

# **Sistem za upravljanje investicionim fondom**

# Uvod

Osnovni ciljevi projekta su:

- implementacija veb servisa koji čine dati sistem;
- pokretanje sistema pomoću alata za orkestriranje kontejnera;

Sistem je namenjen za višekorisnički rad i stoga treba pažljivo da bude dizajniran za potrebe ispravnog i efikasnog rada. Deo funkcionalnosti sistema je javno dostupan, dok je deo funkcionalnosti obezbeđen samo za korisnike koji mogu da se prijave na sistem.

Sistem je potrebno implementirati korišćenjem **Python** programskog jezika, **Flask** radnog okvira i **SQLAlchemy** i **PyMongo** biblioteka. Prilikom obrade zahteva korisnika neophodno je koristiti **SQLAlchemy** i **PyMongo** u što većoj meri, odnosno realizovati obradu pomoću upita svugde gde je to moguće. Pored koda, potrebno je priložiti datoteke na osnovu kojih se kreiraju **Docker Image** artefakti koji predstavljaju delove sistema i koji se mogu iskoristiti za pokretanje odgovarajućih kontejnera. Pored implementacije, potrebno je napisati konfiguracioni fajl pomoću kojeg se ceo sistem može pokrenuti korišćenjem **Kubernetes** alata.

## Konceptualni opis sistema

Investicioni fond se bavi kupovinom i prodajom imovine. Sistem treba da obezbedi registraciju zaposlenih. Zaposleni mogu da vrše pretragu i pregled imovine koju fond poseduje ili je posedovao. Zaposleni takođe mogu da predlože kupovinu nove imovine ili prodaju postojeće. Direktor odobrava kupovinu ili prodaju imovine i može da kreira izveštaje o radu fonda. Prilikom pokretanja sistema potrebno je unapred obezbediti nalog direktoru fonda.

Svaki korisnik treba da bude registrovan pre korišćenja sistema. Za svakog korisnika se u okviru njegovog korisničkog naloga čuvaju sledeće informacije: imejl adresa i lozinka koje se koriste prilikom prijave, ime, prezime i uloga korisnika. Korisnik može imati ulogu direktora ili zaposlenog.

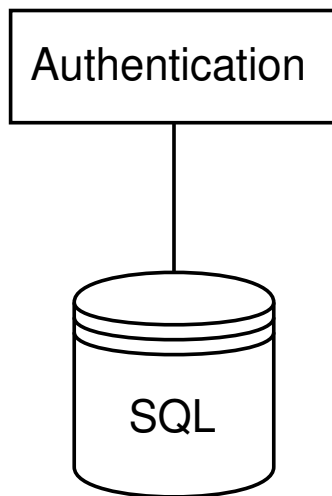
Za imovinu se pamte sledeće informacije: ime, kategorija kojoj imovina pripada (može ih biti više), kupovna cena, datum kupovine, prodajna cena i datum prodaje. Pored ovih informacija, svaka imovina može da ima i dodatne informacije koje su specifične za samu imovinu.

## Funkcionalni opis sistema

Sistem se sastoji iz dva dela: jedan namenjen za upravljanje korisničkim nalogima i jedan namenjen za upravljanje investicionim fondom. I jedan i drugi deo sistema se sastoje iz nekoliko komponenti koje su realizovane pomoću kontejnera kreiranih na osnovu **Docker Image** šablona. Deo ovih **Docker Image** šablona već postoji i nalazi se u okviru javnog repozitorijuma na adresi <https://hub.docker.com/>. Ostatak je neophodno implementirati. U nastavku je dat funkcionalni opis sistema.

# Upravljanje korisničkim nalogima

Izgled dela sistema koji je zadužen za upravljanje korisničkim nalogima dat je na slici 1.



Slika 1. Izgled dela sistema za upravljanje korisničkim nalogima

Ovaj deo sistema se sastoji iz jednog kontejnera i relacione (SQL) baze podataka u kojoj se nalaze samo podaci o korisnicima. Kontejner predstavlja veb servis pomoću kojeg korisnik može da se registruje i dobije [JSON veb token](#) sa kojim će moći da pristupi ostatku sistema.

Opis funkcionalnosti koje pruža ovaj veb servis dat je u nastavku. Svaka adresa je relativna u odnosu na IP adresu kontejnera i broj porta na kojem kontejner sluša.

