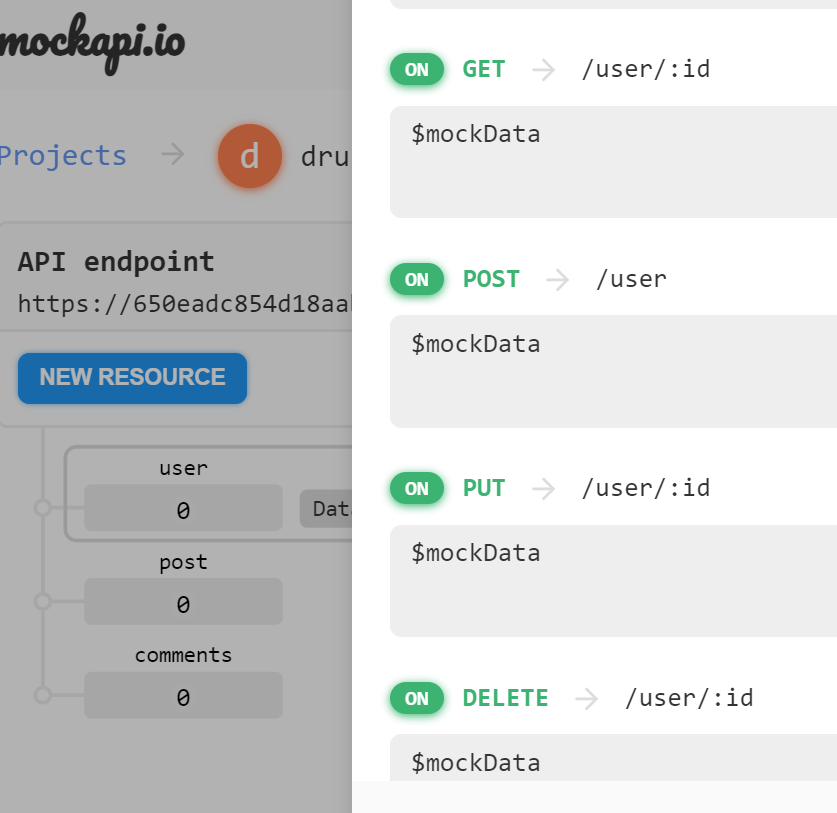
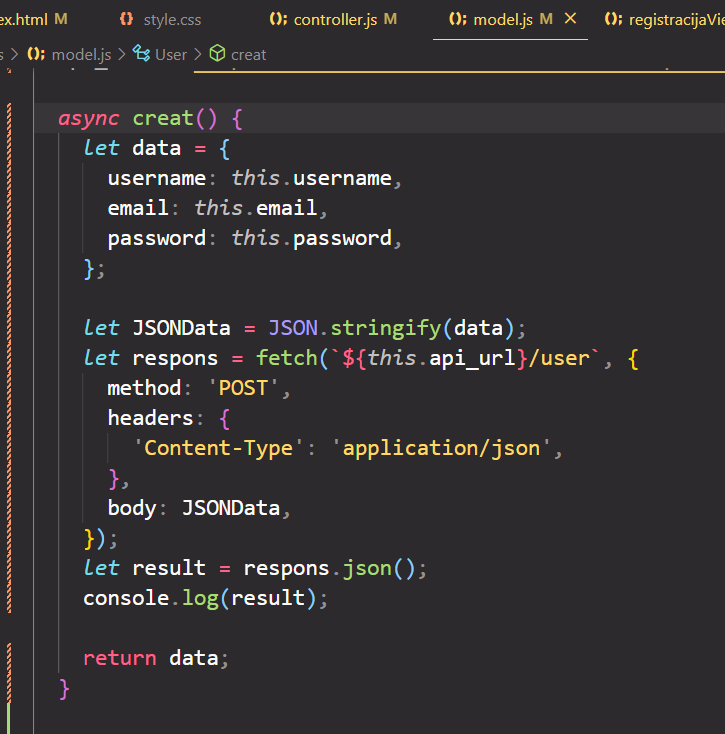
SLANJE PODATAKA NA SERVER ( MOCKAPI )

Dakle kada saljemo podatke na server koristimo POST metodu. Nas mockAPI simulira server i kada udjemo sada na neki kreirani folder na mockAPI-ju vidimo da mozemo koristiti put,get,delete i POST metode:

dakle da bismo postavili neke vrijednosti iz koda na server koristimo POST metodu.Dakle pomocu fetch() metode saljemo zahtjev serveru da zelimo da postavimo na njega neke podatke.



