Univerzitet u Beogradu

Fakultet organizacionih nauka

Katedra za elektronsko poslovanje

Laravel

Domaći zadatak 1

|  |  |  |
| --- | --- | --- |
| Ime | Prezime | Broj indeksa |
| Janko | Kovačević | 141/2018 |
| Link ka Github-u | https://github.com/jkovvv/iteh-laravel-quiz | |

Sadržaj

[1 Korisnički zahtev 3](#_Toc178852707)

[2 Implementacija 4](#_Toc178852708)

[3 REST API 5](#_Toc178852709)

[4 Korisničko uputstvo 12](#_Toc178852710)

# Korisnički zahtev

Aplikacija predstavlja platformu za kvizove, omogućavajući korisnicima da se registruju, prijave i učestvuju u kvizovima. Korisnici mogu birati između dostupnih kvizova, odgovarati na pitanja i videti svoje rezultate nakon završetka. Pored toga, aplikacija omogućava kreiranje kvizova – korisnici mogu dodavati pitanja i definisati tačne odgovore. Takođe, aplikacija beleži svaki pokušaj rešavanja kviza, omogućavajući praćenje napretka korisnika.

**Slučajevi korišćenja**

* **Registracija korisnika:** Korisnik može kreirati nalog putem API zahteva /register tako što šalje podatke o korisničkom imenu, email adresi i lozinci.
* **Login korisnika:** Korisnik se može ulogovati putem API zahteva /login tako što šalje podatke o email adresi i lozinci.
* **Logout korisnika:** Korisnik se može izlogovati putem API zahteva /logout
* **Kreiranje novog kviza:** Korisnik može kreirati kviz putem API zahteva /api/create-quiz tako što šalje podatke o naslovu i opisu kviza.
* **Prikaz kviza:** Korisnik može dobiti informacije o određenom kvizu putem API zahteva /show-quiz/{id} gde šalje ID kviza koji želi da prikaže. Kao odgovor dobija podatke o kvizu.
* **Izmena kviza:** Korisnik može ažurirati informacije o kvizu putem API zahteva /update-quiz/{id}, gde šalje novi naslov ili opis kviza zajedno sa ID-om kviza koji želi da izmeni.
* **Brisanje kviza:** Korisnik može obrisati kviz putem API zahteva DELETE /api/quiz/{id}, gde šalje ID kviza koji želi da obriše.
* **Kreiranje novog pitanja:** Korisnik može kreirati pitanje putem API zahteva /api/create-question tako što šalje podatke o tekstu pitanja, tačnom odgovoru, pogrešnim odgovorima i ID-u kviza kojem pitanje pripada.
* **Prikaz pitanja:** Korisnik može dobiti informacije o određenom pitanju putem API zahteva /show-question/{id} gde šalje ID pitanja koje želi da prikaže. Kao odgovor dobija tekst pitanja i odgovore.
* **Izmena pitanja:** Korisnik može izmeniti podatke o pitanju putem API zahteva /edit-question/{id} gde šalje ID pitanja koje želi da izmeni, zajedno sa novim tekstom pitanja ili odgovorima.
* **Brisanje pitanja:** Korisnik može obrisati pitanje putem API zahteva /delete-question/{id}, gde šalje ID pitanja koje želi da obriše.
* **Kreiranje novog ispunjavanja kviza:** Kreiranje ispunjavanja pitanja putem API zahteva /api/create-quiz-attempt uz podatke o ID-u korisnika koji je ispunio kviz, ID-u kviza koji je ispunjen i rezultatu kviza.
* **Prikaz poslednjeg ispunjavanja kviza:** Korisnik može videti poslednje ispunjavanje kviza putem API zahteva /show-last-quiz-attempt/{quiz\_id}/{user\_id}, gde šalje ID kviza i ID korisnika.
* **Prikaz svih ispunjavanja kviza:** Korisnik može dobiti listu svih ispunjavanja kviza putem API zahteva /show-all-quiz-attempts/{quiz\_id}/{user\_id}, gde šalje ID kviza i ID korisnika, kako bi video sve pokušaje rešavanja tog kviza.
* **Izmena ispunjavanja kviza:** Korisnik može ažurirati podatke o ispunjavanju kviza putem API zahteva /update-quiz-attempt/{id}, gde šalje ID ispunjavanja kviza i nove podatke o rezultatu.
* **Brisanje ispunjavanja kviza:** Korisnik može obrisati ispunjavanje kviza putem API zahteva /delete-quiz-attempt/{id}, gde šalje ID ispunjavanja kviza koje želi da obriše.

