

* Projekat ojačavanja -

Automatizacija API testova pomoću Postmna

Niš, 2020

Sadržaj

[Analiza API-a](#_heading=h.1t3h5sf) **3**

[Pisanje API-poziva u Postmenu](#_heading=h.4d34og8) **4**

[Pisanje API testova](#_heading=h.2s8eyo1) **4**

[Auth testovi](#_heading=h.17dp8vu) 5

[City testovi](#_heading=h.3rdcrjn) 6

[Profile testovi](#_heading=h.26in1rg) 7

[User testovi](#_heading=h.lnxbz9) 9

### Analiza API-a

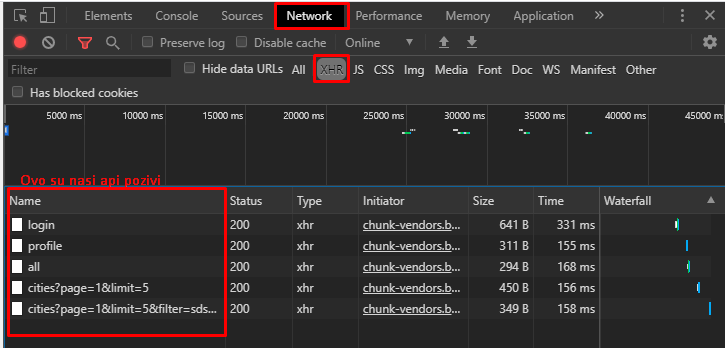
Test API <https://vue-demo.daniel-avellaneda.com/>

Klijent za API koji se testira <https://vue-demo.daniel-avellaneda.com/>

Dokumentacija Rest API-a je [yaml fajl](https://drive.google.com/file/d/1B3mog4ag5t1ivfy3SrYUwG62OZKU-n6M/view?usp=sharing), koji otvarate u [Swagger editoru](https://editor.swagger.io/) tako što izaberete *File > Import File > izaberete pomenuti yaml fajl*

Koristan video trening: Testiranje API-a pomoću Postmena možete kupiti na [udemy-u](https://www.udemy.com/course/postman-the-complete-guide/) ili skinuti sa ovo [linka](https://www.freetutorialsus.com/postman-the-complete-guide-rest-api-testing-udemy-course-free-download/).

Da bismo uspešno istestirali API, moramo se prvo upoznati sa logikom i videti u kojim situacijama se koristi dati API pozivi. U tome će nam dosta pomoći analiza same [klijentske aplikacije](https://vue-demo.daniel-avellaneda.com/). Potrebno je upoznati se sa aplikacijom, i saznati što više informacija za svaki od navedenih api poziva. Sve potrebne informacije možete naći u Network prozoru kao na slici 1.



Slika 1. Praćenje API poziva iz Network prozora

Na kraju dolazimo do prvog zadatka, a to je:

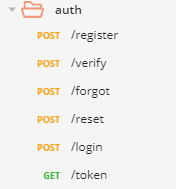
U [eksel tabeli](https://docs.google.com/spreadsheets/d/130TT1WYVlTqANAQEWMIm65pdPnJbHePskbJ_NaiRSAI/edit?usp=sharing), **klonirajte** sheet Sablon i nazovite ga svojim imenom a zatim popunite tabelu.

Za razmevanje procesa prijavljivanja i svrhe u koju se koristi token, potrebno je upoznati se sa [JWT tokenom](https://jwt.io/). Koristan video trening na tu temu možete pogledati na [youtube-u](https://www.youtube.com/watch?v=7Q17ubqLfaM).

### Pisanje API-poziva u Postmenu

Smernice kojima bi se trebali voditi:

* imena poziva su ista kao imena ruta



* Upotreba globalnih promenljivi ili promenljivi okruženja za:
  + naziv host-a, npr: {{base\_url}}/registracija ([udemy](https://www.udemy.com/course/postman-the-complete-guide/), Writing tests and scripts using variables)
  + tokene ([udemy](https://www.udemy.com/course/postman-the-complete-guide/), Authentication / Authorization > JSON Web Tokens (JWT))
  + promenljive podatke, npr: login podaci, id-evi,
* Podatke iz dokumentacije koji se šalju kao *formData*, šaljite u body kao **x-www-form-urlencoded**

