu 4. Provjera da li response ima body 5. Provjera ispravnog statusnog koda
	4	S2.2	Testovi za rutu AddUserAdmin – neispravni podaci, prazno polje	1. Provjera ispravnosti tipa odgovora 2. Provjera postojanja content-type 3. Provjera da li je content-type u JSON formatu 4. Provjera da li response ima body 5. Provjera ispravnog statusnog koda

Testovi za rutu **AddUser**

1) Ispravni podaci

Body :

POST
▼
https://localhost:3000/User/AddUser

Params
Authorization
Headers (9)
Body ●
Pre-request Script
Tests ●
Settings

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

```

1  {
2    "email": "newuser@example.com",
3    "name": "string",
4    "surname": "string",
5    "password": "string",
6    "deletedStatus": true
7  }

```

Response Body :

Body Cookies Headers (12) Test Results (5/5)

Pretty Raw Preview Visualize JSON


```
1 {
2   "id": 24,
3   "email": "newuser@example.com",
4   "name": "string",
5   "surname": "string",
6   "password": "string",
7   "deletedStatus": true
8 }
```

Tests :

```
1  const response = pm.response.json();
2  pm.test("The response has all properties with the expected type", () => {
3    for(var i = 0; i < response.length; i++){
4      pm.expect(response[i].id).to.be.an("number");
5      pm.expect(response[i].email).to.be.a("string");
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
16 pm.test("Content-Type is JSON", function () {
17   pm.response.to.be.json;
18 });
19 pm.test("Response must have a body", function () {
20   pm.response.to.be.withBody;
21   pm.response.to.be.json;
22 });
23 pm.test("Status code is 201", () => {
24   pm.response.to.have.status(201);
25 });
```

Rezultati testiranja :

Body Cookies Headers (12) Test Results (5/5)

 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

2) Neispravni podaci, prazno polje za unos imena

Body :

POST  https://localhost:3000/User/AddUser

Params Authorization Headers (9) Body  Pre-request Script Tests Settings

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

```
1 {  
2   "id": 10,  
3   "email": "user@example.com",  
4   "name": "",  
5   "surname": "prezime",  
6   "password": "pass",  
7   "deletedStatus": true  
8 }
```

Response Body :

Body Cookies Headers (11) Test Results

Pretty

Raw

Preview

Visualize

JSON



```
1  {
2    "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
3    "title": "One or more validation errors occurred.",
4    "status": 400,
5    "traceId": "00-598c9c0120701f3aa2e12c473c3dbdd9-4744af26f56fb9a5-00",
6    "errors": {
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", () => {
3    for(var i = 0; i < response.length; i++){
4      pm.expect(response[i].type).to.be.an("string");
5      pm.expect(response[i].title).to.be.a("string");
6      pm.expect(response[i].status).to.be.a("number");
7      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 () {
17   pm.response.to.be.json;
18 });
19 pm.test("Response must have a body", function () {
20   pm.response.to.be.withBody;
21   pm.response.to.be.json;
22 });
23 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

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 400

Testovi za rutu **AddUserAdmin**

3) Ispravni podaci

Body :

POST ▼ https://localhost:3000/User/AddUserAdmin

Params Authorization Headers (9) **Body** ● Pre-request Script Tests ● Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▼

```
1 {  
2   "email": "newAdmin@example.com",  
3   "name": "admin",  
4   "surname": "adminic",  
5   "password": "string",  
6   "deletedStatus": true  
7 }
```

Response Body :

Body Cookies Headers (12) Test Results (5/5)

Pretty Raw Preview Visualize JSON ▼ ≡


```
1 {  
2   "id": 25,  
3   "email": "newAdmin@example.com",  
4   "name": "admin",  
5   "surname": "adminic",  
6   "password": "string",  
7   "deletedStatus": true  
8 }
```

Tests :

```
1  const response = pm.response.json();
2  pm.test("The response has all properties with the expected type", () => {
3    for(var i = 0; i < response.length; i++){
4      pm.expect(response[i].type).to.be.an("string");
5      pm.expect(response[i].title).to.be.a("string");
6      pm.expect(response[i].status).to.be.a("number");
7      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 () {
17   pm.response.to.be.json;
18 });
19 pm.test("Response must have a body", function () {
20   pm.response.to.be.withBody;
21   pm.response.to.be.json;
22 });
23 pm.test("Status code is 400", () => {
24   pm.response.to.have.status(400);
25 });
```

Rezultati testiranja :

Body Cookies Headers (12) Test Results (5/5)

 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`


Params

Authorization


Headers (9)


Body 


Pre-request Script


Tests 


Settings


 none


 form-data

 x-www-form-urlencoded

 **raw**

 binary

 GraphQL

JSON 

```
1  {}
2  .. "email": "newAdmin@example.com",
3  .. "name": "admin",
4  .. "surname": "adminic",
5  .. "password": "",
6  .. "deletedStatus": true
7  }
```

Tests :

```
1  const response = pm.response.json();
2  pm.test("The response has all properties with the expected type", () => {
3      for(var i = 0; i < response.length; i++){
4          pm.expect(response[i].id).to.be.an("number");
5          pm.expect(response[i].email).to.be.a("string");
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
16 pm.test("Content-Type is JSON", function () {
17     pm.response.to.be.json;
18 });
19 pm.test("Response must have a body", function () {
20     pm.response.to.be.withBody;
21     pm.response.to.be.json;
22 });
23 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

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 400