- **Registracija korisnika**

Adresa /register

Tip POST

Zaglavlje -

Telo Telo zahteva je JSON objekat sledećeg formata:

```
{
  "forename": ".....",
  "surname": ".....",
  "email": ".....",
  "password": "....."
}
```

Sva polja su obavezna i njihov sadržaj je sledeći:

- **forename**: string od najviše 256 karaktera koji predstavlja ime korisnika;
- **surname**: string od najviše 256 karaktera koji predstavlja prezime korisnika;
- **email**: string od najviše 256 karaktera koji predstavlja imejl adresu korisnika;
- **password**: string od najviše 256 karaktera koji predstavlja lozinku korisnika, dužina lozinke mora biti 8 ili više znakova;

Odgovor Ukoliko su svi traženi podaci prisutni u telu zahteva i ispunjavaju navedena ograničenja, rezultat zahteva je kreiranje novog korisnika sa

ulogom zaposlenog i odgovor sa statusnim kodom 200 bez dodatnog sadržaja.

U slučaju greške, rezultat zahteva je odgovor sa statusnim kodom 400 čiji je sadržaj JSON objekat sledećeg formata:

```
{
  "message": "....."
}
```

Sadržaj polja **message** je:

- **"Field <FIELD\_NAME> is missing."** ukoliko neko od polja nije prisutno ili je vrednost polja string dužine 0, **<FIELD\_NAME>** je ime polja koje je očekivano u telu zahteva;
- **"Invalid email."** ukoliko polje **email** nije odgovarajuće formata;
- **"Invalid password."** ukoliko polje **password** nije odgovarajuće dužine;
- **"Email already exists."** ukoliko u bazi postoji korisnik sa istom imejl adresom;

Odgovarajuće provere se vrše u navedenom redosledu.

- **Prijava korisnika**

Adresa     /login

Tip        POST

Zaglavlje -

Telo        Telo zahteva je JSON objekat sledećeg formata:

```
{
  "email": ".....",
  "password": "....."
}
```

Sva polja su obavezna i njihov sadržaj je sledeći:

- **email**: string od najviše 256 karaktera koji predstavlja imejl adresu korisnika;
- **password**: string od najviše 256 karaktera koji predstavlja lozinku korisnika;

Odgovor    Ukoliko su svi traženi podaci prisutni u telu zahteva i ispunjavaju navedena ograničenja i u bazi podataka postoji korisnik sa navedenim kredencijalima, rezultat zahteva je odgovor sa statusnim kodom 200 čiji je sadržaj JSON objekat sledećeg formata:

```
{
  "accessToken": "....."
}
```

Polje **accessToken** je string koji predstavlja JSON veb token koji se koristi za pristup ostalim funkcionalnostima sistema. Token je validan narednih sat vremena. Lice za koje se token izdaje se identifikuje na osnovu imejl

adrese. Token sadrži dodatna polja čiji sadržaj predstavlja informacije koje je korisnik zadao prilikom registracije (bez lozinke) i indikator uloge. Imena polja su ista kao ona navedena u opisu registracije.

U slučaju greške, rezultat zahteva je odgovor sa statusnim kodom 400 i JSON objektom sledećeg formata:

```
{
  "message": "....."
}
```

Sadržaj polja **message** je:

- **"Field <FIELD\_NAME> is missing."** ukoliko neko od polja nije prisutno ili je vrednost polja string dužine 0, **<FIELD\_NAME>** je ime polja koje je očekivano u telu zahteva;
- **"Invalid email."** ukoliko polje imejl nije odgovarajućeg formata;
- **"Invalid credentials."** ukoliko korisnik ne postoji ili je uneo pogrešnu lozinku;

Odgovarajuće provere se vrše u navedenom redosledu.

- **Brisanje korisnika**

Adresa      /delete

Tip          POST

Zaglavlje   Zaglavlje i njihov sadržaj su:

```
{
  "Authorization": "Bearer <ACCESS_TOKEN>"
}
```

Vrednost **<ACCESS\_TOKEN>** je string koji predstavlja JSON veb token za pristup koji je izdat korisniku koji želi da obriše svoj nalog.