# Implementacija

**Modeli**

* **User:** Model za rad sa korisničkim podacima (ime, email, lozinka). Povezan je sa modelom QuizAttempt.
* **Quiz:** Model koji predstavlja kviz, sadrži informacije o naslovu i opisu kviza. Povezan je sa modelima Question I QuizAttempt.
* **Question:** Model koji predstavlja jedno pitanje u kvizu, sadrži informacije o tekstu pitanja, tačnom i netačnim odgovorima kao i ID-u kviza kojem pripada. Povezan je sa modelom Question.
* **QuizAttempt:** Model koji predstavlja jedno ispunjavanje kviza, sadrži informacije o ID-u korisnika koji je ispunio kviz, ID-u kviza koji je ispunjen kao i postignutom rezultatu tog ispunjavanja. Povezan je sa modelima Quiz i User.

**Migracije**

* **create\_users\_table:** Migracija koja kreira tabelu za čuvanje korisnika.
* **create\_questions\_table:** Migracija koja kreira tabelu za čuvanje pitanja.
* **create\_quizzes\_table:** Migracija koja kreira tabelu za čuvanje kvizova.
* **create\_quiz\_attempts\_table:** Migracija koja kreira tabelu za čuvanje ispunjavanja kvizova.

**Kontroleri**

* **AuthController:** Kontroler za upravljanje logovanjem i autentifikacijom korisnika. Sadrži funkcije za registrovanje, login i logout korisnika.
* **QuestionController:** Kontroler za upravljanje pitanjma. Sadrži funkcije za kreiranje, prikazivanje, ažuriranje I brisanje pitanja.
* **QuizController:** Kontroler za upravljanje kvizovima. Sadrži funkcije za kreiranje, prikazivanje, ažuriranje I brisanje kvizova.
* **QuizAttemptController:** Kontroler za upravljanje ispunjavanjima kvizova. Sadrži funkcije za kreiranje, prikazivanje, ažuriranje i brisanje ispunjavanja kvizova.

**API rute**

|  |  |  |
| --- | --- | --- |
| **Metod** | **Ruta** | **Opis** |
| POST | /register | Registrovanje novog korisnika |
| POST | /login | Login korisnika |
| POST | /logout | Logout korisnika |
| POST | /api/create-quiz | Kreiranje novog kviza |
| GET | /api/show-quiz/{id} | Prikaz kviza |
| PATCH | /api/update-quiz/{id} | Izmena kviza |
| DELETE | /api/delete-quiz/{id} | Brisanje kviza |
| POST | /api/store-question | Kreiranje novog pitanja |
| GET | /api/show-question/{id} | Prikaz pitanja |
| PATCH | /api/edit-question/{id} | Izmena pitanja |
| DELETE | /api/delete-question/{id} | Brisanje pitanja |
| POST | /api/create-quiz-attempt | Kreiranje novog ispunjavanja kviza |
| GET | /api/show-last-quiz-attempt/{quiz\_id}/{user\_id} | Prikaz poslednjeg ispunjavanja kviza |
| GET | /api/show-all-quiz-attempts/{quiz\_id}/{user\_id} | Prikaz svih ispunjavanja kviza |
| PATCH | /api/update-quiz-attempt/{id} | Izmena ispunjavanja kviza |
| DELETE | /api/delete-quiz-attempt/{id} | Brisanje ispunjavanja kviza |

