**QuizMakerAPI – Korisnička dokumentacija**

Važna napomena – datumi su u UTC vremenu pa će ih se na frontendu trebati prebaciti u željenu vremensku zonu

**Quiz**

* A screenshot of a computer

  Description automatically generatedPOST: api/quiz  
  Način poziva: localhost:{port}/api/quiz  
  Body: u name treba ići ime kviza, a u questions lista pitanja sa text i answer  
  Slika:  
  A screenshot of a computer

  Description automatically generated  
  Vraća natrag kreirani kviz sa njegovim pitanjima i tagovima(kasnije se dodavaju)  
  Slika:
* GET: api/quiz/{id}  
  Način poziva: localhost:{port}/api/quiz/1  
  Vraća kviz i njegova pitanja i tagove.  
  Slika:  
  A screenshot of a computer

  Description automatically generated
* A screenshot of a computer

  Description automatically generated// PUT: api/quiz/{id}  
  Način poziva: localhost:{port}/api/quiz/1  
  Mijenja samo naziv kviza, a pitanja dodajemo, mičemo i mijenjamo sa drugim endpointima  
  Slika:

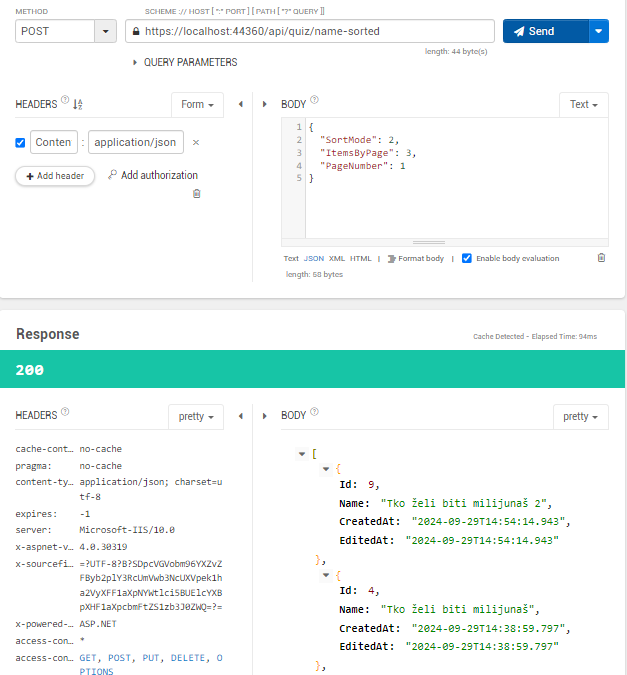
* // POST: api/quiz/addQuestion  
  Način poziva: localhost:{port}/api/quiz/addQuestion  
  Body: prima QuizId i QuestionId kao brojeve i veže pitanje na kviz ako nije vezano  
  Slika:  
    
  A screenshot of a computer

  Description automatically generated
* POST: api/quiz/removeQuestion  
  Isto kao addQuestion ali ova ruta miče vezu
* DELETE: api/quiz/{id}  
  Način poziva: localhost:{port}/api/quiz/3  
  Kvizu postavlja deleted polje na trenutno vrijeme i više neće biti vidljivo  
  Slika:  
  A screenshot of a computer