Telo          -

Odgovor Ukoliko su sva tražena zaglavlja prisutna, rezultat zahteva je brisanje korisnika iz baze podataka i odgovor sa statusnim kodom 200 bez dodatnog sadržaja.

U slučaju da zaglavlje nedostaje, rezultat je odgovor sa statusnim kodom 401 i JSON objektom sledećeg formata i sadržaja:

```
{  
    "msg": "Missing Authorization Header"  
}
```

U slučaju greške, rezultat zahteva je odgovor sa statusnim kodom 400 i JSON objektom sledećeg formata:

```
{  
    "message": "....."  
}
```

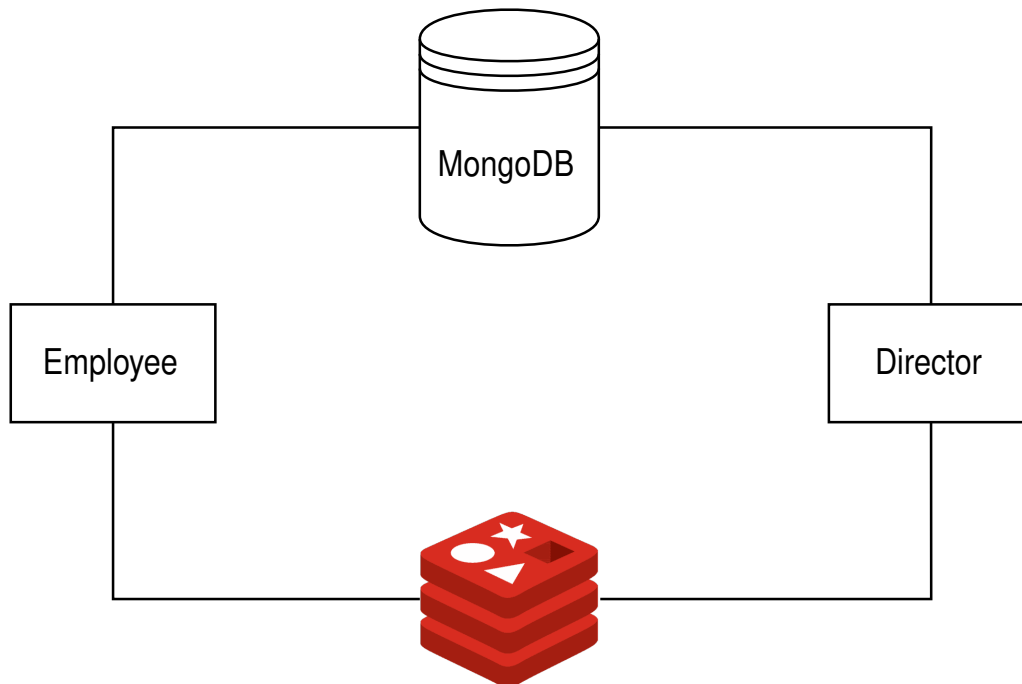
Sadržaj polja **message** je:

- **"Unknown user."** ukoliko korisnik sa datom imejl adresom ne postoji;

Odgovarajuće provere se vrše u navedenom redosledu.

# Upravljanje investicionim fondom

Izgled dela sistema koji je zadužen za upravljanje investicionim fondom dat je na slici 2.



Slika 2. Izgled dela sistema za upravljanje investicionim fondom

Ovaj deo sistema se sastoji iz sledećih komponenti:

- nerelacione baze podataka u kojoj se čuvaju sve informacije neophodne za rad investicionog fonda;
- Redis servisa koji se koristi za čuvanje privremenih informacija neophodnih za rad investicionog fonda;
- kontejnera koji predstavlja veb servis sa funkcionalnostima dostupnim direktorima;
- kontejnera koji predstavlja veb servis sa funkcionalnostima dostupnim zaposlenima;

U nastavku su dati opisi funkcionalnosti svakog kontejnera. Svaka adresa je relativna u odnosu na IP adresu kontejnera i broj porta na kojem kontejner sluša. Svaka od funkcionalnosti zahteva odgovarajući token za pristup.

Funkcionalnosti kontejnera koji je namenjen za rad sa zaposlenima su date u nastavku.