# REST API

|  |  |
| --- | --- |
| Opis funkcije | Registrovanje novog korisnika |
| HTTP metoda | POST |
| URL | /register |
| URL parametri | (nema) |
| HTTP body parametri | {    "name": "Janko Kovacevic",    "email": "janko@mejl.com",    "password": "sifra123"  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "data": {          "name": "Janko Kovacevic",          "email": "janko@mejl.com",          "updated\_at": "2024-10-03T09:17:27.000000Z",          "created\_at": "2024-10-03T09:17:27.000000Z",          "id": 5      },      "access\_token": "15|RPX5jMvXiGMdA94plyVWMaIaoGa3fRRj6KyIO4GE",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Login novog korisnika |
| HTTP metoda | POST |
| URL | /login |
| URL parametri | (nema) |
| HTTP body parametri | {    "email": "janko@mejl.com",    "password": "sifra123"  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "data": {          "id": 5,          "name": "Janko Kovacevic",          "email": "janko@mejl.com",          "email\_verified\_at": **null**,          "created\_at": "2024-10-03T09:17:27.000000Z",          "updated\_at": "2024-10-03T09:17:27.000000Z"      },      "access\_token": "16|W3l1a6fdYqBENolyDi6DrZuaJka1pCBDjEpNILdQ",      "token\_type": "Bearer"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Logout novog korisnika |
| HTTP metoda | POST |
| URL | /logout |
| URL parametri | (nema) |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Successfully logged out!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Kreiranje novog kviza |
| HTTP metoda | POST |
| URL | /api/create-quiz |
| URL parametri | (nema) |
| HTTP body parametri | {      "title" : "Geografski kviz",      "description" : "Kviz koji testira vase znanje iz geografije!"  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Quiz created successfully!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz kviza |
| HTTP metoda | GET |
| URL | /api/show-quiz |
| URL parametri | ?quiz\_id=5 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "id": 5,      "created\_at": "2024-10-03T09:21:24.000000Z",      "updated\_at": "2024-10-03T09:21:24.000000Z",      "title": "Geografski kviz",      "description": "Kviz koji testira vase znanje iz geografije!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Izmena kviza |
| HTTP metoda | PATCH |
| URL | /api/update-quiz |
| URL parametri | ?quiz\_id=5 |
| HTTP body parametri | {      "description" : "Ovo je kviz iz geografije."  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Quiz updated successfully!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje kviza |
| HTTP metoda | DELETE |
| URL | /api/delete-quiz |
| URL parametri | ?quiz\_id=5 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Quiz deleted successfully!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Kreiranje novog pitanja |
| HTTP metoda | POST |
| URL | /api/store-question |
| URL parametri | (nema) |
| HTTP body parametri | {    "question\_text": "Koji je najviši planinski vrh na svetu?",    "right": "Mont Everest",    "wrong1": "Ben Nevis",    "wrong2": "K2",    "wrong3": "Crni vrh",    "quiz\_id": 6  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Question stored successfully!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz pitanja |
| HTTP metoda | GET |
| URL | /api/show-question |
| URL parametri | ?question\_id=4 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "id": 4,      "created\_at": "2024-10-03T09:31:13.000000Z",      "updated\_at": "2024-10-03T09:31:13.000000Z",      "question\_text": "Koji je najviši planinski vrh na svetu?",      "right": "Mont Everest",      "wrong1": "Ben Nevis",      "wrong2": "K2",      "wrong3": "Crni vrh",      "quiz\_id": 6  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Izmena pitanja |
| HTTP metoda | PATCH |
| URL | /api/edit-question |
| URL parametri | ?question\_id=4 |
| HTTP body parametri | {      "wrong3": "Pančićev vrh"  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Question updated successfully!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje pitanja |
| HTTP metoda | DELETE |
| URL | /api/delete-question |
| URL parametri | ?question\_id=4 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Question deleted successfully!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Kreiranje novog ispunjavanja kviza |
| HTTP metoda | POST |
| URL | /api/create-quiz-attempt |
| URL parametri | (nema) |
| HTTP body parametri | {      "user\_id" : 5,      "quiz\_id" : 6,      "score" : 90  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Quiz attempt created successfully!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz poslednjeg ispunjavanja kviza |
| HTTP metoda | GET |
| URL | /api/show-last-quiz-attempt |
| URL parametri | ?quiz\_id=6&user\_id=5 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "id": 8,      "user\_id": 5,      "quiz\_id": 6,      "score": 93,      "attempted\_at": "2024-10-03 09:39:59",      "created\_at": "2024-10-03T09:39:59.000000Z",      "updated\_at": "2024-10-03T09:39:59.000000Z"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Prikaz svih ispunjavanja kviza |
| HTTP metoda | GET |
| URL | /api/show-all-quiz-attempts |
| URL parametri | ?quiz\_id=6&user\_id=5 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | [      {          "id": 7,          "user\_id": 5,          "quiz\_id": 6,          "score": 90,          "attempted\_at": "2024-10-03 09:38:52",          "created\_at": "2024-10-03T09:38:52.000000Z",          "updated\_at": "2024-10-03T09:38:52.000000Z"      },      {          "id": 8,          "user\_id": 5,          "quiz\_id": 6,          "score": 93,          "attempted\_at": "2024-10-03 09:39:59",          "created\_at": "2024-10-03T09:39:59.000000Z",          "updated\_at": "2024-10-03T09:39:59.000000Z"      }  ] |
| Format izlaznih parametara | application/json |