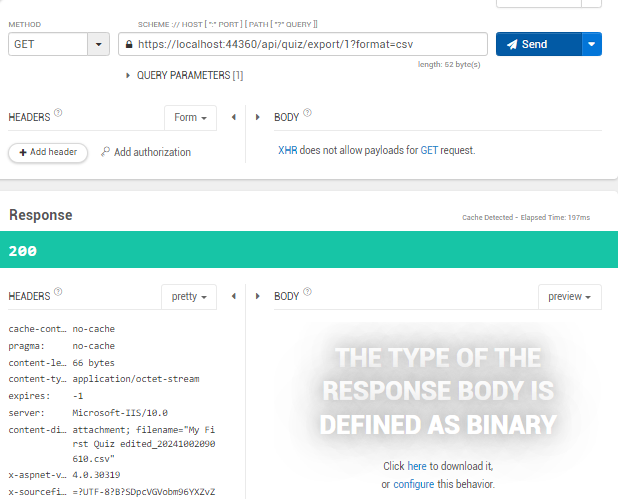
  Description automatically generated
* POST: api/quiz/{id}/revive  
  Način poziva: localhost:{port}/ api/quiz/3/revive  
  Kvizu miče deleted polje i opet postaje vidljivo
* POST: api/quiz/{id}/addTags  
  Način poziva: localhost:{port}/api/quiz/3/addTags  
  U Body mu se šalje string Tags u kojem su tagovi odvojeni sa „;“ (točka zarez)  
  Slika:  
  A screenshot of a computer

  Description automatically generated

* POST: api/quiz/removeTag  
  Način poziva: localhost:{port}/api/quiz/removeTag  
  U Body se šalju QuizId i TagId i onda se miče ta veza ako postoji  
  Slika:  
  A screenshot of a computer

  Description automatically generated
* POST: api/quiz/name-sorted  
  Način poziva: localhost:{port}/api/quiz/name-sorted  
  U Body se šalje sorted (1 za ascending sort, 2 za descending), ItemsByPage (koliko kvizova će se dohvatiti) i pageNumber (na kojoj smo stranici)  
  Vraća se lista kvizova ako ih ima na toj stranici.  
  Slika:  
  
* POST: api/quiz/modifiedSorted  
  Isto kao i name-sorted – jedina razlika je što je ovo sortirano po tome koji je zadnje promjenjen/napravljen (pri kreiranju se popuni i editedAt polje po kojem se sortira)
* POST: api/quiz/byTags  
  Način poziva : localhost:{port}/api/quiz/byTags  
  Sličan je kao prethodna 2 ali umjesto sortMode ima Tags u koji se trebaju napisati tagovi odvojeni sa ;  
  Slika:  
  A screenshot of a computer

  Description automatically generated
* GET: api/quiz/export/formats  
  Način poziva: localhost:{port}/api/quiz/export/formats  
  Vraća listu svih raspoloživih formata za export  
  Slika:  
  A screenshot of a computer

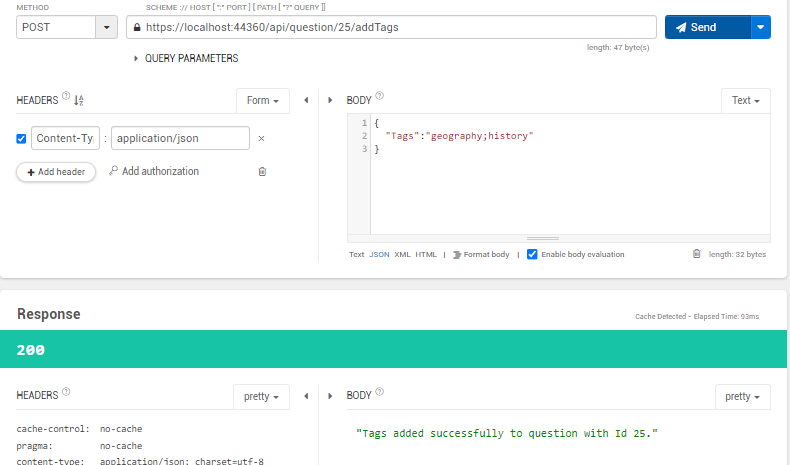
  Description automatically generated
* GET: api/quiz/export/{quizId}?format={format}  
  Način poziva: localhost:{port}/api/quiz/export/1?format=csv  
  Vraća kviz u csv datoteci koja se može downloadati  
  Slika:  
  

