

Skavtko v2.0

Osnovni podatki

Člani skupine

Št. skupine: 01

Peter Rovtar

Peter Šavron

Github povezava

<https://github.com/provtar/PRPO-Skavtek>

Kratek opis

Oba sva skavtska voditelja. Pri izbiri teme smo pomislili na potrebe skavtske organizacije. Cilj projekta je sestaviti mikrororitve, ki se bodo integrirale z mobilno aplikacijo, ki olajša administrativno upravljanje srečanj preko beleženja prisotnosti članov ter pomaga pri iskanju terminov preko mikrororitve, ki omogoča članom izbrati termine, ko so na razpolago.

Ogrodje in razvojno okolje

Uporabljali bomo IDE Visual Studio Code in dodatke, ki so na voljo, da olajšajo programiranje v Javi, typescriptu, pa drugih jezikih, ki jih bomo uporabljali.

Naš build tool bo Maven, spodaj prilagamo seznam odvisnosti, ki smo jih vključili v projekt:

- kumuluzee-bom: za usklajevanje verzij osnovnih kumuluzee gradnikov;
 - kumuluzee-openapi: za ustvarjanje openapi dokumentacije preko anotacij znaotraj kode
 - kumuluzee-openapi-ui: za izdelavo grafičnega vmesnika z dostop do openapi dokumentacije
 - kumuluzee-ee: za razrešitev CORS zahtev brskalnikov
 - kumuluzee-streaming-kafka: za uporabo sporočilnega sistema Kafka, ki bistveno olajša komunikacijo med mikrororitvami
 - postgresql: za dostop do podatkovne baze
 - HikariCP: za upravljanje bazena povezav do podatkovne baze
 - hibernate-hikaricp: kotimplementacijo JPA specifikacij za objektno relacijsko mapiranje (ORM)
 - gson: za serializacijo in deserializacijo objektov, ki se pošiljajo preko Kafka sporočilnega sistema, da se izognemo težavam s konfiguracijo standardnih serializatorjev
 - jackson-datatype-jsr310: za serializacijo javanskega LocalDateTime formata;
- Kot že povedano smo za razvoj mikrororitv uporabljali kumuluzee enostavno, v Sloveniji razvito ogrodje, ki omogoča hitro integracijo gradnikov za razvoj modernih, skalabilnih mikrororitv.

Gradnja slik

Gradnjo slik smo izvedli v Dockerju, mikrororitve se poženejo na openjdk sliki, pri tem se slike za lokalno in produkcijsko okolje razlikujejo, namreč v lokalnem okolju, dodajamo plasti nad sliko openjdk:17-alpine,

ki podpira samo x86 procesorje, medtem ko v oblčnih storitvah uporabljamo openjdk:17-slim, zgrajeno za ARM64 procesorje. Slika za kafka je apache/kafka:latest.

Docker smo uporabljali tudi za lokalno integracijo in testiranje slik, vse mikrostoritve smo lahko naenkrat pgnali z docker-compose datoteko.

Namestitev v gruče

Naš ponudnik oblčnih storitev je Oracle, ki ima določene omejitve za poskusne račune, namreč ne dovoljuje več kot 3 porazdelilce obremenitve na poskusni račun, zato smo uporabili Ingress preslikovalno tabelo, ki usmerja zahteve na port 80 na različne mikrostoritve glede na pot http zahtevka. Usmerjanje opravlja storitev traefik:v2.9. Na gručo je tudi nameščen Kafka sporočilni sistem.

Grafični vmesnik

Je razvit v Angularju, izgled elementov je oblikovan v CSS-ju. Http zahtevke pošiljamo na zaledje s pomočjo HttpClient knjižnice, znotraj komponent pa uporabljamo še ReactiveFormsModule in CommonModule. Frontend trenutno ni nameščen v oblak.

Oblčne storitve:

Kubernetes gruča je nameščena na Oraclu, ima 2 vozlišči, in en sam porazdelilec obramenitve, kar zmanjša stroške. Podatkovno bazo nudi supabase, ki je gostovan na AWS strežnikih. Oba strežnika se nahajata v srednji Evropi, zato komunikacija med njima poteka hitro.

Zahteve

Repozitorij

Je dostopen na GitHubu (<https://github.com/provtar/PRPO-Skavtek>), razvoj je potekal na veji dev in njenih sinovih. V datoteki readMe so razni zagoni za v ukazno vrstico.

Razvojno okolje

Bolj obširno je razloženo zgoraj, datotečni sistem je razdeljen na frontend, ki se še razveji v storitve (za povezavo do zaledja) in komponente, ki se delijo po namenu: skrita okna, sezname, ogrodje spletne strani in obrazci.

Vsaka mikrostoritev je v svoji mapi, ima svoj maven projekt s tremi moduli: api za implementacijo REST API-ja, storitve za vso poslovno logiko in entitete za ustvarjanje entitetnih preslikav.

REST API

Vsaka mikrostoritev ima podprtih le toliko klicev, kolikor jih je potrebnih za njeno izvajanje. Storitve skupine ima tako končne točke za delo s skupinami, kot končne točke za upravljanje s člani.

Ob uspehu vračajo storitve 200, delete metode 204, post metode pa 201. Iskanje po identifikatorju vrne 404, če entiteta se ni našla.

