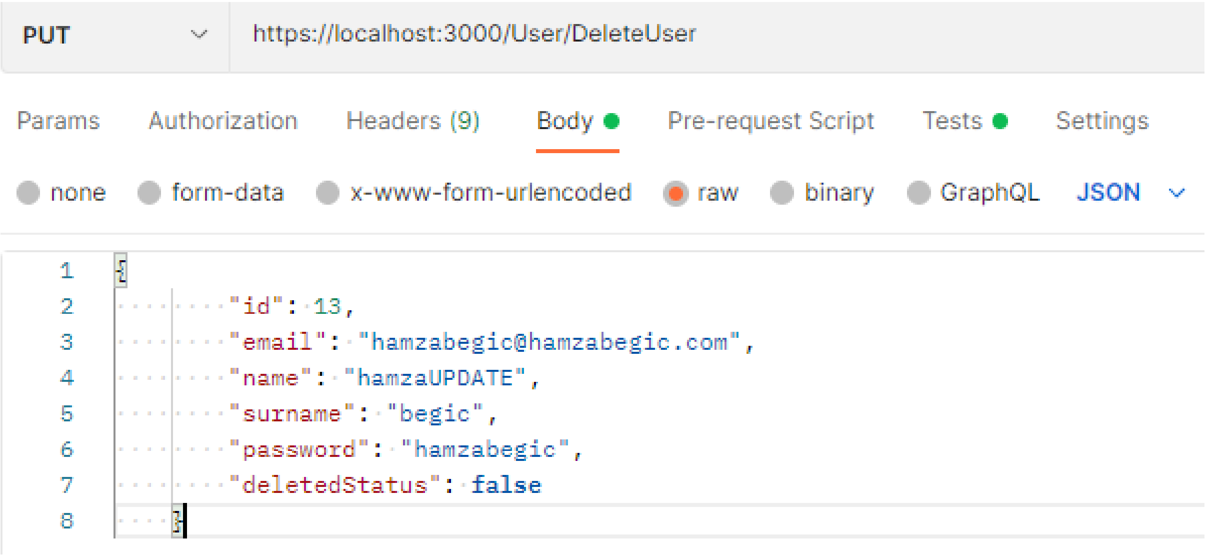
Testovi za Task 7

Task	Test Scenario #	Scenario ID	Test Scenario Description	Test Cases
1	1	S1.1	Testovi za rutu DeleteUser– ispravni podaci	1. Provjera ispravnog statusnog koda
	2	S1.2	Testovi za rutu DeleteUser – 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

Testovi za rutu **AddUser**

1) Ispravni podaci

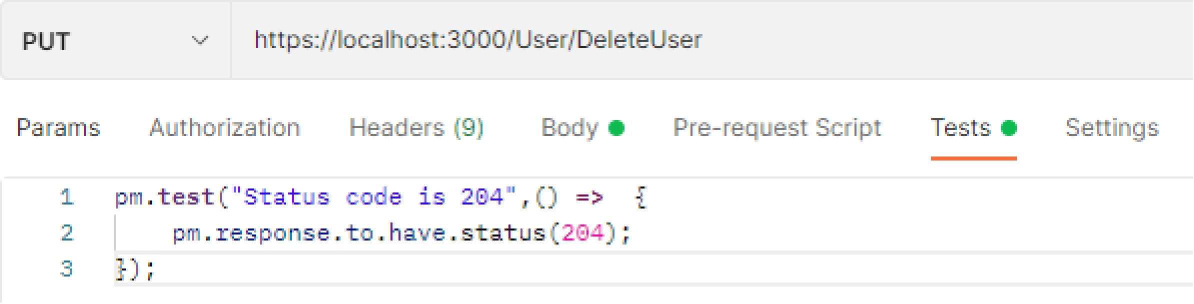
Body :



PUT https://localhost:3000/User/DeleteUser

```
{
        "id": 13,
        "email": "hamzabegic@hamzabegic.com",
        "name": "hamzaUPDATE",
        "surname": "begic",
        "password": "hamzabegic",
        "deletedStatus": false
    }
```