Napomena 1: Validne vrednosti za user role su “admin” i “user”

Napomena 2: Token je deo zaglavlja i mora se unositi u formatu sa ključnom rečju Bearer **{{token}}**

**https://api-demo.daniel-avellaneda.com/**

### Pisanje API testova

U Postmenu testovi se pišu u JavaScript-u, tako da bi bilo dobro upoznati se malo sa tim programskim jezikom ([udemy](https://www.udemy.com/course/postman-the-complete-guide/): *JavaScript fundamentals, Writing tests and scripts, Writing tests and scripts using variables, Advanced assertions, Running a request multiple times with different data sets, ...*)

Primeri nekih osnovnih testova [link](https://documenter.getpostman.com/view/1559645/RzZFCGFR?version=latest) (Otvorite u Postmanu klikom na dugme Run in Postman). Neki od njih su zastareli ali je moguce naći novu veriju na [linku](https://learning.postman.com/docs/postman/scripts/test-examples/).

Postoji mnogo načina za kreiranje testova, ali moja stroga preporuka je koristiti [Chai Assertion Library](https://www.chaijs.com/) ([udemy](https://www.udemy.com/course/postman-the-complete-guide/): *Advanced assertions* )

U početku će vam biti teže da se povežete sa testiranjem u Postmanu, preporuka je da za debugiranje koristite konzolu ([udemy](https://www.udemy.com/course/postman-the-complete-guide/): *Writing tests and scripts using variables > Debugging tests* )

Potrebno je implementirati testove koji su navedeni u daljem tekstu. Postoje 4 logičke grupe testova:

* auth testovi
* city testovi
* profile testovi
* user testovi

#### **Auth testovi**

|  |  |  |  |
| --- | --- | --- | --- |
| Zahtev | Naziv testa | Testiranje odgovora | Opis |
| GET / | it should GET home API url | Status 200 | Testiranje hosta. Testiramo glavnu rutu |
| POST /login | it should GET token | Status 200 | Testiranje prijavljivanja sa validnim podacima za prijavu. |
| body treba da bude object |
| u body-ju treba da postoji property token i user |
| POST /register | it should POST register | Status 201 | Testiranje registracije sa validnim podacima.(Sačuvati verification property za buduće potrebe)  Ako vam bude trebalo generisanje generisanje unique emailova mozete baciti pogled na [link](https://sqa.stackexchange.com/questions/36809/how-to-randomly-generate-email-address-on-postman-request-using-pre-request-scri) |
| body treba da bude object |
| body treba da ima propertije token i user |
| POST /register | it should NOT POST a register if email already exists | Status 422 | testiranje registracije sa postojecim emailom |
| body treba da bude object |
| body treba da ima property errors |
| errors treba da ima property msg i da bude jednak EMAIL\_ALREADY\_EXISTS |
| POST /verify | it should POST verify | Status 200 | Testiranje verifikacije sa validnim id-jem od get Profile sa tokenom bez ““. |
| body treba da bude object |
| body treba da ima propertije email i verified |
| verified treba da bude true |
| POST /forgot | it should POST forgot | Status 200 | Testiranje sa validnim email-om. (Sacuvati verification sa sledece API pozive) |
| body treba da bude object |
| body treba da ima propertije msg i verification |
| POST /reset | it should POST reset | Status 200 | Resetovanje sa validnim passwordom i id-em. (verification sačuvan iz prethodnog api poziva se salje) |
| body treba da bude object |
| body treba da ima property msg koji treba da ima vrednost PASSWORD\_CHANGED |
| GET /token | it should NOT be able to consume the route since no token was sent | Status 401 | Testiranje api poziva u kom se ne salje token. |
| GET /token | it should GET a fresh token | Status 200 | Testiranje api poziva slanjem tokena |
| body treba da bude object |
| body treba da ima property token |