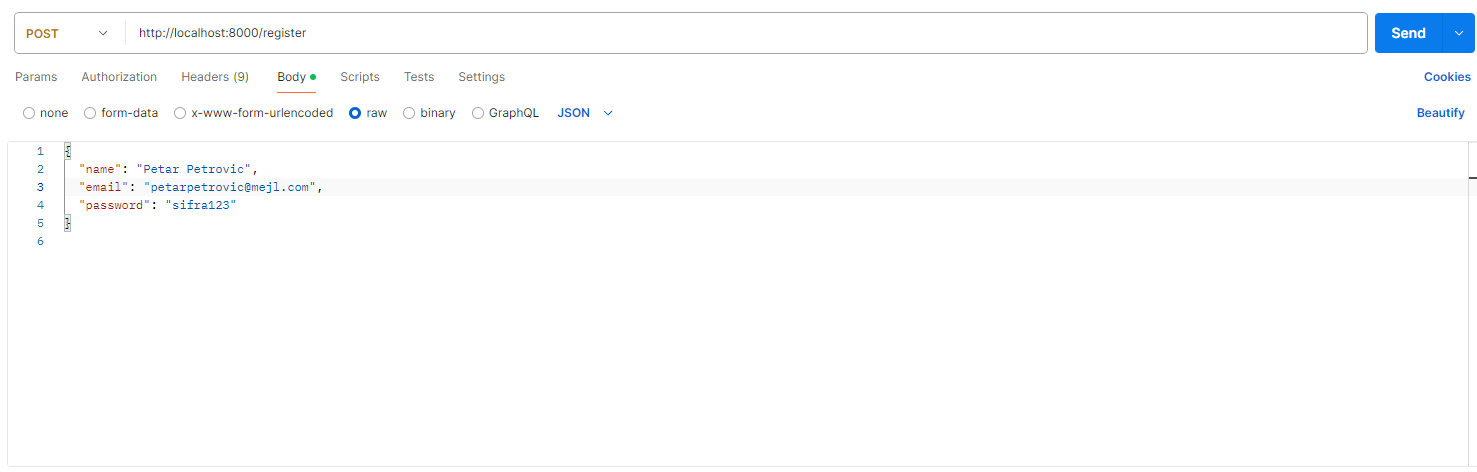
|  |  |
| --- | --- |
| Opis funkcije | Izmena ispunjavanja kviza |
| HTTP metoda | PATCH |
| URL | /api/update-quiz-attempt |
| URL parametri | ?quiz\_attempt\_id=8 |
| HTTP body parametri | {      "user\_id" : 5,      "quiz\_id" : 6,      "score" : 60  } |
| Format HTTP body parametara | JSON |
| Izlazni parametri | {      "message": "Quiz attempt updated successfully!"  } |
| Format izlaznih parametara | application/json |

|  |  |
| --- | --- |
| Opis funkcije | Brisanje ispunjavanja kviza |
| HTTP metoda | DELETE |
| URL | /api/delete-quiz-attempt |
| URL parametri | ?quiz\_attempt\_id=8 |
| HTTP body parametri | (nema) |
| Format HTTP body parametara | (nema) |
| Izlazni parametri | {      "message": "Quiz attempt deleted successfully!"  } |
| Format izlaznih parametara | application/json |

