PRPO-Skavtek

Repository for a project at the PRPO subject

Gradnja in zagon projekta

Kot prvo build projekta

```
mvn clean package
```

Projekt po novem uporablja enviromental variables v persistence.xml datoteki, zato je treba poganjati s parametri.

Opcije za build docker slik lokalno

Primer zagona z javo direktno (powershell, bash, ipd):

```
java -DDB_URL="jdbc:postgresql://localhost:5435/db" -DDB_USER=admin -
DDB_PSW=admin -DDB_DRIVER="org.postgresql.Driver" -DKAFKA_URL="localhost:9093"
-jar .\api\target\api-1.0.0-SNAPSHOT.jar
```

Zagon v dockerju brez compose filu (ne vem zakaj bi si tega zelel)

```
docker build -t skavtko:plain .
docker run -e DB_URL="jdbc:postgresql://host.docker.internal:5435/db" -e
DB_USER=admin -e DB_PSW=admin -e DB_DRIVER="org.postgresql.Driver" -p 8085:8080
skavtko:plain
```

Zagon z rabo docker compose

Lokalni docker deployment:

```
docker build -t skavtko:alpine-1 .
docker compose -f .\docker-compose-db.yaml up -d
docker compose -f .\docker-compose-skavtko-local.yaml up -d
```

Dostopa do databasa preko spremenljivk v .env datoteki (ki ni javna, ker jo docker ignora)

```
docker compose -f .\docker-compose-skavtko-supabase.yaml up -d
```

Primer .env fila:

```
DB_URL=jdbc:postgresql://db.com:5432/ime_db
DB_USER=user
DB_PSW=moje_geslo
DB_DRIVER=org.postgresql.Driver
```

Zagon v kubernetesu

Pogoji: usposobljen kubernetes na dockerju, kubectl

kubectl create configmap skavtko-ms-config --from-env-file=.env

Preveri, da ciljas na pravi cluster (docker-desktop):

```
kubectl config current-context
kubectl config get-contexts
```

Za spremeniti context

```
kubectl config use-context <ime-context>
```

Nastavi podatke DB v secret dovolj narediti enkrat (nastavi podatke databasa, ki bos uporabil, to je za notranji container, ki si ga mores sam nastaviti v docker, ne deluje, ker nima dostopa do localhost):

```
kubectl create secret generic skavtko-db-secret --from-
literal=DB_URL="jdbc:postgresql://host.docker.internal:5435/db" --from-
literal=DB_USER=admin --from-literal=DB_PSW=admin --from-
literal=DB_DRIVER="org.postgresql.Driver"
```

Nastavitev registry-ja za docker slike za kubernetes (docker tag se uporabi za taggat ze obstojeco sliko):

```
Kubernetes lokalno:
docker run -d -p 5000:5000 --restart=always --name registry registry:2

docker build -t localhost:5000/skavtko:k8s-2 .

docker tag skavtko:alpine-1 localhost:5000/skavtko:k8s-2

docker push localhost:5000/skavtko:k8s-2
```

Ustvari service za pod:

```
kubectl apply skavtko-service.yaml
```

Ustvari deployment na kubernetes:

```
kubectl create -f skavtko-deployment.yaml
```

Unici deployment po potrebi:

```
kubectl delete deployment skavtko-api
```

Uporabni ukazi za upravljanje s podi:

```
kubectl get secrets
kubectl delete secret <ime-secret>
kubectl get deployments
kubectl get nodes
```

Build za Oracle virtualko

Login do repositorija:

```
docker login fra.ocir.io
```

Ustvarjanje docker slike

Rabimo arm instance zato treba specificirati platformo

```
docker pull --platform linux/arm64/v8 openjdk:17-slim
```

Build slike (trenutno ni potrebno, ker je slika ze v registru):

```
docker build -t fra.ocir.io/frcrlvuwzvij/skavtko:api-1 --
platform=linux/arm64/v8 -f oracle.Dockerfile .
```

V slucaju, da je se kje zbuildana slika:

```
docker tag <slika> fra.ocir.io/frcrlvuwzvij/skavtko:api-1
```

Push na Oracle registry:

```
docker push fra.ocir.io/frcrlvuwzvij/skavtko:api-1
```

Kubernetes deployment na oracle

Login v cluster: dobis pri detajlih clusterja na oracle cloud spletni strani, podobno temu.

```
 oci ce cluster create-kubeconfig --cluster-id ocid1.cluster.oc1.eu-frankfurt-
1.<veliko crk OCID> --file $HOME/.kube/config --region eu-frankfurt-1 --token-
version 2.0.0  --kube-endpoint PUBLIC_ENDPOINT
```

Ustavri secret za dostop do registra:

```
kubectl create secret docker-registry ocr-login-secret --docker-server=<region-
key>.ocir.io --docker-username=<tenancy-name>/<username>    --docker-password=
<oci-api-key> --docker-email=<email>
```

Secret za dostop do db:

```
kubectl create secret generic supabase-db-login-secret --from-
literal=DB_URL="jdbc:postgresql://host.docker.internal:5435/db" --from-
literal=DB_USER=admin --from-literal=DB_PSW=admin --from-
literal=DB_DRIVER="org.
```

