



QA Automation Zadatak

Cilj zadatka je da proceni veštine kandidata u testiranju API-ja i razumevanje QA procesa. Zadatak se sastoji od pitanja koja će omogućiti kandidatu da pokaže svoje tehničke sposobnosti, analitičko razmišljanje i sposobnost rešavanja problema u kontekstu automatizovanog testiranja.

Fokusirajte se na tačnost rešenja i objašnjenje svog pristupa. Test je dizajniran da proceni vaše sposobnosti i kreativnost, pa shodno tim istražite i koristite alate i tehnike koji vam najviše odgovaraju.

Test se sastoji od 5 zadataka. Potrebno je razviti detaljne automatske testove za svaku od funkcionalnosti ovih 5 API poziva, kako bi se osigurala pokrivenost svih pozitivnih i negativnih scenarija, uključujući “edge” case-ove i druge specifične situacije. Ovi test slučajevi treba da omoguće sveobuhvatnu proveru ispravnosti sistema i detektovanje potencijalnih problema u različitim uslovima.

Moguće je koristiti tehnologiju po izboru.

Base URL:

```
http://127.0.0.1:8000/api/v1/
```

Headers:

```
“Authorization”: Bearer b27912f4-78fd-4d15-977c-1325d4b64732
```



Zadatak 1 – Brisanje korisnika

Kreiranje automatskog testa za funkcionalnost koja omogućava administratorima da obrišu postojeće korisnike iz sistema.

Test case:

1. Pošaljite **GET** zahtev na **/users/** endpoint kako biste dobili listu korisnika.
2. Izaberite bilo kog korisnika iz liste i zabeležite njegov **user uuid**.
3. Pošaljite **DELETE** zahtev na **/users/{uuid}/** endpoint sa prethodno uzetim **user uuid** kao parametrom.
4. Osigurajte da response sa statusom 204 bude primljen.
5. Validirajte da je korisnik obrisani iz liste korisnika slanjem **GET** zahteva na **/users/** endpoint.
6. Verifikujte da informacije o korisniku ne budu vraćene slanjem **GET** zahteva sa **user uuid** na **/users/{uuid}/** endpoint.

Kriterijumi prihvatanja

1. Statusni kod respons-a nakon brisanja je 204.
2. Korisnik se ne prikazuje u listi korisnika.
3. Obrisani korisnik se ne prikazuje kada se pretraži pomoću njegovog **uuid-a**.



Zadatak 2 – Pretraga po ključnoj reči

Testiraj funkcionalnost koja omogućava korisnicima da pretražuju pesme koristeći ključne reči ili fraze.

Test case:

1. Pošaljite **GET** zahtev na **/songs/** endpoint kako biste dobili prikaz liste svih pesama.
2. Izaberite bilo koju pesmu i kopirajte deo njenog imena.
3. Pripremite query parametar za pretragu sa preuzetim delom naziva pesme.

Primer query parametra:

```
{  
  "search": "requiem"  
}
```

4. Pošaljite **GET** zahtev na **/songs/** endpoint sa prethodno pripremljenim parametrom upita.
5. Proverite da je kod statusa respons-a **200**.
6. Proverite da su polja respons-a jednaka poljima iz liste pesama koje sadrže izabranu ključnu reč.

Kriterijumi prihvatanja

1. Statusni kod odgovora je 200.
2. Funkcija pretrage vraća rezultate isključivo na osnovu unetih ključnih reči ili fraza.
3. Sistem ispravno obrađuje prazne ili nevažeće upite za pretragu.
4. Rezultati pretrage su ispravno formatirani i sadrže relevantne informacije o pesmama.



Zadatak 3 – Kreiranje korisnika

Testiraj funkcionalnost koja omogućava administratoru aplikacije za prodaju pesama da kreira nove korisnike u sistemu.

