

Projekat iz Razvoja mobilnih aplikacija

Spirala 3

Cilj projekta je da se razvije mobilna aplikacija za vođenje ličnih finansija. Razvoj aplikacije je planiran iterativno tako da se svaka spirala nadovezuje na prethodnu. Niz funkcionalnosti će postepeno rasti, neke će se mijenjati, a neke možda i odbacivati u narednim spiralama. Krajnja aplikacija treba biti upotrebljiva i predstavljati demonstraciju znanja stečenih na predmetu.

Detaljno pročitajte postavku. Svi ID-evi, nazivi i specifikacije klasa trebaju biti kao u postavci. Veoma je važno da se držite postavke!

Potrebno je povezati aplikaciju sa web servisom. Sljedeće funkcionalnosti trebete prepraviti tako da sada dohvataju podatke sa web servisa [10B]:

- Prikaz liste transakcija - obratite pažnju na periodične transakcije one su u web servisu snimljene na datum prve transakcije, ali se u aplikaciji trebaju prikazati nakon svakog intervala sve do isteka transakcije. Ako je interval 30 onda se trebaju prikazati jednom u svakom mjesecu između početka i kraja, ako je interval 15 onda se trebaju prikazati dva puta, ako je 60 onda svaki drugi mjesec itd.
- Filtriranje liste transakcija
- Sortiranje liste transakcija
- Editovanje accounta
- Dodavanje transakcije - servis neće ažurirati budžet nakon dodavanja transakcije vi trebate nakon što se doda transakcija pozvati editovanje accounta i upisati novo stanje. Početno stanje accounta uključuje sve početne transakcije.
- Brisanje transakcije - ažurirati budžet kao kod dodavanja transakcije
- Editovanje transakcije - ažurirati bužet kao kod dodavanja transakcije
- Prikaz transakcije na grafovima

Pristup servisu

Pristupite stranici na [link-u](#). Prijavite se sa fakultetskim pristupnim podacima. Tada bi trebali dobiti kreiran vaš account. Kreiranje accounta može potrajati. Nakon kreiranja accounta u browseru ćete dobiti link do vašeg accounta. Posljednji dio url-a predstavlja id kojeg ćete koristiti za sve pozive prema web servisu. Ovaj ključ snimite u string resource vaše aplikacije pod nazivom `api_id`.

Metode web servisa

ROOT = <http://rma20-app-rmaws.apps.us-west-1.starter.openshift-online.com>

Dohvatanje liste tipova transakcija

GET: ROOT/transactionTypes

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** <http://rma20-app-rmaws.apps.us-west-1.starter.openshift-online.com/transactionTypes>
- Status:** 200 OK
- Time:** 239 ms
- Size:** 786 B

The response body is displayed in JSON format:

```
{
  "count": 5,
  "rows": [
    {
      "id": 1,
      "name": "Regular payment",
      "createdAt": "2020-05-07T16:32:31.941Z",
      "updatedAt": "2020-05-07T16:32:31.941Z"
    },
    {
      "id": 2,
      "name": "Regular income",
      "createdAt": "2020-05-07T16:32:32.063Z",
      "updatedAt": "2020-05-07T16:32:32.063Z"
    },
    {
      "id": 3,
      "name": "Purchase",
      "createdAt": "2020-05-07T16:32:32.175Z",
      "updatedAt": "2020-05-07T16:32:32.175Z"
    },
    {
      "id": 4,
      "name": "Individual income",
      "createdAt": "2020-05-07T16:32:32.287Z",
      "updatedAt": "2020-05-07T16:32:32.287Z"
    },
    {
      "id": 5,
      "name": "Individual payment",
      "createdAt": "2020-05-07T16:32:32.434Z",
      "updatedAt": "2020-05-07T16:32:32.434Z"
    }
  ]
}
```

Slika 1. Dohvatanje liste tipova transakcija

Dohvatanje podataka o accountu

GET: ROOT/account/ID

ID predstavlja pristupni ključ kojeg ste dobili na prvom koraku kada ste se registrovali.