Ono sto ovde vidimo da smo u respons varijabli pozvali fetch() metodu.U fetch() metodi smo definisali argument URL ovog apija ${this.api\_url} te smo jos definisali da se radi o post metodi zato sto mo na ovaj URL dodali ‘/user’.

Tako smo definisali potpuni url. Sada jos moramo da definisemo i zaglavlje.To takodje radimo tako sto postavljamo drugi argument ove fetch() metode koji nam je objekat i koji prestavlja zaglavlje koje ce objasniti nasem serveru sta treba da radi.U zaglavlju definisemo method:’POST’ => da zelimo da postavimo nesto na server (ne da brisemo -> delete, ne da citamo GET, nego da postavimo POST). Sledece svojstvo ovog zaglavlja je ‘headers’ koje govori nasem serveru da mu zelimo poslati podatke u vidi JSON –a. I trece svojstvo ovog zaglavlja jest to da prije nego sto kreiramo to trece svojstvo u zaglavolju prvo moramo prije fetch-a() da kreiramo taj JSON => tj. da objekat iz data varijable parsujemo( pretvorimo ) u JSON tip podatka. Dakle ovo smo uradili u varijabli JSONData=JSON.stringify( data ) => ovo data je data objekat.

Kada smo to uradili onda kao trece svojstvo fetch() metode pozivamo u body svojstvu u fetch zaglavlju ovu varijablu JSONData.

Sada kada smo naucili koristiti se gitHubom najbolje je da zavirimo u granu js-dataApi da vidimo sav nas kod posto radimo sa MVC arhitekturom pa je malo komplikovaniji kod.

controller.js:

import Validator from './Validator.js';

import \* as model from './model.js';

import registracijaView from './View/registracijaView.js';

import User from './model.js';

*let* modelCreateUser = new User();

import 'core-js/stable';

import 'regenerator-runtime/runtime';

import { async } from 'regenerator-runtime/runtime';

export *let* validator = new Validator(model.config, '#registrationForm');

*async* *function* createUser(*formData*) {

  await modelCreateUser.creat(formData);

}

*function* init() {

  registracijaView.open();

  registracijaView.pokreniValidacijuForme(validator, createUser);

}

init();

model.js:

import { async } from 'regenerator-runtime';

import Validator from './Validator.js';

export *let* config = {

  korisnicko\_ime: {

    required: true,

    minLength: 5,

    maxLength: 40,

  },

  register\_email: {

    required: true,

    email: true,

    minLength: 5,

    maxLength: 40,

  },

  register\_lozinka: {

    required: true,

    minLength: 5,

    maxLength: 20,

    matching: 'ponovi\_lozinku',

  },

  ponovi\_lozinku: {

    required: true,

    minLength: 5,

    maxLength: 20,

    matching: 'register\_lozinka',

  },

};

export default *class* User {

  user\_id = '';

  username = '';

  email = '';

  password = '';

  api\_url = 'https://650eadc854d18aabfe995dda.mockapi.io';

*async* creat(*data*) {

    data = {

      username: data.username,

      email: data.email,

      password: data.password,

    };

    await *this*.sendDataToApi(data);

*/\* return data; \*/*

  }

*async* sendDataToApi(*data*) {

*let* JSONData = JSON.stringify(data);

*let* respons = await fetch(`${*this*.api\_url}/user`, {

      method: 'POST',

      headers: {

        'Content-Type': 'application/json',

      },

      body: JSONData,

    });

*let* result = await respons.json();

    console.log(result);

  }

}

registracijaView.js:

*class* RegistracijaView {

  \_btn = document.querySelector('#registracija');

  \_parrentEl = document.querySelector('.custom-modal');

  \_forma = document.querySelector('#registrationForm');

  open() {

*this*.\_btn.addEventListener('click', *this*.openBind.bind(*this*));

*this*.close();

  }

  openBind(*e*) {

    e.preventDefault();

*this*.\_parrentEl.style.display = 'block';

  }

  close() {

*let* x = document.querySelector('#closeModal');

    x.addEventListener('click', () => (*this*.\_parrentEl.style.display = 'none'));

  }

  gatherFormData() {

*const* formData = {

      username: document.querySelector('#korisnicko\_ime').value,

      email: document.querySelector('#email').value,

      password: document.querySelector('#lozinka').value,

    };

    return formData;

  }

  pokreniValidacijuForme(*validator*, *data*) {

    document

      .querySelector('#registrationForm')

      .addEventListener('submit', *e* => {

        e.preventDefault();

        if (validator.validationPassed()) {

*let* formData = *this*.gatherFormData();

          data(formData);

        } else {

          console.log('nije dobro nesto');