#### **City testovi**

|  |  |  |  |
| --- | --- | --- | --- |
| Zahtev | Naziv testa | Testiranje odgovora | Opis |
| POST /login | it should GET token (city) | Status 200 | Testiranje prijavljivanja sa validnim podacima za prijavu. |
| body treba da bude object |
| u body-ju treba da postoji property token |
| GET /cities | it should NOT be able to consume the route since no token was sent | Status 401 | Testiranje api poziva u kom se ne salje token. |
| GET /cities | it should GET all the cities | Status 200 | Testiranje api poziva u kom se salje token. |
| body treba da bude object |
| property docs treba da bude niz |
| GET /cities?filter=Bucaramanga&fields=name | it should GET the cities with filters | Status 200 | Testiranje api poziva u kom se salje token.  (Sacuvati id grada, trebace za naredne pozive) |
| body treba da bude object |
| property docs treba da bude niz |
| docs treba da bude niz duzine 1. |
| docs[0] treba da ima property name i name treba da ima vrednost Bucaramanga |
| POST /city | it should NOT POST a city without name | Status 422 | Testiranje api poziva u kom se salje token.  Za city se ne salje name. |
| body treba da bude object |
| body treba da ima property errors |
| POST /city | it should POST a city | Status 201 | Testiranje api poziva u kom se salje token. |
| body treba da bude object |
| body treba da ima propertije name i \_id |
| POST /city | it should NOT POST a city that already exists | Status 422 | Testiranje api poziva u kom se salje token.  Kreiranje grada sa vec postojecim imenom. |
| body treba da bude object |
| body treba da ima property errors |
| GET /city/:id | it should GET a city by the given id | Status 200 | Testiranje api poziva u kom se salje token. (Poslati id grada koji je sacuvan u prethodnom pozivu) |
| body treba da bude object |
| body treba da ima property name |
| body treba da ima property id, i da je id jednam poslatom id-ju. |
| PATCH /city/:id | it should UPDATE a city given the id | Status 200 | Testiranje api poziva u kom se salje token.  Kreiranje grada sa vec postojecim imenom. |
| body treba da bude object |
| body treba da ima property id, i da je id jednam poslatom id-ju. |
| body treba da ima property name, i da name ima vrednost prosledjenog naziva (novo ima grada) |
| DELETE /city/:id | it should DELETE a city given the id | Status 200 | Testiranje api poziva u kom se salje token. |
| body treba da bude object |
| body treba da ima property msg i da msg ima vrednost DELETED |

#### **Profile testovi**

|  |  |  |  |
| --- | --- | --- | --- |
| Zahtev | Naziv testa | Testiranje odgovora | Opis |
| POST /login | it should GET token (profile) | Status 200 | Testiranje prijavljivanja sa validnim podacima za prijavu. |
| body treba da bude object |
| u body-ju treba da postoji property token |
| GET /profile | it should NOT be able to consume the route since no token was sent | Status 401 | Testiranje api poziva u kom se ne salje token. |
| GET /profile | it should GET profile | Status 200 | Testiranje api poziva u kom se salje token. |
| body treba da bude object |
| u body-ju treba da postoji propertiji email i name |
| PATCH /profile | it should NOT UPDATE profile empty name/email | Status 422 | Testiranje api poziva u kom se salje token. (Email i name su prazni) |
| body treba da bude object |
| u body-ju treba da postoji property errors |
| PATCH /profile | it should UPDATE profile | name == Test123456  urlTwitter == https://hello.com  urlGitHub == https://hello.io  phone == 123123123  city == Bucaramanga  country == Colombia | Testiranje api poziva u kom se salje token.  user = {  name: 'Test123456',  urlTwitter: 'https://hello.com',  urlGitHub: 'https://hello.io',  phone: '123123123',  city: 'Bucaramanga',  country: 'Colombia'  } |
| Status 200 |
| body treba da bude object |
| PATCH /profile | it should NOT UPDATE profile with email that already exists | Status 422 | Testiranje api poziva u kom se salje token.  Testirati sa emailom programmer@programmer.com |
| body treba da bude object |
| body treba da ima property errors |
| PATCH /profile | it should NOT UPDATE profile with not valid URL´s | Staus 422 | Testiranje api poziva u kom se salje token.  user = {  name: 'Test123456',  urlTwitter: 'hello',  urlGitHub: 'hello',  phone: '123123123',  city: 'Bucaramanga',  country: 'Colombia'  } |
| body treba da bude object |
| body treba da ima property errors |
| errors treba da imaju property msg |
| msg[0] treba da ima property msg koji je jednak NOT\_A\_VALID\_URL |
| POST profile/changePassword | it should NOT change password | Status 409 | Testiranje api poziva u kom se salje token.  Kao oldPassword salje se nevazeci password. |
| body treba da bude object |
| body treba da ima property errors |
| errors treba da ima property msg koji je jednak WRONG\_PASSWORD |
| POST profile/changePassword | it should NOT change a too short password | Status 422 | Testiranje api poziva u kom se salje token.  Novi password treba da ima manje od 5 karaktera. |
| body treba da bude object |
| body treba da ima property errors |
| msg[0] treba da ima property msg koji je jednak PASSWORD\_TOO\_SHORT\_MIN\_5 |
| POST profile/changePassword | it should change password | Status 200 | Testiranje api poziva u kom se salje token. Salju se validne lozinke. |
| body treba da bude object |
| body treba da ima property msg, koji je jednak PASSWORD\_CHANGED |