**Question**

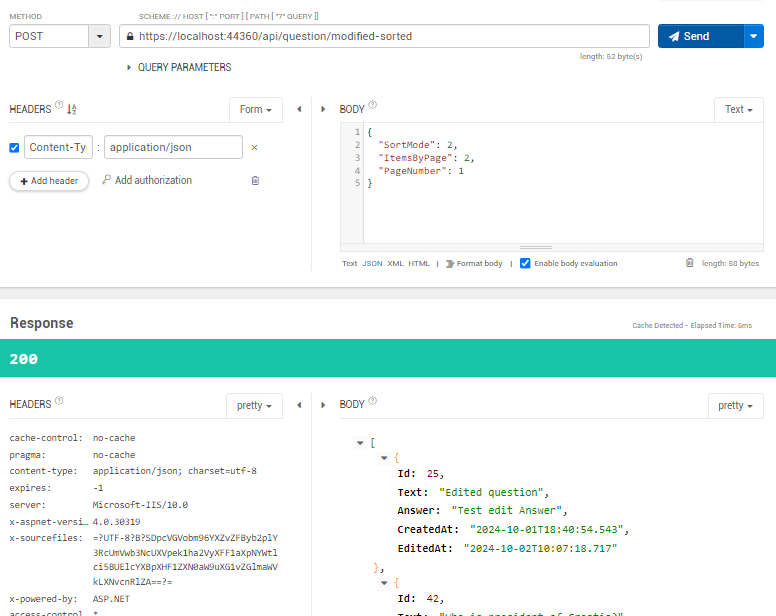
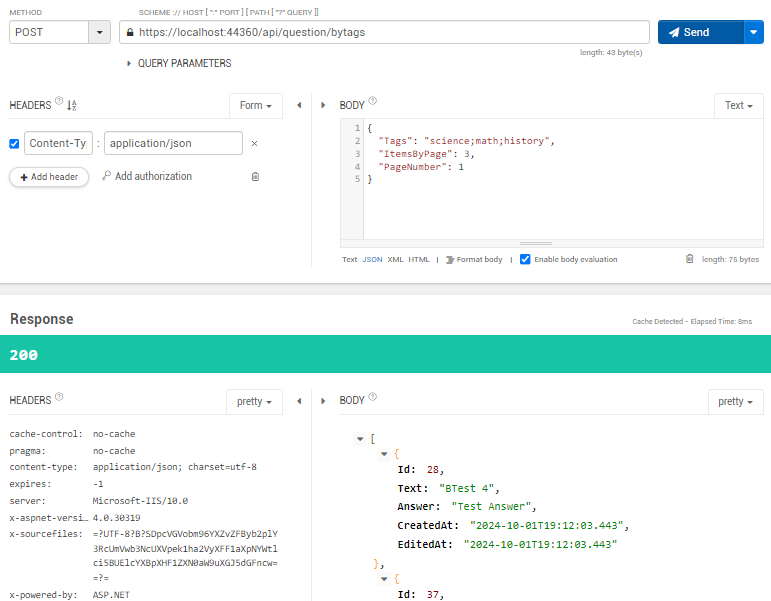
* POST: api/question  
  Način poziva: localhost:{port}/api/question  
  U Body se šalje Text, Answer i QuizId (opcionalno, ako želimo to pitanje odmah pridružiti nekom kvizu).  
  Vraća napravljeno pitanje sa listom kvizova gdje se nalazi i listom tagova (dodajemo ih kasnije)  
  Slika:  
  A screenshot of a computer

  Description automatically generated
* GET: api/question/{id}  
  Način poziva: localhost:{port}/api/question/25  
  Vraća pitanje sa povezanim kvizovima i tagovima  
  Slika:  
  A screenshot of a computer