        }

      });

  }

}

export default new RegistracijaView();

css:

\* {

  margin: 0;

  padding: 0;

  box-sizing: border-box;

}

body {

  font-family: sans-serif;

  padding: 50px;

  background-image: url('../img/bg.jpg');

  color: white;

  background-size: cover;

  box-shadow: rgba(0, 0, 0, 0.8) 0 0 0 1000px inset;

}

button {

  cursor: pointer;

}

h1 {

  text-align: center;

  margin-bottom: 100px;

}

h2 {

  margin-bottom: 15px;

}

*.logo* {

  text-align: center;

  margin: 50px;

}

*.main-wrapper* {

  max-width: 650px;

  margin: auto;

  background-color: rgba(255, 255, 255, 0.3);

  padding: 50px;

}

form div {

  padding: 15px 0;

}

form div label {

  display: block;

  font-weight: bold;

  font-size: 16px;

}

form div input {

  height: 50px;

  width: 100%;

  font-size: 18px;

  padding: 0 10px;

}

form div button {

  cursor: pointer;

  padding: 10px 30px;

  font-size: 18px;

}

form ul {

  list-style: none;

}

form ul li {

  color: red;

}

*.no-acc-register* {

  text-align-last: center;

  margin-top: 50px;

}

*.no-acc-register* button {

  padding: 15px;

  font-size: 18px;

}

*.no-acc-register* p {

  font-size: 22px;

  margin-bottom: 15px;

}

*.custom-modal* {

  display: none;

  background-color: black;

  box-shadow: rgba(255, 255, 255, 0.8) 0 0 1000px 1000px;

  max-width: 640px;

  margin: auto;

  padding: 50px;

  position: absolute;

  top: 10%;

  left: 0;

  right: 0;

  width: 100%;

}

*.custom-modal* button*#closeModal* {

  height: 20px;

  width: 20px;

  color: white;

  background-color: transparent;

  border: 0;

  font-size: 25px;

  font-weight: bold;

  position: absolute;

  right: 30px;

  top: 30px;

}

html:

<html *lang*="en">

  <head>

    <meta *charset*="UTF-8" />

    <meta *http-equiv*="X-UA-Compatible" *content*="IE=edge" />

    <meta *name*="viewport" *content*="width=device-width, initial-scale=1.0" />

    <title>drustvena mreza</title>

    <link *rel*="stylesheet" *href*="src/css/style.css" />

    <script *src*="src/js/controller.js" *type*="module"></script>

  </head>

  <body>

    <div *class*="logo">

      <img *src*="src/img/logo.png" *alt*="logo" />

    </div>

    <h1>drustvena mreza</h1>

    <div *class*="main-wrapper">

      <form *id*="loginForm">

        <h2>lgoin</h2>

        <div>

          <label>Email:</label>

          <input

*type*="email"

*name*="email"

*id*="login\_email"

*placeholder*="niko.nikolic@gmail.com"

          />

        </div>

        <div>

          <label>Lozinka:</label>

          <input

*type*="password"

*name*="lozinka"

*id*="login\_lozinka"

*placeholder*="\*\*\*\*\*\*"

          />

        </div>

        <div>

          <button>uloguj se</button>

        </div>

      </form>

    </div>

    <div *class*="no-acc-register">

      <p>nemas nalog?</p>

      <button *id*="registracija">Registruj se!</button>

    </div>

*<!-- skrivena registracija -->*

    <div *class*="custom-modal">

      <button *id*="closeModal">X</button>

      <form *id*="registrationForm">

        <h2>Registracija korisnika</h2>

        <div>

          <label *for*="korisnicko\_ime">Korisnicko ime: </label>

          <input

*type*="text"

*id*="korisnicko\_ime"

*name*="korisnicko\_ime"

*placeholder*="nikola nikolic"

          />

        </div>

        <div>

          <label *for*="email">Email addresa: </label>

          <input

*type*="email"

*id*="email"

*name*="register\_email"

*placeholder*="niko.nikolic@gmail.com"

          />

        </div>

        <div>

          <label *for*="lozinka">Lozinka: </label>

          <input

*type*="password"

*id*="lozinka"

*name*="register\_lozinka"

*placeholder*="\*\*\*\*"

          />

        </div>

        <div>

          <label *for*="ponovi\_lozinku">Ponovi lozinku: </label>

          <input *type*="password" *id*="ponovi\_lozinku" *name*="ponovi\_lozinku" />

        </div>

        <div>

          <button *class*="registrujSe">Registruj se</button>

        </div>

      </form>

    </div>

  </body>

</html>