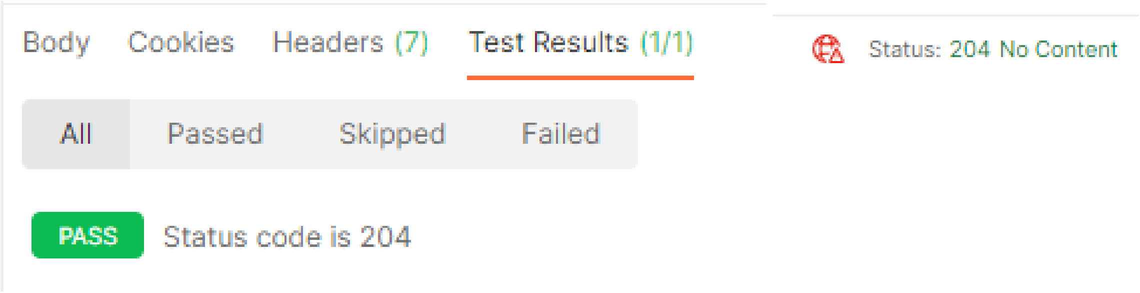
Tests :



PUT https://localhost:3000/User/DeleteUser

```
pm.test("Status code is 204",() =>  {
    pm.response.to.have.status(204);
});
```

Rezultati testiranja :

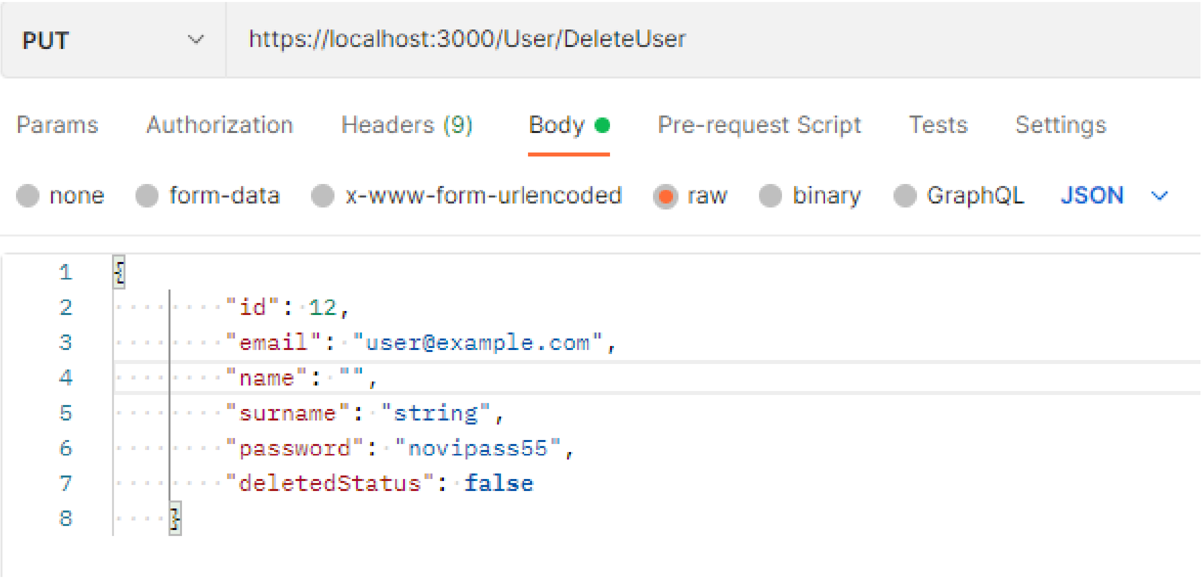


Status: 204 No Content

PASS Status code is 204

2) Neispravni podaci, prazno polje za unos imena

Body :