# Korisničko uputstvo

**Autentifikacija korsnika**

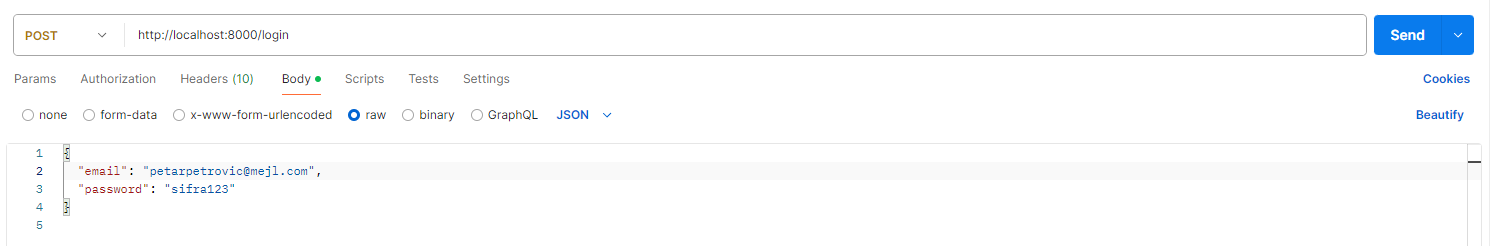
* **Registrovanje:** Kada se korisnik registruje, potrebno je poslati POST zahtev sa potrebnim poljima (name, email, password).



Ukoliko je zahtev ispravan, kao povratnu informaciju dobijamo unete podatke o korisniku, uz timestamp registrovanja korsnika. Uz to dobijamo token koji je potreban za autentifikaciju pri svim zahtevima u programu, osim zahteva o prikazu (dakle za sve zahteve kreiranja, izmena i brisanja).



* **Login:** Kada korisnik želi da se uloguje, potrebno je poslati POST zahtev sa potrebnim poljima (email, password).



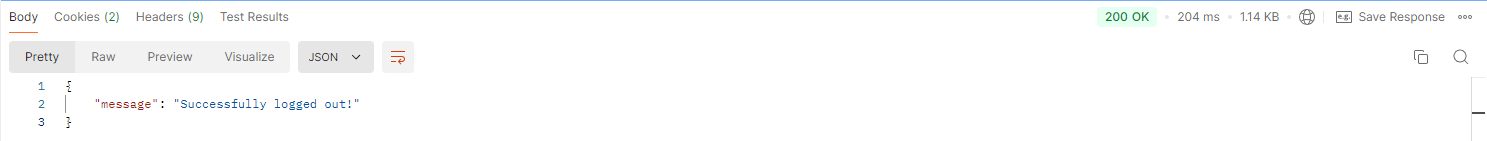
Ukoliko je zahtev ispravan, kao povratnu informaciju dobijamo podatke o korisniku, uz timestamp registrovanja korsnika. Uz to dobijamo i token koji je potreban za autentifikaciju.



* **Logout:** Kada korisnik želi da se uloguje, potrebno je poslati POST zahtev. Ovaj zahtev nema body, jer ne postoje informacije koje je potrebno proslediti.



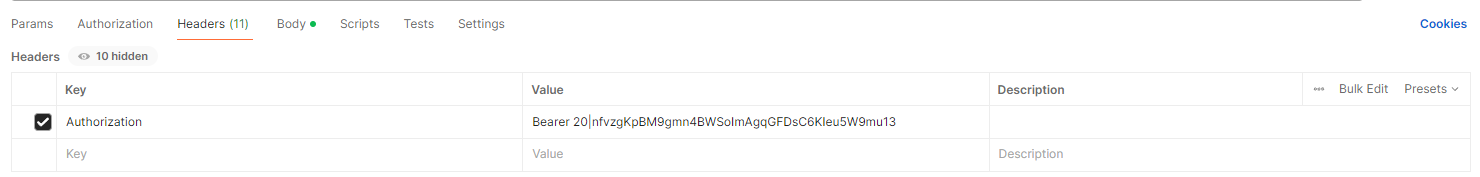
Ukoliko je zahtev ispravan, kao povratnu informaciju dobijamo poruku o uspešnom izlogovanju.