The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** <http://rma20-app-rmaws.apps.us-west-1.starter.openshift-online.com/account/77fb0b9c-6c65-410b-a0f9-8fa0956f72c1>
- Status:** 200 OK
- Time:** 417 ms
- Size:** 404 B

The response body is displayed in JSON format:

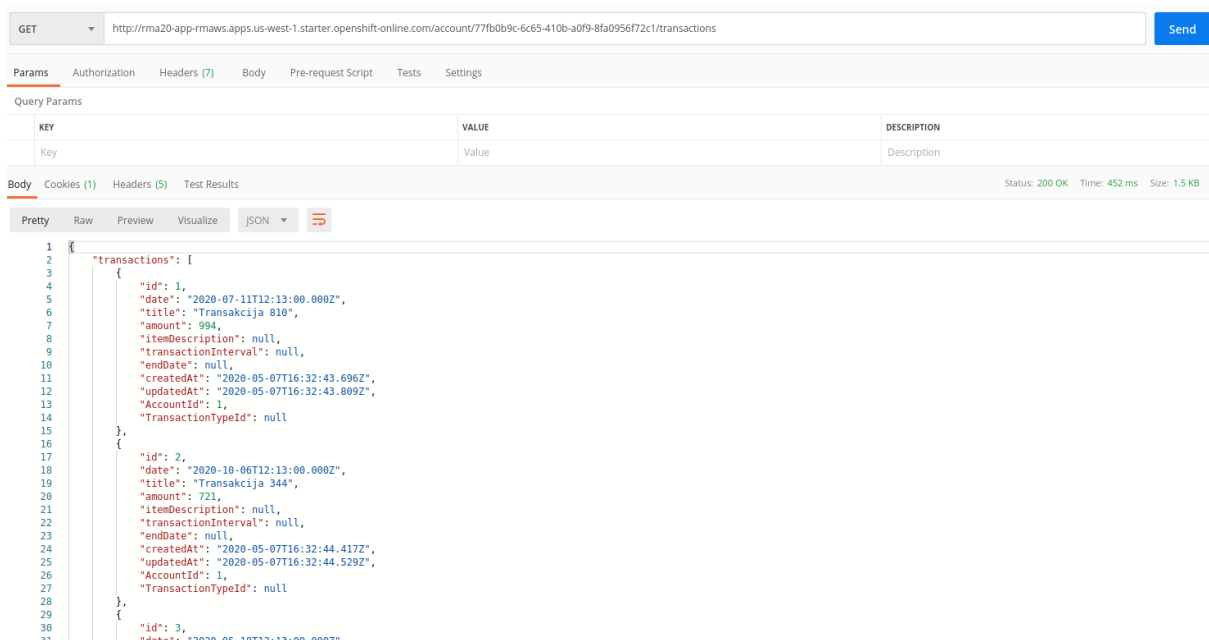
```
{
  "id": 1,
  "budget": 1000,
  "totalLimit": 1000,
  "monthLimit": 100,
  "aHash": "77fb0b9c-6c65-410b-a0f9-8fa0956f72c1",
  "email": "iprazina1@tf.unsa.ba",
  "createdAt": "2020-05-07T16:32:43.586Z",
  "updatedAt": "2020-05-07T16:32:43.586Z"
}
```

