

Test

Aplikacija Coetus

Potrebno je realizovati front-end prikaz za aplikaciju Coetus. Funkcionalnosti:

1. Aplikacija, kada se ucita, prikazuje sve poruke, datum, i autora poruke.
2. Ispod prikaza svih poruka postoje dva text polja i jedno dugme za unos nove poruke.
 - Prilikom klika na dugme, potrebno je registrovati novu poruku na server(back-endu), a zatim apdejtovati sadrzaj liste poruka. Username za poruku je podatak iz prvog tekst polja. Tekst poruke je podatak iz drugog tekst polja
3. Sta radi Fetch?
4. Koja je razlika izmedju GET-a, POST-a?
5. Pomocu koje reči exportujemo funkciju/klasu unutar fajla?
6. Sta je to React?

Napomena

Realizovati aplikaciju preko modula.

Dodatno:

- Potrebno je updejtovati prikaz svih poruka periodicno svakih 10 sekundi. Pogledati [JavaScript tajmere](#)
- Napraviti dizajn za aplikaciju - +100% poena

Dokumentacija API-a

Link aplikacije: <https://coetus.herokuapp.com>

LINK METODE MESSAGE: <https://coetus.herokuapp.com/api/message>

GET - Dohvata sve poruke Primer odgovora:

```
1 {
2   "success": true,
3   "data": [
4     {
5       "id": 0,
6       "username": "dule",
7       "message": "prva poruka",
8       "timestamp": 1575235584388
9     },
10    {
11      "id": 1,
12      "username": "dule",
13      "message": "prva poruka",
14      "timestamp": 1572235584388
15    }
16  ]
17 }
```

POST - Dohvata sve poruke za poslatog user-a

Primer zahteva:

```
1 {
2   "username": "dule"
3 }
```

Primer odgovora:

```
1 {
2   "success": true,
3   "messages": [
4     {
5       "id": 0,
6       "username": "dule",
7       "message": "prva poruka",
8       "timestamp": 1575235584388
9     }
10  ]
11 }
```

PUT - Ubacuje poruku za poslatog user-a

Primer zahteva:

```
1 {  
2   "username": "dule",  
3   "message": "druga poruka"  
4 }
```

Primer odgovora:

```
1 {  
2   "success": true,  
3   "message": "Message added Succesfully"  
4 }
```