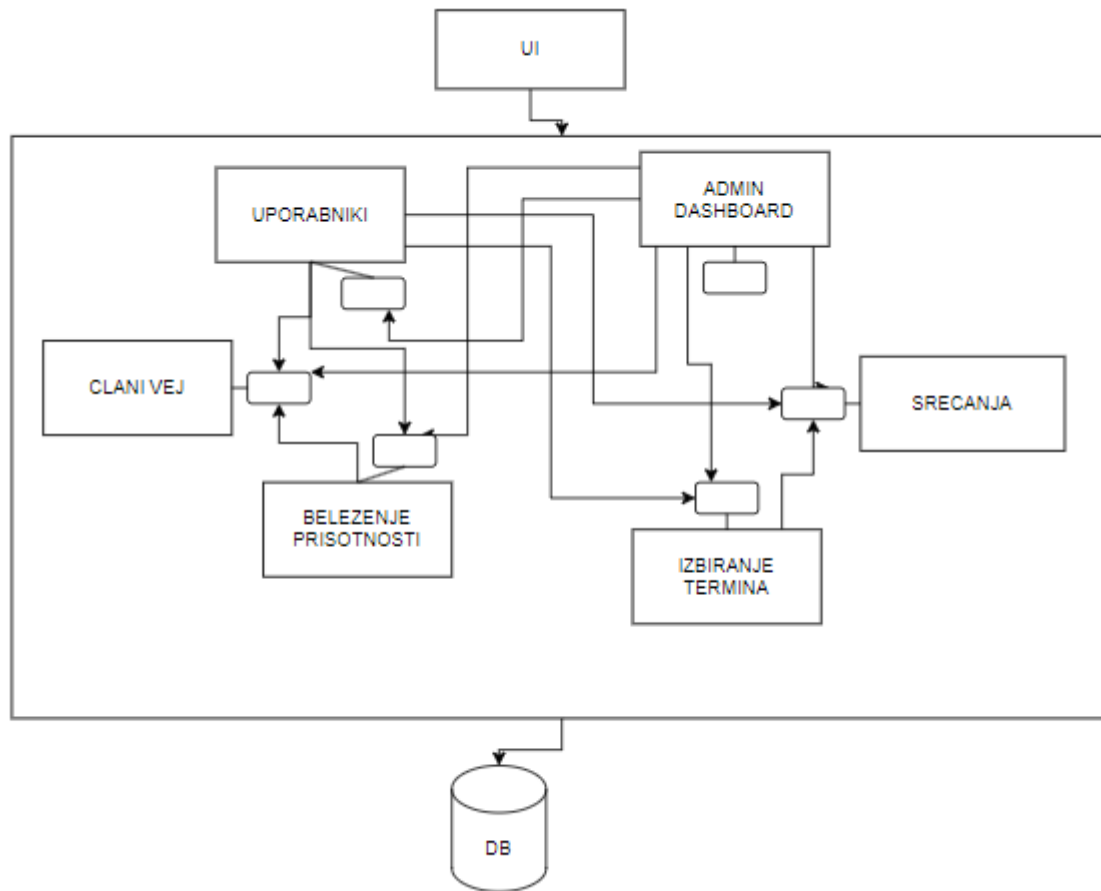
Dokumentacija

API je dostopna v grafični obliki na strežnikih.

Vsebniki

Lokalno jih lahko zaganjamo z ukazom java -ja, z docker run, preko docker compos-a, pa preko namestitve na docker desktop kubernetes gručo.

Shema arhitekture



Mikrostoritve

1 - Upravljanje članov

- Registracija uporabnikov
- Posodabljanje podatkov in pravic uporabnikov
- Vpis v storitev
- Dostop do podatkov drugih uporabnikov

2 - Delovne skupine

- Dodeljevanje skupin članom
- Posodabljanje skupin
- Dostop do podatkov o skupini

3 - Beleženje prisotnosti

- Pridobivanje prisotnosti članov na enem srečanju

- Pridobivanje prisotnosti člana skozi časovno obdobje
- Pridobivanje prisotnosti za vsa srečanja skupine (v časovnem intervalu)
- Pridobivanje statistike za skupino (kolkokrat vsak prisoten odsoten)
- Dodatne analize po potrebi
- Dodajanje prisotnosti na srečanju (trenutno za vse, se da status ni bilo namenjeno njemu - siv)
- Posodabljanje posameznih prisotnosti
- Brisanje prisotnosti(za srečanje)
- Vpisovanje prisotnosti članov na srečanjih

4 - Srečanja

- Omogoča, da ustvariš novo srečanje
- Udeležencem srečanja komunicira podatke o srečanju
- Posodobi razpoložljivost udeležencev srečanja
- Udeleženec potrdi svojo prisotnost na srečanju

5 - Izbiranje razpoložljivosti za srečanja

- Vsak uporabnik posreduje svoje proste termine
- Posodabljanje izbire prostih terminov
- Dostop do podatkov

6 - Ustvarjanje zapisov osebnega napredovanja

- Voditelj lahko spremlja svoje varovance in jim dodaja zapiske s svojimi opaznanji
- Zapise lahko posodablja in briše
- Ter si ogleda zapiske sovoditeljev

Primeri uporabe

Jurij si beleži prisotnost udeležencev skupine Bobri

Jurij si želi na enostaven in preprost način beležiti prisotnost udeležencev na njegovih srečanj in želi do teh podatkov dostopati na več platformah. Ustvaril si bo račun na Skavtku, potem bo ustvaril skupino Bobri, pa dodal v njo člane skupine. Prihodnje srečanje začne z beleženjem prisotnosti. Po pomoti je napisal, da je bil Pavel odsoten, ko se tega zave, popravi zapis in ga označi za prisotnega. Po nekaj časa se Franc zapusti skupino, zato ga Jurij odstrani iz skupine. Konec leta pregleda prisotnosti vseh članov, da ugotovi kdo je bil prisoten na več kot 80% srečanjih.