- **Pretraga imovine**

Adresa /search

Tip POST

Zaglavlje Zaglavlje i njihov sadržaj su:

```
{
  "Authorization": "Bearer <ACCESS_TOKEN>"
}
```

Vrednost <ACCESS\_TOKEN> je string koji predstavlja JSON veb token za

pristup koji je izdat zaposlenom prilikom prijave.

### Telo

Telo zahteva je JSON objekat sledećeg formata:

```
{
  "name": "...",
  "category": "...",
  "buying_date": "2026-06-08T22:12:00.000Z",
  "selling_date": "2026-06-08T22:12:00.000Z",
  "info_filters": [
    {
      "field": "field0.field1.field2",
      "operator": "eq",
      "value": 2
    },
    ...
  ]
}
```

Sva polja su opcionalna i, ukoliko su prisutna u telu zahteva, koriste se prilikom formiranja upita. Značenje polja je sledeće:

- **name:** string od najviše 256 karaktera. Ukoliko je prisutan, u rezultatu treba ostaviti samo imovine u čijem se imenu nalazi prosleđeni string;
- **category:** string od najviše 256 karaktera. Ukoliko je prisutan, u rezultatu treba ostaviti samo imovine koje pripadaju zadatoj kategoriji;
- **buying\_date:** string koji predstavlja datum i vreme u ISO 8601 formatu. Ukoliko je prisutan, u rezultatu treba ostaviti samo imovine kupljene nakon ovog datuma;
- **selling\_date:** string koji predstavlja datum i vreme u ISO 8601 formatu. Ukoliko je prisutan, u rezultatu treba ostaviti samo imovine prodate pre ovog datuma. Neprodate imovine ne treba uključiti u rezultat.
- **info\_filters:** niz objekata koji predstavljaju dodatne filtere nad opcionalnim informacijama vezanim za imovinu.

Svaki objekat u nizu **info\_filters** sadrži sledeća polja:

- **field:** string koji predstavlja naziv opcionalnog polja nad kojim se vrši filtriranje. Ukoliko je opcionalno polje ugnježđeno, putanja do njega se navodi korišćenjem tačke kao separatora. Na primer, vrednost **"field0.field1.field2"** označava da se filtriranje vrši nad poljem **field2**, koje se nalazi unutar polja **field1**, a ono unutar polja **field0**. Putanja je relativna u odnosu na **info** polje imovine.
- **operator:** string koji predstavlja validan **MongoDB** operator poređenja koji se koristi prilikom filtriranja.
- **value:** vrednost sa kojom se poredi vrednost polja navedenog u atributu **field**. Tip ove vrednosti zavisi od tipa opcionalnog polja nad kojim se vrši filtriranje.

Ukoliko je zadato više filtera, imovina treba da bude uključena u rezultat samo ako zadovoljava sve navedene uslove.

Odgovor Ukoliko su sva tražena zaglavlja prisutna, rezultat zahteva je odgovor sa statusnim kodom 200 čiji je sadržaj JSON objekat sledećeg formata:

```
{
  "assets": [
    {
      "id": "...",
      "name": "...",
      "categories": ["category0", "category1", ...],
      "buying_date": "2026-06-08T22:12:00.000Z",
      "selling_date": "2026-06-08T22:12:00.000Z",
      "buying_price": 10000,
      "selling_price": 20000,
      "info": {
        "field0": {
          "field1": {
            "field2": 2
          }
        }
      }
    },
    ...
  ]
}
```

Polje **assets** predstavlja niz JSON objekata. Svaki objekat u ovom nizu predstavlja jednu imovinu koja zadovoljava uslove pretrage.

Svaki JSON objekat u nizu **assets** sadrži sledeća polja:

- **id** - string koji predstavlja jedinstveni identifikator dokumenta u MongoDB bazi podataka. Vrednost ovog polja odgovara vrednosti MongoDB **ObjectId** identifikatora;
- **name** - string koji predstavlja naziv imovine;
- **categories** - lista stringova koji predstavljaju kategorije kojima data imovina pripada;
- **buying\_date** - string koji predstavlja datum i vreme kupovine imovine u ISO 8601 formatu;
- **buying\_price** - broj koji predstavlja cenu po kojoj je imovina kupljena;
- **selling\_date** - string koji predstavlja datum i vreme prodaje imovine u ISO 8601 formatu, prisutan je jedino ako je imovina prodana;
- **selling\_price** - broj koji predstavlja cenu po kojoj je imovina prodana, prisutan je jedino ako je imovina prodana;
- **info** - JSON objekat koji sadrži dodatne informacije o imovini. Struktura ovog objekta nije unapred fiksirana i može sadržati proizvoljna dodatna, uključujući i ugnježdjena, polja.