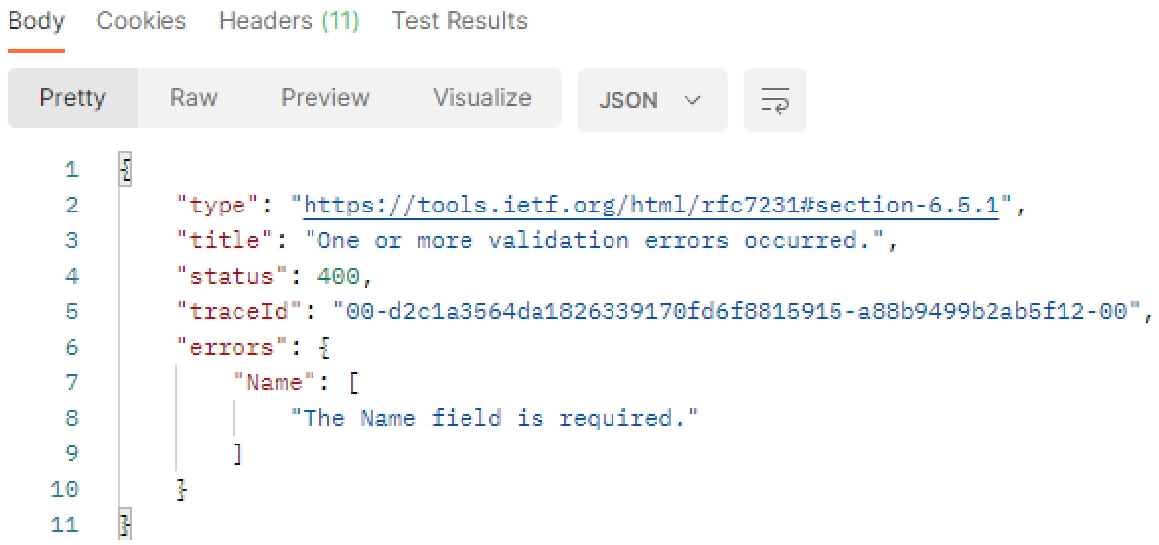


PUT https://localhost:3000/User/DeleteUser

● raw JSON

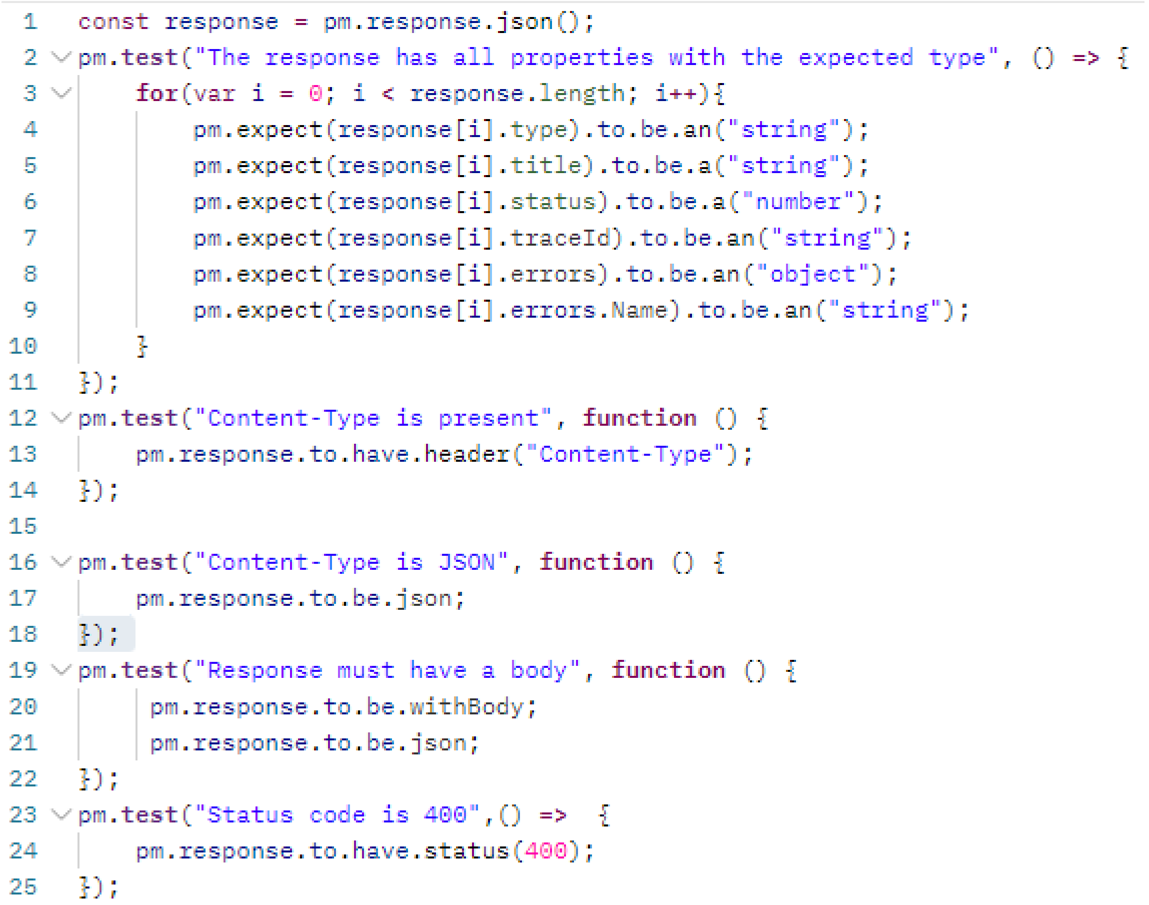
```json
{
    "id": 12,
    "email": "user@example.com",
    "name": "",
    "surname": "string",
    "password": "novipass55",
    "deletedStatus": false
    }
```

