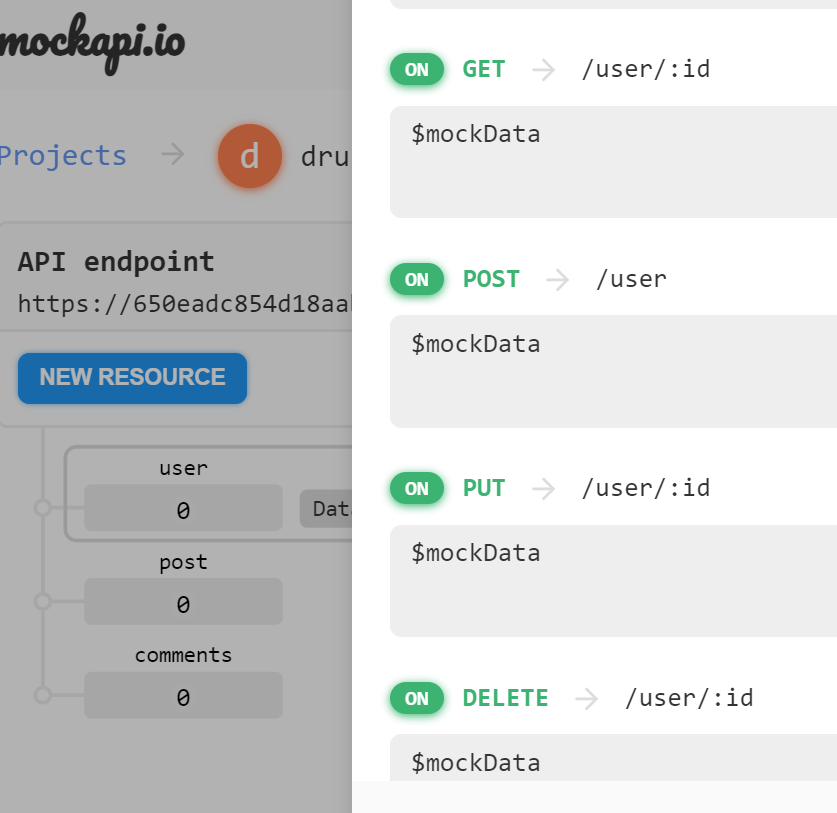
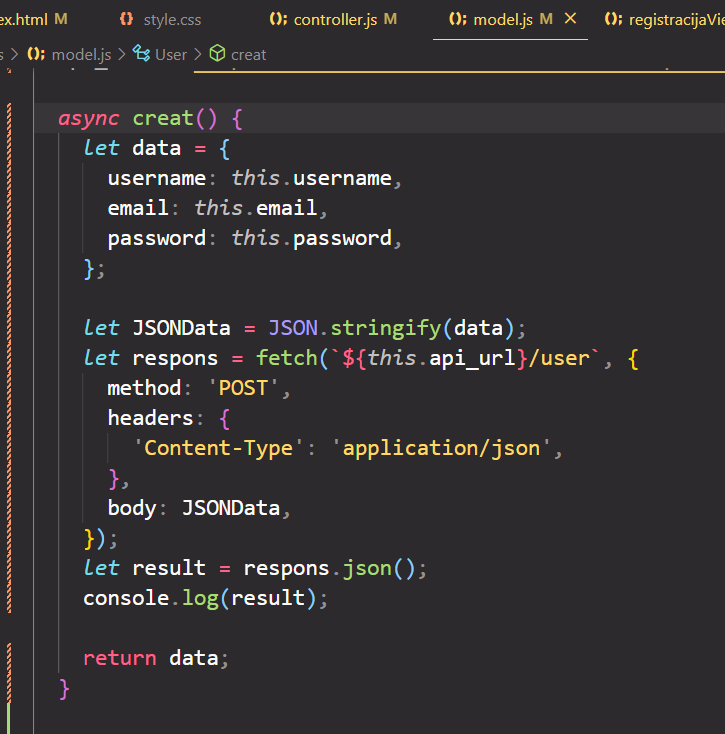
SLANJE PODATAKA NA SERVER ( MOCKAPI )

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 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.

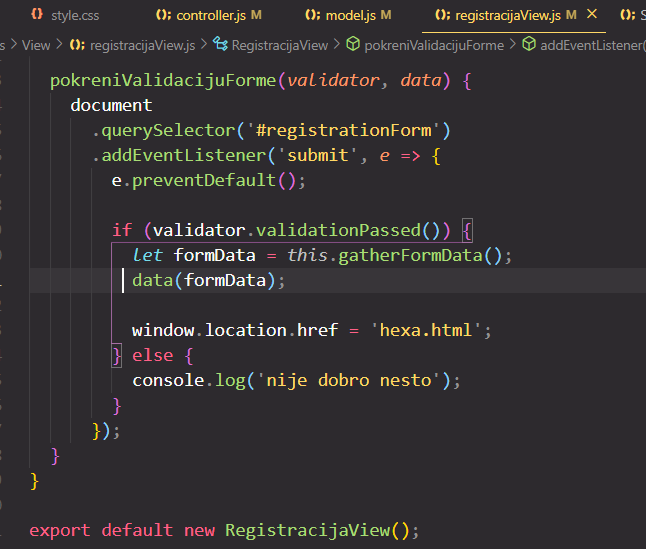
Kada smo to uradili onda kao trece svojstvo fetch() metode pozivamo u body svojstvu u fetch zaglavlju ovu varijablu JSONData.

sada kada se dogodii submit podaci iz forme ce se poslati na server tj. na nas mockAPI.

