Picasso

(orodje za risanje slik)

Člani skupine: Matic Pristavnik Vrešnjak

Opis projekta

V okviru projekta za RSO sem naredil orodje za urejanje in risanje slik. Funkcionalnosti oz. storitve, ki sem jih implementiral so urejanje uporabnikovega računa ter risanje in shranjevanje slik. Uporabnik lahko izdelek, ki ga je naredil shrani v Dropbox repozitorij v katerem se ustvari datoteko v kateri se hranijo njegovi izdelki.

Opis okolja

Za izdelavo projekta sem uporabil sledeča ogrodja:

* Spring boot sem uporabil za zaledni del aplikacije. Zaradi tega sem uporabil programski jezik Java
* React za sprednji del aplikacije. React ogrodje je namenjeno za programski jezik JavaScript.
* Uporabil sem Postgress podatkovno bazo za trajno hranjenje podatkov

Postavljeni sem imel dve Kubernetes okolji.

Prvo okolje je bilo vzpostavljeno na Azure , kjer sem za zaledni sistem uporabil ingress kontrole, ki usmerja zaledni sistem kot sprednji del. Konfiguracijski strežnik je bil nameščen kot LoadBalancer, saj drugače ni mogel pravilno delovati. Azure sem uporabil za končno objavo

Drugo okolje je bilo postavljeno z orodjem Minikube. To okolje sem uporabil , ko sem testiral delovanje Kubernetes-a ali v primeru, ko je prihajalo do težav z Azure. V primeru minikube uporabljam samo ingress kontrole kjer so konfigurirane vse poti , tako do zalednega kot tudi sprednjega dela aplikacije.

Povezave do projekta

Povezava do Github repozotrijev:

Zaledni del za uporabnike in frontend : <https://github.com/MaticVP/RSO_APP>

Zaledni del za risalnik : <https://github.com/MaticVP/RSO_DRAW>

Povezava do Docker hub :

Docker hub image za sprednji del oz. React:

<https://hub.docker.com/repository/docker/pbsem2/frontend_image/general>

Docker hub image za risalnik: <https://hub.docker.com/repository/docker/pbsem2/drawer_image/general>

Docker hub image za uporabnike:

<https://hub.docker.com/repository/docker/pbsem2/new_image/general>

Url naslovi

Azure: <http://20.84.17.217>

Minikube: <http://127.0.0.1>

Shema arhitekture in interakcij med mikrostoritvami

Slika, ki vsebuje besede besedilo, risanje, skica, rokopis

Opis je samodejno ustvarjen

Seznam funkcionalnosti

RSO\_APP skrbi za urejanje uporabnikov in njihovih storitev. Tako ob prijavi uporabnika preverja, če je geslo in uporabniško ime pravilno. Omogoča tudi posodabljane informacij oz. nastavitev uporabniškega računa. To dosega s sledečimi kontrolnimi točkami:

* add
* recover
* login
* get-id
* get-photo
* upload-photo

Swagger dokumentacija vseh končnih točk za uporabniški API: Slika, ki vsebuje besede besedilo, posnetek zaslona, številka, vzporedno

Opis je samodejno ustvarjen

RSO\_DRAW skrbi za urejanje podatkov slik. RSO\_DRAW skrbi, da uporabnik lahko shrani sliko na Dropbox. Preverja oz. skrbi tudi, da uporabnik vidi samo svoje projekte in ne projekte drugih uporabnikov. To dosega s sledečimi točkami:

* add
* save-project
* get-project
* get-projects
* save-projects

Swagger dokumentacija vseh končnih točk: Slika, ki vsebuje besede besedilo, posnetek zaslona, številka, vzporedno

Opis je samodejno ustvarjen

Seznam implementiranih kontrol stanja in metrik

Implementirana kontrole stanja so:

* DropboxHealth, ki preveri povezavo do DropBox API
* FrontendHealth, ki preveri povezavo do sprednjega dela
* DatabaseHealth, ki preveri povezavo do podatkovne baze
* UserHealth, ki preveri povezavo do mikrostoritve za uporabnike

Implementirane metrike so:

* dropboxAPICall, ki pove koliko krat se je klical Dropbox API
* responseTime, ki pove koliko časa je potreboval za odziv Dropbox API

Zunanji API

Za zunanji API sem uporabil Dropbox API. Potreboval bi namreč FTP strežnik,izkazalo se je, da je to dobra nadomestitev. Dropbox API potrebuje žetone za pravilno dostopanje zaradi tega sem mogel nastaviti žetone s katerim se lahko povežem do njihovega API. Dropbox API se uporablja za shranjevanje slik

Opis centraliziranega beleženja dnevnikov

Za beleženje dnevnikov sem uporabil log4j in logit.io. V Spring boot se lahko izognemo interceptorjem tako, da podamo log4j yaml datoteko v katero zapišemo katere informacije želimo poslati. V mojem primeru pošljem čas, verzijo API, ID niti, ime API, sporočilo. Sporočilom sledim tako, da javim vstop in izstop iz storitve oz. posameznih končnih točk, ki so na poti do končnega rezultata.

Viri konfiguracije

Vir konfiguracije v Spring boot je datoteka application.properties. V tej datoteki so najbolj relevantne spremenljivke server.port in dropbox.accessToken. Server.port nam pove na kateri port posluša oz. uporablja Spring boot storitev. Dropbox.accessToken hrani vrednost žetona, ki ga storitev uporabi, da se poveže za dropbox API. Okolijske spremenljivke od podatkovne baze se naložijo iz Kubernetes datoteke. Okolijske spremenljivke, ki se naložijo iz Kubernetes so uporabniško ime, geslo in povezava do podatkovne baze. Dropbox.accessToken se lahko v aplikaciji prenese iz konfiguracijskega strežnika, ki prebere vrednost iz datoteke, ki se nahaja na zasebnem Github repozotoriju.

Toleranca napak

Za toleranco napak sem implementiral circuit braker, reatry in time out. Tolerance se izvedejo na komunikaciji med RSO\_DRAW in RSO\_APP. Reatry poskusi trikrat ponoviti komunikacijo. V primeru treh neuspešnih klicev pokliče fallback metodo, ki vrne privzeto vrednost. Time out med komunikacijo se zgodi po 50000ms. Braker po treh neuspešnih poskusih preneha klicati komunikacijo funkcijo oz. v primeru, da je funkcija ponovno poklican vrne, da storitev ni na voljo oz. vrne napako 503.