Podatke koje mozete koristiti za user testove. (biće vam potrebe promenljive za admin token i user token).



#### **User testovi**

|  |  |  |  |
| --- | --- | --- | --- |
| Zahtev | Naziv testa | Testiranje odgovora | Opis |
| POST /login | it should GET token as admin (user) | Status 200 | Salju se admin podaci. (Sacuvati token sa responsa u admin token) |
| body treba da bude object |
| u body-ju treba da postoji property token |
| POST /login | it should GET token as user (user) | Status 200 | Salju se user podaci. (Sacuvati token sa responsa u user token) |
| body treba da bude object |
| u body-ju treba da postoji property token |
| GET /users | it should NOT be able to consume the route since no token was sent | Status 401 | Ne salje se token. |
| GET /users | it should GET all the users | Status 200 | Salje se admin token. |
| body treba da bude object |
| u body-ju treba da postoji docs i da je niz. |
| GET /users?filter=admin&fields=name,email,city,country,phone | it should GET the users with filters | Status 200 | Salje se admin token. |
| body treba da bude object |
| u body-ju treba da postoji docs i da je niz. |
| dosc treba da bude duzine 1. |
| docs[0] treba da ima propery email i da bude jednak admin@admin.com |
| POST /user | it should NOT POST a user without name | Status 422 | Salje se admin token. |
| body treba da bude object |
| body treba da ima property errors |
| POST /user | it should POST a user | Status 201 | Salje se admin token.  Salje se validan user objekat.  (Sacuvati id usera, za kasnije) |
| body treba da bude object |
| body treba da ima propertije \_id, name, email, verification |
| POST /user | it should NOT POST a user with email that already exists | Status 422 | Salje se admin token.  Salje se user sa emailom koji vec postoji u bazi. |
| body treba da bude object |
| body treba da ima property errors |
| POST /user | it should NOT POST a user with not known role | Status 422 | Salje se admin token.  Validne role su “admin” i “user”. |
| body treba da bude object |
| body treba da ima property errors |
| GET /user/:id | it should GET a user by the given id | Status 200 | Salje se admin token. |
| body treba da bude object |
| body treba da ima property name |
| body treba da ima property \_id i da je taj id jednak prosledjenom. |
| PATCH /user/:id | it should UPDATE a user given the id | Status 200 | Salje se admin token.  Salje se novo ime za usera i novi email koji ne postoji u bazi. |
| body treba da bude object |
| body treba da ima property name i da je jednak prosledjenom name-u |
| body treba da ima property \_id i da je taj id jednak prosledjenom. |
| body treba da ima property email i prosledjenom emailu |
| PATCH /user/:id | it should NOT UPDATE a user with email that already exists | Status 422 | Salje se admin token.  Salje se email korisnka koji vec postoji u bazi npr: admin@admin.com |
| body treba da bude object |
| body treba da ima property errors |
| POST /user | it should POST a user (deleting test) | Status 201 | Salje se admin token.  Sacuvati id usera jer se taj user brise u sledecem api pozivu. |
| body treba da bude object |
| body treba da ima propertije \_id, name, email, verification |
| DELETE /user/:id | it should DELETE a user given the id | Status 201 | Salje se admin token.  Salje se id usera koji je kreiran u prethodnom api pozivu. |
| body treba da bude object |
| body treba da ima property msg i da taj msg ima vrednost DELETED |

**Bonus deo**  
Za neke od ključnih api poziva proveriti da vreme odgovora bude manje od 200ms.

}