................................................................................prelazak na hexa.html stranicu..........................................................................

ovde smo bili ostali jako dugo iz razloga sto nam je kod zabagovao a nista nismo uradili...

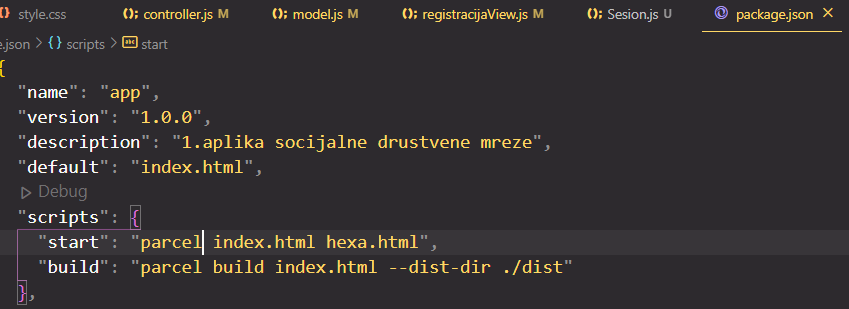
kada zelimo da predjemo sa stranice na stranicu to radimo pomocu window.location.href=’hexa.html’. Ovo unosimo u dio koda kada se dogodi submit.



Ovde vidimo da smo unijeli ispod data(formData).Dakle kada se podaci posalju na api onda zelim da se dogodi prelazak sa index.html na hexa.html stranicu.

.....................................parcel greska........................

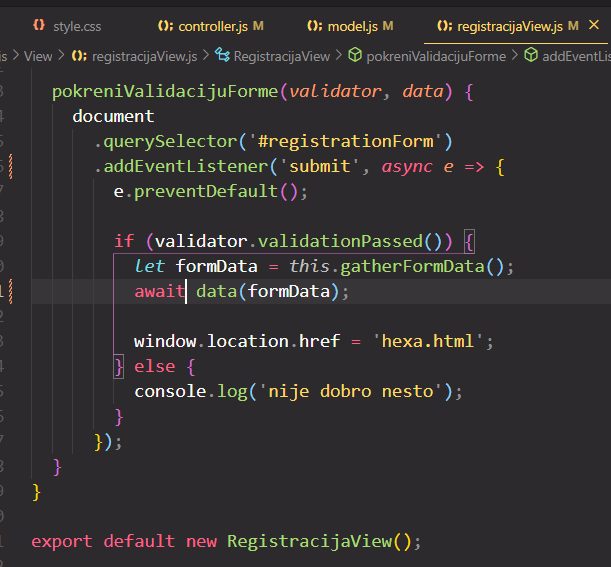
E u ovom dijelu se dogadjaju 2 baga zbog kojeg smo izgubili 15 dana trazeci rjesenje.Nekada se desi da parcel nece ucita podatke iz dist foldera ... sve sto treba jeste daa izbrisemo dist folder i poslije toga opet ucitamo npm run start te ce se sada opet dist instalirati saa svojim nezavisnostima.Ako opet nece onda treba da promjenimo package.json fajl:

Ovde vidimo da sam promjenio start gdje sam postavio da mi parcel ucita sve stranice sa kojima cu raditi:

start: “parcel index.html hexa.html” => dobio sam i savjet na mom gitHubu gdje me savjetuje da odjednom ucitam sve html stranice na ovaj nacin: “start”: “parcel start \*.html” => ali ovo iz nekog razzloga nije radilo pa sam pojedinacno naveo sve stranice koje zelim da mi parcel ocita.

.............................greska sa window.location.href................................

jos jedna greska koja se podkrada je da mi sada samo ucita moju novu hexa.html stranicu kada se dogodi subimt.To se desava iz razloga kada dodje do pokretanja submita dolazi do ucitavanja nove stranice cime se prekidaju procesi sa stare stranice.To cemo sprijeciti upotrebom asinhronog ponasanja nase e funkcije:

Ovde vidimo da smo u .addEventListener => ispred e=> dodali async sto nam daje pravo da koristimo await funkciju na data(formData).  
Dakle sada ce nasa await data(formData) sacekati da se ispuni pa ce tek onda kada se ti podaci (formData) posalju na mockApi , tek poslije toga ce se ucitati nasa hexa.html stranica.