Test case:

1. Ispunite payload zahteva sa validnim podacima prema zahtevima iz dokumentacije.

Primer Payload-a:

```
{  
  "email": "string",  
  "password": "string",  
  "name": "string",  
  "nickname": "string"  
}
```

2. Pošaljite **POST** zahtev na **/users/** endpoint sa prethodno pripremljenim payload-om.
3. Proverite da je statusni kod odgovora **200**.
4. Proverite da polja u odgovoru odgovaraju očekivanim poljima iz dokumentacije i da sadrže vaše podatke.

Primer respons-a:

```
{  
  "email": "john.doe@gmail.com",  
  "name": "John Doe",  
  "nickname": "johndoe",  
  "avatar_url": "",  
  "uuid": "00000000-0000-4562-b3fc-2c963f66afa6"  
}
```

5. Potvrdite da je novi korisnik kreiran i da postoji u listi korisnika slanjem **GET** zahteva na **/users/** endpoint.
6. Proverite da je novokreirani korisnik uspešno vraćen slanjem **GET** zahteva sa korisničkim uuid-om na **/users/{uuid}/** endpoint.

Kriterijumi prihvatanja

1. Statusni kod respons-a je 200.
2. Odgovor sadrži vaše podatke i ne gubi vrednosti.
3. Korisnik je prikazan u listi korisnika.
4. Kreirani korisnik je ispravno prikazan kada se dohvati pomoću svog uuid-a.



Zadatak 4 - Nemogućnost ažuriranja korisnika sa već zauzetim podacima

Testiraj funkcionalnost koja onemogućava ažuriranje korisničkih podataka ako su podaci već zauzeti.

Test case:

1. Pošaljite **GET** zahtev na **/users/** endpoint za prikaz liste korisnika.
2. Odaberite korisnika čije podatke želite da ažurirate sa podacima nekog od postojećih korisnika. (email i nickname)
3. Ispunite payload zahtev sa prethodno uzetim email-om ili nadimkom:

Primer Payload-a:

```
{  
  "email": "string"  
}
```

4. Koristite prethodno uzeti uuid drugog korisnika kao parametar u **/users/{uuid}/** endpoint-u.
5. Pošaljite **PATCH** zahtev sa prethodno pripremljenim payload-om na endpoint iz prethodnog koraka.
6. Osigurajte da odgovor sa statusom 409 bude primljen i da poruka "User email/nickname already taken" bude prikazana.

Primer respons-a:

```
{  
  "code": 409,  
  "message": "User email already taken: alex@gmail.com"  
}
```

Kriterijumi prihvatanja

1. Statusni kod respons-a je 409.
2. U odgovoru se nalazi poruka "already taken" (već zauzeto).



Zadatak 5 – Ažuriranje korisničke liste želja

Napiši test scenario koji proverava mogućnost dodavanja stavki na korisničku listu želja.

Test case:

1. Pošaljite **GET** zahtev na **/users/** endpoint za prikaz liste korisnika.
2. Odaberite bilo kog korisnika u čiju wishlist-u želite da dodate pesmu i uzmite njegov uuid.
3. Pošaljite **GET** zahtev na **/songs/** endpoint za prikaz liste svih pesama.
4. Odaberite pesmu koju želite da dodate i uzmite njen uuid.
5. Ispunite polje item_uuid sa prethodno uzetim uuid-om pesme u payload zahtevu.

Primer Payload-a:

```
{  
  "item_uuid": "string"  
}
```

6. Koristite prethodno uzeti uuid korisnika kao parametar u **/users/{uuid}/wishlist/add/** endpoint-u.
7. Pošaljite **POST** zahtev sa prethodno pripremljenim payload-om na endpoint iz prethodnog koraka.
8. Proverite da je statusni kod odgovora **200**.
9. Potvrdite da je odabrana pesma uspešno dodata na wishlist izabranog korisnika slanjem **GET** zahteva na **/users/{uuid}/wishlist/** endpoint.

Kriterijumi prihvatanja

1. Statusni kod odgovora nakon što je pesma dodata na wishlist-u je 200.
2. Odabrana pesma je prikazana na wishlist-i izabranog korisnika.