  Description automatically generated
* DELETE: api/question/{id}  
  Način poziva: localhost:{port}/api/question/25  
  Postavlja trenutno vrijeme u deletedAt i više to pitanje neće biti vidljivo.  
  Ograničenje: Ne mogu se brisati pitanja koja su povezana sa nekim kvizom. Važno je napomenuti da se ne mogu brisati ni ona pitanja koja su povezana sa obrisanim kvizom. Ako se želi obrisati takvo pitanje trebalo bi oživjeti kviz i maknuti pitanje sa njega.
* POST: api/question/{id}/revive  
  Način poziva: localhost:{port}/ api/question/25/revive  
  Miče vrijeme iz deletedAt i pitanje opet postaje vidljivo
* PUT: api/question/{id}  
  Način poziva: localhost:{port}/api/question/25  
  U Body se šalje Text i Answer koji se onda mijenjaju u pitanju  
  Ograničenje: Ne mogu se mijenjati pitanja koja su vezana na više od jednog kviza. Ako je vezano samo na jedan kviz ili nije vezano ni sa jednim kvizom onda je promjena moguća.  
  Slika:  
  A screenshot of a computer

  Description automatically generated
* POST: api/question/{id}/addTags  
  Način poziva: localhost:{port}/api/question/25/addTags  
  U Body se šalje Tags koji je string u kojem su tagovi odvojeni sa ;  
  Ti tagovi se onda spajaju sa pitanjem  
  Slika:  
  
* POST: api/question/removeTag  
  Način poziva: localhost:{port}/ api/question/removeTag  
  U Body se šalje QuestionId i TagId i ako veza postoji miče ju  
  Slika:  
  A screenshot of a computer

  Description automatically generated
* POST: api/question/text-search  
  Način poziva; localhost:{port}/ api/question/text-search  
  U Body se šalje SearchText – to je podstring koji će tražiti u tekstu pitanja, ItemsByPage - broj koji kaže koliko pitanja će dohvatiti i PageNumber – broj koji kaže na kojoj smo stranici.  
  Dohvaća se onoliko pitanja koliko je u itemsByPage ako ih ima na danoj stranici.  
  Slika:  
  A screenshot of a computer

  Description automatically generated
* POST: api/question/modified-sorted  
  Način poziva: localhost:{port}/ api/question/modified-sorted  
  U Body se šalje sortMode – 1 je za ascending sort, a 2 za descending sort i isto kao u prethodnom šalju se itemsByPage i PageNumber  
  Vraća pitanja sortiranja po vremenu promjene/kreiranja – pri kreiranju se popunjava i EditedAt polje  
  Slika:  
  
* POST: api/question/byTags  
  Način poziva: localhost:{port}/api/question/byTags  
  U Body se šalju Tags – string u kojem su tagovi odvojeni sa ; i onda se dohvaćaju ona pitanja koja imaju neke od tih tagova, isto imamo ItemsByPage i PageNumber koji rade isto kao na prethodnima  
  Slika:  
  

**Zaštita**

Aplikacija je postavljena na Azure server i tamo je zaštićena sa JWT zaštitom. Da bi se moglo pristupati rutama apija prvo je potrebna prijava u api preko api/auth/login rute u kojoj će se u body-u slati username i password. Username i password će biti dostavljeni u secret.txt datoteci. Link na online api će biti poslan u emailu.

Kada se na api/auth/login pošalju točan username i password api će vratiti token. Taj token će se trebati koristiti za sve ostale pozive tako da se napravi Header Authorization i u njega se stavi Bearer + Token koji se dobio sa prijavom.

Slike:

Bez headera sa autorizacijom:

A screenshot of a computer

Description automatically generated

Dohvaćanje tokena:

A screenshot of a computer

Description automatically generated

Ispravan poziv sa Authorization headerom

A screenshot of a computer

Description automatically generated

**CORS**

Dopušten je CORS da se api može isprobati iz frontenda. To nije dobra praksa za produkciju ali je praktično za testiranje. Bez toga se ne bi mogli raditi pozivi iz React-a (osim ako točno znamo na kojem je serveru pa da se otvori pristup sa te domene).

Slika:

A screenshot of a computer

Description automatically generated