Slika 2. Dohvatanje podataka o accountu

Dohvatanje svih transakcija

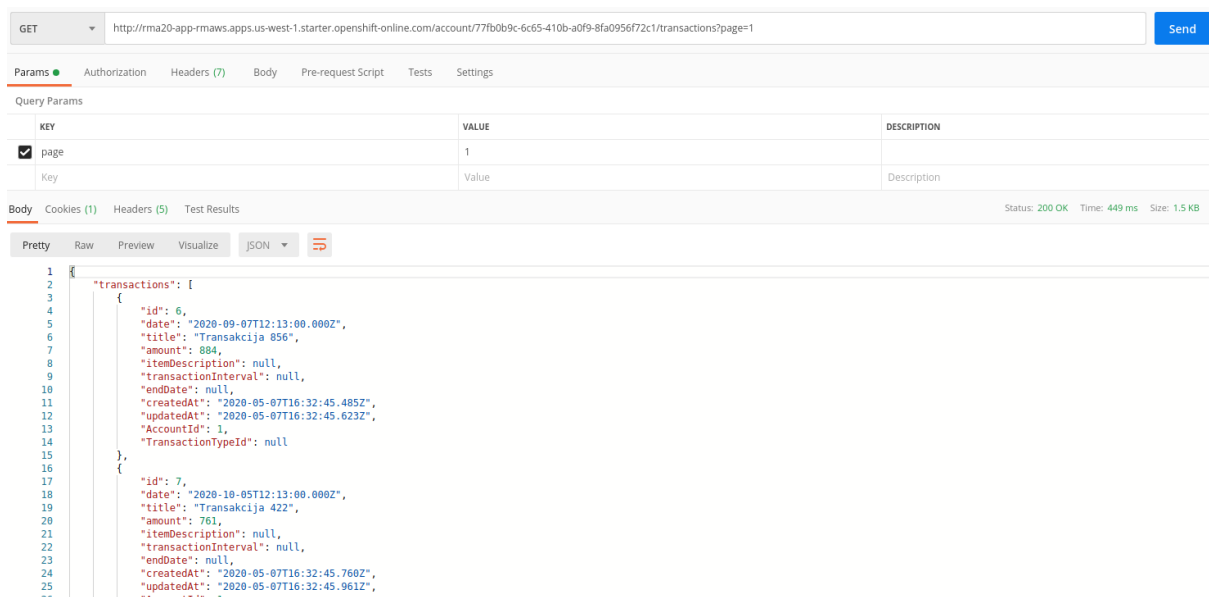
GET: ROOT/account/ID/transactions[?page]

Vraća se lista prvih 5 transakcija ako page parametar nije poslan - ovo predstavlja page=0, svaki sljedeći skup 5 transakcija dobijate tako što page uvećate za jedan.



Slika 3. Dohvatanje svih transakcija

Sljedećih 5 transakcija dobijamo:



Slika 4. Dohvatanje svih transakcija

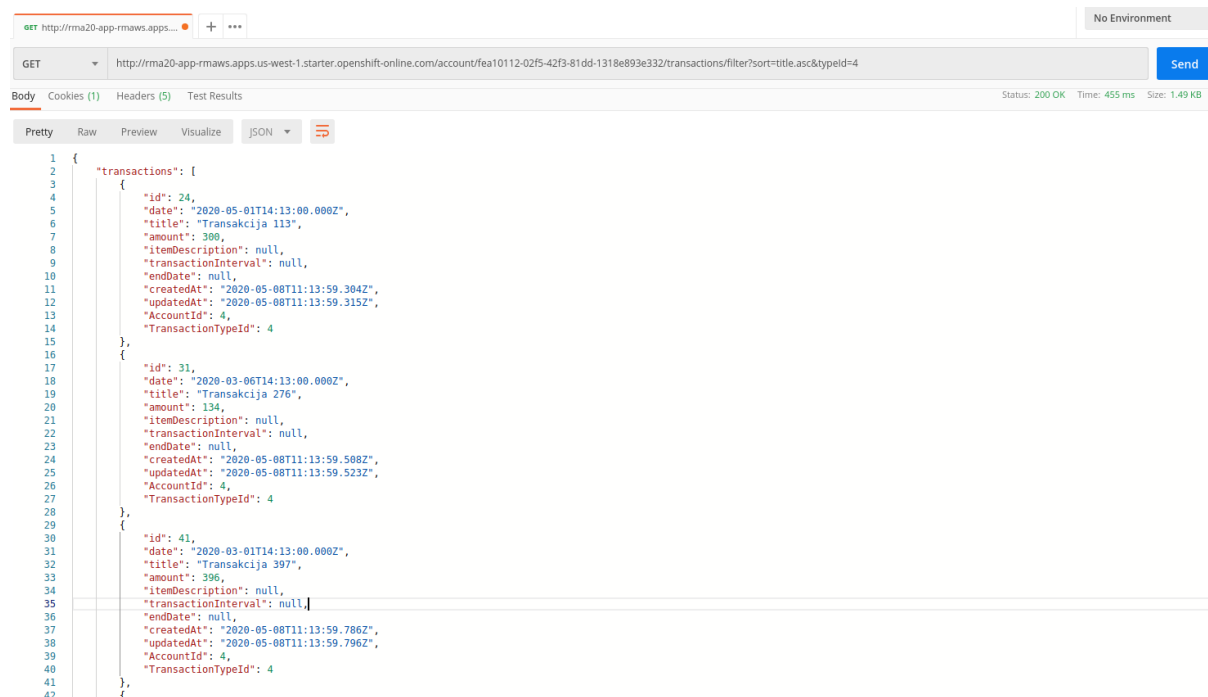
Filtriranje i sortiranje transakcija

GET: ROOT/account/ID/transactions/filter[?page,typeId,sort,month,year]