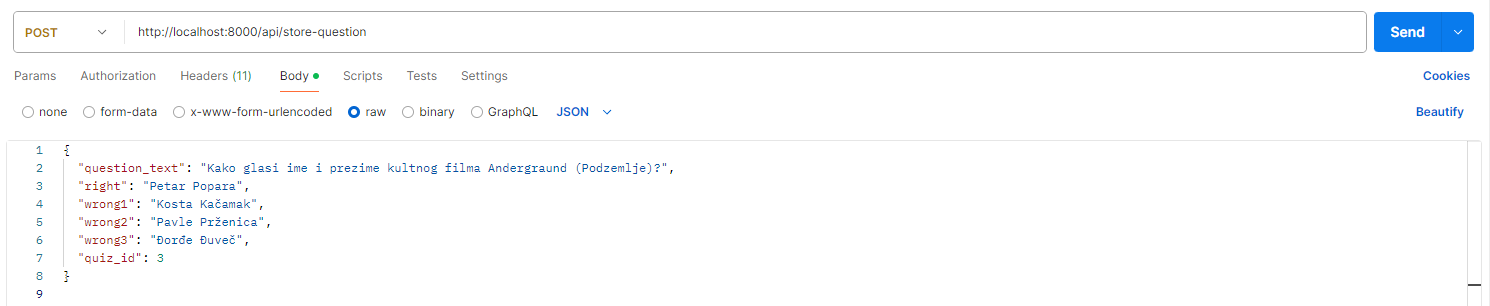
**Kreiranje API zahteva**

Kako se svi API zahtevi u programu izvršavaju na isti način, prikazaćemo kreiranje API ruta na jednom primeru. Neka to bude vezano za model Question.

Pre svega. Za sve zahteve osim prikaza (kreiranje, izmena i brisanje), potrebno je autentifikovati korisnika. Da li je korisnik ulogovan proveravamo putem tokena dodeljenog u header-u samog zahteva. Header sa adekvatno dodeljenim tokenom treba izgledati na sledeći način:



* **Kreiranje pitanja:** Kada korisnik kreira pitanje, potrebno je poslati POST zahtev sa potrebnim poljima (tekst pianja, tačan odgovor, netačni odgovori i ID kviza kojem pripada). Uz to, potreban je i token za autentifikaciju u header delu zahteva.



Ukoliko je zahtev ispravan, kao povratnu informaciju dobijamo poruku o uspešno kreiranom pitanju.



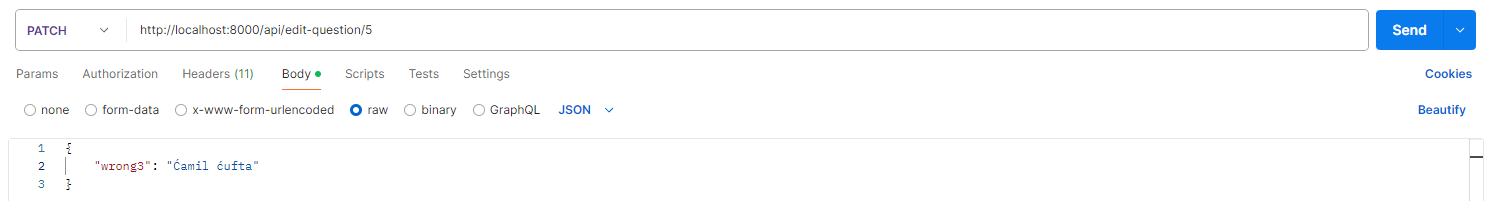
* **Prikaz pitanja:** Kada korisnik želi prikazati određeno pitanje, potrebno je poslati GET zahtev. Ovaj zahtev nema body. Nije potrebna autentifikacija.



Ukoliko je zahtev ispravan, kao povratnu informaciju dobijamo podatke o pitanju koje je trebalo prikazati.



* **Izmena pitanja:** Kada korisnik unosi izmene u pitanje, potrebno je poslati PATCH zahtev, kao i nove informacije koje je potrebno izmeniti. Uz to, potreban je i token za autentifikaciju u header delu zahteva.



Ukoliko je zahtev ispravan, kao povratnu informaciju dobijamo poruku da su podaci uspešno izmenjeni.



* **Brisanje pitanja:** Korisnik briše postojeće pitanje, potrebno je poslati DELETE zahtev. Ovaj zahtev nema body. Potrebno je uneti token za autentifikaciju.



Ukoliko je zahtev ispravan, kao povratnu informaciju dobijamo poruku da je pitanje uspešno obrisano.