Response Body :



```json
{
    "type": "https://tools.ietf.org/html/rfc7231#section-6.5.1",
    "title": "One or more validation errors occurred.",
    "status": 400,
    "traceId": "00-d2c1a3564da1826339170fd6f8815915-a88b9499b2ab5f12-00",
    "errors": {
        "Name": [
            "The Name field is required."
        ]
    }
}
```

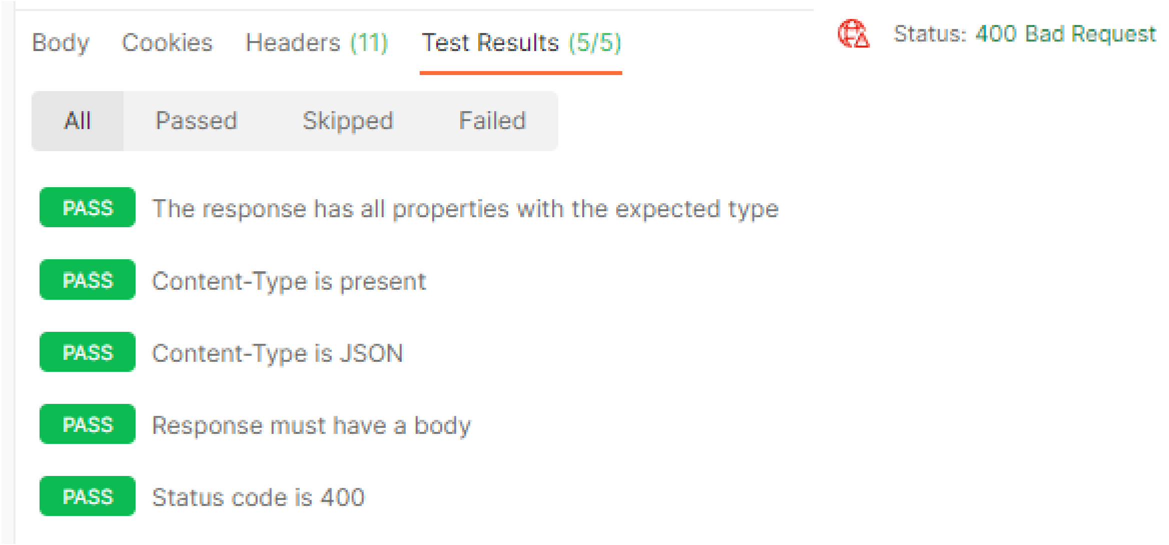
Tests :



```
const response = pm.response.json();
pm.test("The response has all properties with the expected type", () => {
    for(var i = 0; i < response.length; i++){
        pm.expect(response[i].type).to.be.an("string");
        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");
        pm.expect(response[i].errors.Name).to.be.an("string");
    }
});
pm.test("Content-Type is present", function () {
    pm.response.to.have.header("Content-Type");
});

pm.test("Content-Type is JSON", function () {
    pm.response.to.be.json;
});
pm.test("Response must have a body", function () {
     pm.response.to.be.withBody;
     pm.response.to.be.json;
});
pm.test("Status code is 400",() =>  {
    pm.response.to.have.status(400);
});
```

Rezultati testiranja :



Status: 400 Bad Request

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