Parametar page je isti kao i ranije. Ostali parametri su:

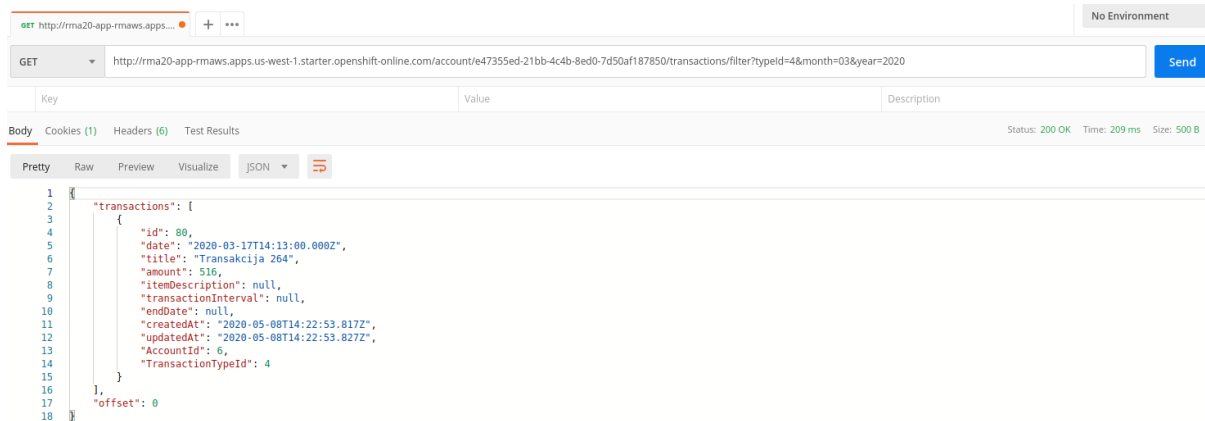
- typeId - id tipa transakcije po kojem filtrirate
- month - dvocifreni broj mjeseca, za jednocifrene mjesece dodajte 0 ispred, sve transakcije koje počinju u zadatom mjesecu
- year - broj godine, sve transakcije koje počinju u zadatoj godini
- Prethodne parametre je moguće kombinovati, kada se više njih navede uslov između njih je logičko i (and).
- sort - sortira transakcije na osnovu vrijednosti koja može biti A.B, gdje je A bilo koji element iz skupa title,date,amount, a B je element iz skupa asc,desc. Ako želite sortirati transakcije po cijeni od jeftinije do skuplje onda vrijednost sort atributa treba biti sort=amount.asc

Svi parametri su opcionalni i mogu se kombinovati sa drugima Primjer: transakcije tipa individual income sortirane po nazivu



Slika 5. Filtriranje i sortiranje transakcija

Primjer: transakcije iz mjeseca marta, trenutne godine koje su tipa Individual income:



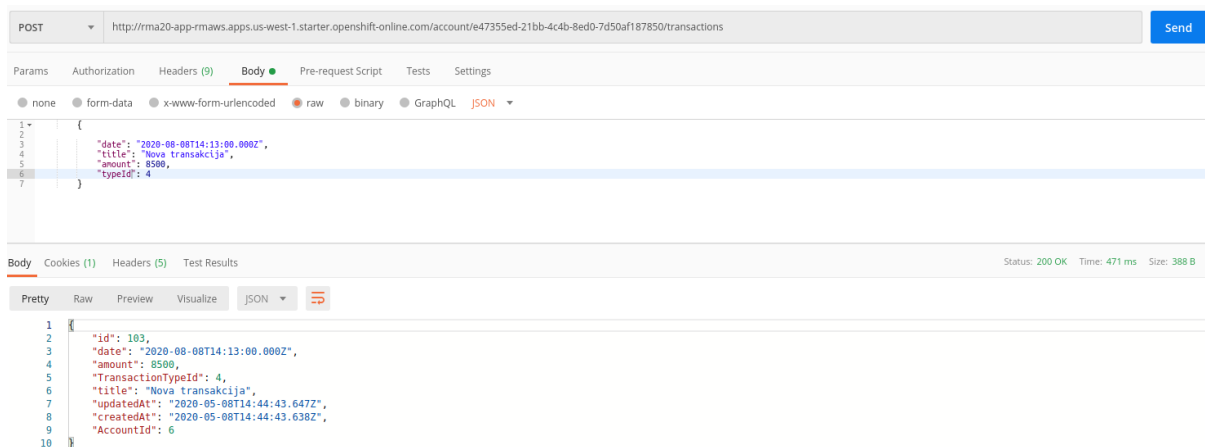
Slika 6. Filtriranje i sortiranje transakcija

Dodavanje transakcije

POST: ROOT/account/ID/transactions

Atribut transakcije trebate navesti u tijelu zahtjeva u JSON formatu. Atributi koje možete navesti su: "date", "title", "amount", "endDate", "itemDescription", "transactionInterval", "typeId".

Primjer: dodavanje transakcije individual income:



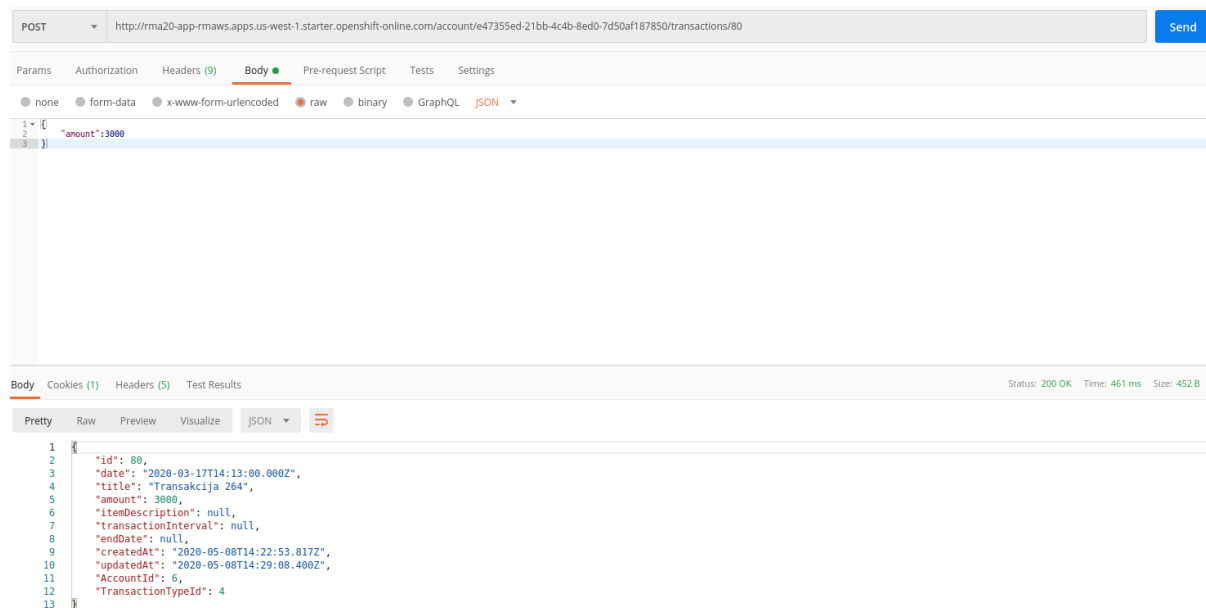
Slika 7. Dodavanje transakcije

Ažuriranje transakcije

POST: ROOT/account/ID/transactions/TRANSACTIONID

Atribut transakcije koje želite ažurirati šaljete u tijelu zahtjeva u JSON formatu. Atributi kod kojih je dozvoljena izmjena su: "date", "title", "amount", "endDate", "itemDescription", "transactionInterval", "typeId". U zahtjevu možete kombinovati više atributa istovremeno.