Zazeni se service in pode:

```
kubectl apply -f skavtko-service-oracle.yaml
kubectl apply -f skavtko-deployment-oracle.yaml
```

Za dostop poglej:

```
kubectl get svc
```

IP naslov load balancerja

Razno

Nek tutorial za aplikacijo na compute instance (mi tega ne bomo delali):
https://docs.oracle.com/en/learn/java_app_ampere_oci/index.html

Stari podatki

Startaj container:

```
docker start resources-db-1 resources-adminer-1
```

Stop container:

```
docker stop resources-db-1 resources-adminer-1
```

Dostop do adminer

http://localhost:8081/

Database specifikacije za login (so tud u docker compose):

```
Tip Postgresql
Database: db
User: admin
Password: admin
```

Namestitev v oblak

Dtabase predogi:

- https://codeless.co/free-postgresql-hosting/
- https://heliohost.org/ (problem : je v ameriki, velik ping)
  Za mikrostoritve:
-

API Dokumentacija

Link do dokumentacije>

- http://localhost:8080/api-specs/ui/

Angular

instalacija:

- npm install -g @angular/cli@18.2
- npm run ng server (zazene server, stran dostopna na http://localhost:4200)
- npm run ng generate component|service (ustvari nove komponente za spletno stran)
- stvar je se v delu, treba se routing zrihtat

TODO

- ustvari en class, v katerem so definirana imena vseh column, ki vsebujejo imena atributov po katerih se preslika med relacijami
- Dodaj equal in hash vsem entitetam @Override

Pametni linki do dokumentacij

- JPQL
- https://docs.oracle.com/cd/E11035_01/kodo41/full/html/ejb3_langref.html
- Hibernate
- https://docs.jboss.org/hibernate/orm/6.6/introduction/html_single/Hibernate_Introduction.html
- HQL
- https://hibernate.org/orm/documentation/5.6/
- https://docs.jboss.org/hibernate/orm/3.3/reference/en/html/queryhql.html

Entity dodatno

- https://www.baeldung.com/jpa-hibernate-associations
- https://www.baeldung.com/hibernate-inheritance
- Rabiti enum kot diskriminatorje za podtipe
- https://stackoverflow.com/questions/3639225/single-table-inheritance-strategy-using-enums-as-discriminator-value#:~:text=If you try to use an enum as,discriminator types allowed are String%2C Char and Integer.
- Raba getReference vs find
- https://stackoverflow.com/questions/1607532/when-to-use-entitymanager-find-vs-entitymanager-getreference-with-jpa
- HQL Query z null paarmetri (2. odgovor)
- https://stackoverflow.com/questions/2123438/hibernate-how-to-set-null-query-parameter-value-with-hql

Database

Hibernate ima nastavitve, da se poveže na postgres DB

CLAN

- id generated
- ime required
- priimek required
- skavtsko_ime
- steg
- vloga (aktiven-pasiven-admin) required

AktivenClan (podtip)

- username
- password
- mail

PasivenClan

- master - skupina kateri pripada
- mail

PasivenClan

- master - skupina kateri pripada

SKUPINA

- id
- ime
- opis
- []{link, ime}

CLAN-SKUPINA

- id_clana
- id_skupine
- vloga

SRECANJE

- id
- ime
- belezenje (ja-ne)
- kraj
- datum-ura od
- datum-ura do
- opis
- id_skupine

PRISOTNOSTI

- id_clan
- id_srecanja
- prisotnost (pris, online, ods, ods_opr)
- komentar

TERMINI

- id_clanAktiven

- timestamp_od
- timestamp_do

PERMISSION

- id_clanAktiven
- id_clanAktiven
- id_skupine
- #tip premission (view, view_termini, crud, clani; view, upravljaj skupina; view_srecanja, crud_srecanje; view_prisotnosti, crud_priostnposti)
- token (?strategija)

TIPI_PERMISSION

- tip
- ime
- opis

LOG

- id_oseba
- id_skupina
- id_oseba
- id_skupina
- akcijo

API na kratko

Path /registracija

Dodajanje active userjev

Path /clani

Dodajanje navadnih userjev

Path /clani/id

Upravljaniìje z userji po id-ju

Path /skupina

Upravljanje skupin

Path /skupina/id/clani

Upravljanje s clani skupine

Path /srecanje

Path /prisotnosti/id

Path prisotnost/id/clani/clanId

Termini /clanId

Zrna

Clan

- Dodaj uporabnika
- Dodaj aktivnega uporabnika
- Posodabljaj uporabnika
- Dostopaj do javnih podatkov uporabnikov

Frontend

Form

Tukaj so komponente vseh formov, ki se jih posilja apiju:

- Samo form za posiljanje podatkov
- Podatki, ki formu rabjo, das preko inputov, dodas init, da se ga lahko na novo poslje, ce je potrebno
- Inicializacija v ngOnChanges, da se sproti ponovnoustvarja form, ko se inputi spreminjao