

VVS PROJEKAT 2 – ZADATAK 2

API testovi – <https://restcountries.com>

1. Test

Ruta: <https://restcountries.com/v3.1/name/deutschland>

Metoda: GET

4 validacije:

Response status, format odgovora (json), name.common i name.official

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** <https://restcountries.com/v3.1/name/deutschland>
- Tests Tab:** Contains four test cases:

```
1 pm.test('response is ok', function () {
2   pm.response.to.have.status(200);
3 })
4
5 pm.test('response is json', function () {
6   pm.response.to.be.json;
7 })
8
9 pm.test('name correct', function () {
10  pm.expect(pm.response.json()[0].name.common).to.equal('Germany');
11 })
12
13 pm.test('official correct', function () {
14  pm.expect(pm.response.json()[0].name.official).to.equal('Federal Republic of Germany');
15 })
```
- Test Results (4/4):** All tests passed.
 - PASS** response is ok
 - PASS** response is json
 - PASS** name correct
 - PASS** official correct
- Status:** 200 OK
- Time:** 5.05 s

2. Test

Ruta: <https://restcountries.com/v3.1/capital/tallinn>

Metoda: GET

4 validacije: response status, format odgovora, name.common, name.official

The screenshot shows a REST client interface with the following components:

- Request Bar:** Method **GET**, URL **https://restcountries.com/v3.1/capital/tallinn**.
- Navigation Tabs:** Params, Authorization, Headers (6), Body, Pre-request Script, **Tests** (active), Settings.
- Test Script:**

```
1 pm.test('response is ok', function () {  
2   pm.response.to.have.status(200);  
3 })  
4  
5 pm.test('response is json', function () {  
6   pm.response.to.be.json;  
7 })  
8  
9 pm.test('name correct', function () {  
10  pm.expect(pm.response.json()[0].name.common).to.equal('Estonia');  
11 })  
12  
13 pm.test('official correct', function () {  
14  pm.expect(pm.response.json()[0].name.official).to.equal('Republic of Estonia');  
15 })
```
- Response Bar:** Body, Cookies, Headers (8), **Test Results (4/4)** (active), Status: 200 OK, Time.
- Test Results Summary:** Perform API tests faster with templates for integration testing, regression testing, and more.
- Test Results Table:**

All	Passed	Skipped	Failed	Refresh
PASS	response is ok			
PASS	response is json			
PASS	name correct			
PASS	official correct			

3. Test

Ruta: <https://restcountries.com/v3.1/translation/bosnia>

Metoda: GET

4 validacije: response status, format odgovora, name.common, capital

The screenshot shows the Postman interface for a REST client. The top bar indicates the request is a GET to `https://restcountries.com/v3.1/translation/bosnia`. The 'Tests' tab is active, displaying four test cases in a code editor:

```
1 pm.test('response is ok', function () {
2   pm.response.to.have.status(200);
3 })
4
5 pm.test('response is json', function () {
6   pm.response.to.be.json;
7 })
8
9 pm.test('name correct', function () {
10  pm.expect(pm.response.json()[0].name.common).to.equal('Bosnia and Herzegovina');
11 })
12
13 pm.test('capital correct', function () {
14  pm.expect(pm.response.json()[0].capital[0]).to.equal('Sarajevo');
15 })
```

Below the code editor, the 'Test Results (4/4)' section shows the outcome of the tests:

- PASS** response is ok
- PASS** response is json
- PASS** name correct
- PASS** capital correct

The status bar at the bottom right indicates 'Status: 200 OK' and 'Time: 4.95 s'.

AZURE:

```
azure-pipelines-amna.yml
Edit

Contents History Compare Blame

1 pool:
2   vmImage: 'windows-latest'
3
4 steps:
5 - script: |
6   powershell.exe -NoProfile -InputFormat None -ExecutionPolicy AllSigned -Command "[System.Net.ServicePointManager]::SecurityProt
7   displayName: 'Install Postman CLI'
8
9 - task: CmdLine@2
10  displayName: 'Run automated API tests using Postman CLI'
11  inputs:
12    script: |
13      postman login --with-api-key PMAK-65a7ef5a4802590f70d061e5-b991003796c4b637ce61e7dc823c7d71bf
14      # Run your collection using Postman CLI
15      postman collection run "27702635-d604c6eb-6fc6-44db-b0da-97da5789b07c"
```

```
✓ Run automated API tests using Postman CLI
View

1 Starting: Run automated API tests using Postman CLI
2 =====
3 Task      : Command line
4 Description : Run a command line script using Bash on Linux and macOS and cmd.exe on Windows
5 Version    : 2.231.1
6 Author     : Microsoft Corporation
7 Help       : https://docs.microsoft.com/azure/devops/pipelines/tasks/utility/command-line
8 =====
9 Generating script.
10 ===== Starting Command Output =====
11 "C:\Windows\system32\cmd.exe" /D /E:ON /V:OFF /S /C "CALL "D:\a\_temp\5343eb8b-8a8b-4bb2-91c7-c4f1804cbe74.cmd""
12 Logged in using api key of user: security-technologist-9407034
13 Logged in successfully.
14 '#' is not recognized as an internal or external command,
15 operable program or batch file.
16 postman
17
18 REST Countries
19
20 → name
21 GET https://restcountries.com/v3.1/name/deutschland [200 OK, 1.87kB, 191ms]
22 ✓ response is ok
23 ✓ response is json
24 ✓ name correct
25 ✓ official correct
26
27
```

Activate Windows
Go to Settings to activate

→ capital

GET <https://restcountries.com/v3.1/capital/tallinn> [200 OK, 1.64kB, 36ms]

- ✓ response is ok
- ✓ response is json
- ✓ name correct
- ✓ official correct

→ translation

GET <https://restcountries.com/v3.1/translation/bosnia> [200 OK, 1.85kB, 39ms]

- ✓ response is ok
- ✓ response is json
- ✓ name correct
- ✓ capital correct

✓ capital correct

	executed	failed
iterations	1	0
requests	3	0
test-scripts	3	0
prerequisite-scripts	0	0
assertions	12	0
total run duration: 531ms		
total data received: 4.53kB (approx)		
average response time: 88ms [min: 36ms, max: 191ms, s.d.: 72ms]		

Postman CLI run data uploaded to Postman Cloud successfully.

You can view the run data in Postman at: <https://go.postman.co/workspace/6156ce89-24a5-4fd6A9ed1v437ae6181ac1/run/27702635-ee064ae6-1a01-47e3-bb1ac1-nw>

Finishing: Run automated API tests using Postman CLI

Go to Settings to activate Windows.