Primjer: ažuriranje iznosa transakcije sa id-em 80:

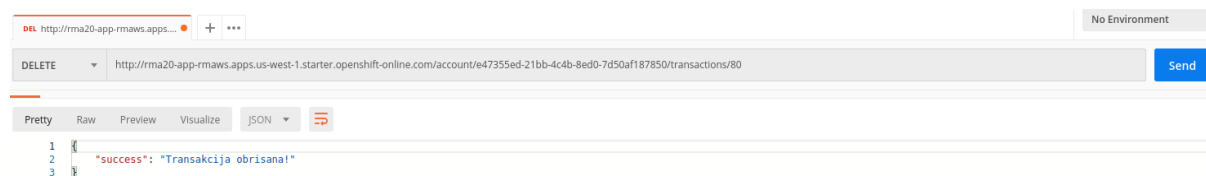


Slika 8. Ažuriranje transakcije

Brisanje transakcije

DELETE: ROOT/account/ID/transactions/TRANSACTIONID

Primjer: brisanje transakcije sa id-em 80



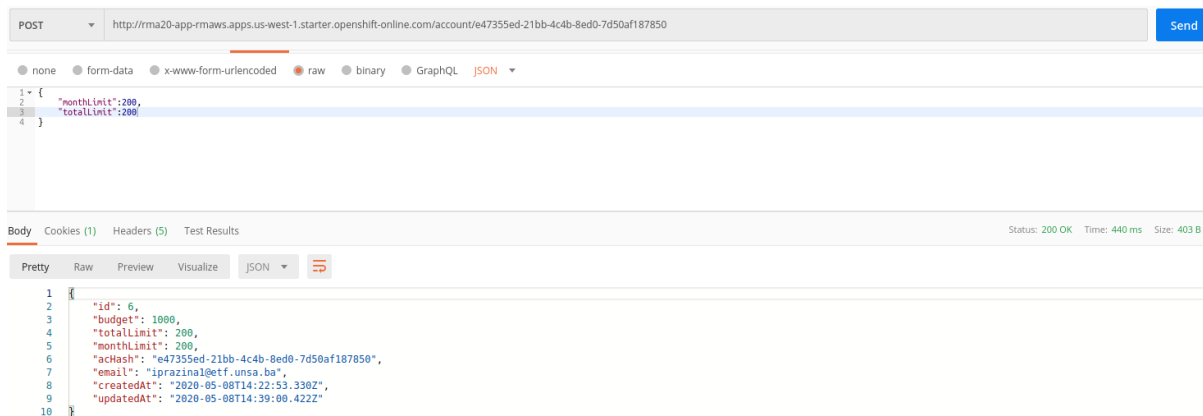
Slika 9. Brisanje transakcije

Ažuriranje accounta

POST: ROOT/account/ID

Atribut accounta koje želite ažurirati šaljete u tijelu zahtjeva u JSON formatu. Atributi koji se mogu mijenjati su: "budget", "totalLimit", "monthLimit". U zahtjevu možete kombinovati više atributa istovremeno.

Primjer: ažuriranje mjesečnog i totalnog limita na 200



Slika 10. Ažuriranje accounta

Alat korišten za ručno pozivanje servisa je [Postman](#).

Za slanje zahtjeva drugim HTTP metodama osim GET, u ovom slučaju trebat će vam i metode POST i DELETE pogledajte dokumentaciju [URLConnection](#) klase i sljedeći [primjer](#).

Opšte postavke projekta

Minimalna verzija SDK Androida: **21**.

Target verzija SDK Androida: **23**.

Verzija MPAndroidChart biblioteke: **v3.1.0**

Rok za rad na spirali 3 je **23.05.2020 do 23:59**. Spirala se treba nalaziti na istom repozitoriju kao i prethodna spirala. Novu spiralu postavite na master branch, a prethodnu spiralu postavite na branch **spirala2**.

Implementirajte sve u skladu sa **MVP** arhitekturom. U slučaju da ne koristite MVP arhitekturu možete ostvariti maksimalno **80%** bodova spirale.

Pitanja vezana za postavku spirale postavljajte kroz **Google Chat**.