U slučaju da zaglavlje nedostaje, rezultat je odgovor sa statusnim kodom 401 i JSON objektom sledećeg formata i sadržaja:

```
{
  "msg": "Missing Authorization Header"
}
```

- **Formiranje predloga za kupovinu imovine**

Adresa     /create\_buy\_order

Tip        POST

Zaglavlje   Zaglavlje i njihov sadržaj su:

```
{
  "Authorization": "Bearer <ACCESS_TOKEN>"
}
```

Vrednost **<ACCESS\_TOKEN>** je string koji predstavlja JSON veb token za pristup koji je izdat zaposlenom prilikom prijave.

Telo        Telo zahteva je JSON objekat sledećeg formata:

```
{
  "name": "...",
  "categories": ["category0", "category1", ...],
  "buying_price": 10000,
  "info": {...}
}
```

Sva polja su obavezna i njihov sadržaj je sledeći:

- **name** - string od najviše 256 karaktera koji predstavlja naziv imovine koja se kupuje;
- **categories** - lista stringova od najviše 256 karaktera koji predstavljaju kategorije kojima imovina pripada;
- **buying\_price** - broj koji predstavlja cenu po kojoj se imovina kupuje. Vrednost mora biti veća od 0;
- **info** - JSON objekat proizvoljnog formata koji sadrži dodatne informacije o imovini;

Odgovor    Ukoliko su sva tražena zaglavlja prisutna i ukoliko su sva tražena polja prisutna u telu zahteva i odgovarajućeg su tipa, zahtev za kupovinu se upisuje u **Redis** servis radi kasnije obrade. Rezultat zahteva je odgovor sa statusnim kodom 200 bez dodatnog sadržaja.

U slučaju da zaglavlje nedostaje, rezultat je odgovor sa statusnim kodom 401 i JSON objektom sledećeg formata i sadržaja:

```
{
  "msg": "Missing Authorization Header"
}
```

U slučaju da neko polje tela nedostaje ili je nekorektno, rezultat zahteva je odgovor sa statusnim kodom 400 i JSON objektom sledećeg formata:

```
{
  "message": "....."
}
```

Sadržaj polja **message** je:

- **"Field <FIELD\_NAME> is missing."** ukoliko neko od polja nije prisutno ili je vrednost polja string dužine 0, **<FIELD\_NAME>** je ime polja koje je očekivano u telu zahteva;
- **"Categories list is empty."** ukoliko je lista kategorija prazna;
- **"Invalid buying price."** ukoliko polje **buying\_price** ne predstavlja broj ili je u pitanju broj 0 ili manji.

Odgovarajuće provere se vrše u navedenom redosledu.

- **Formiranje zahteva za prodaju imovine**

Adresa     /create\_sell\_order

Tip        POST

Zaglavlje   Zaglavlje i njihov sadržaj su:

```
{
  "Authorization": "Bearer <ACCESS_TOKEN>"
}
```

Vrednost **<ACCESS\_TOKEN>** je string koji predstavlja JSON veb token za pristup koji je izdat zaposlenom prilikom prijave.

Telo        Telo zahteva je JSON objekat sledećeg formata:

```
{
  "id": "...",
  "selling_price": 20000
}
```

Sva polja su obavezna i njihov sadržaj je sledeći:

- **id** - string koji predstavlja jedinstveni identifikator imovine u MongoDB bazi podataka. Vrednost ovog polja odgovara vrednosti MongoDB **ObjectId** identifikatora;
- **selling\_price** - broj koji predstavlja cenu po kojoj se predlaže prodaja imovine. Vrednost mora biti veća od 0;

Odgovor Ukoliko su sva tražena zaglavlja prisutna i ukoliko su sva tražena polja prisutna u telu zahteva i odgovarajućeg su formata, zahtev za prodaju se upisuje u Redis servis radi kasnije obrade. Rezultat zahteva je odgovor sa statusnim kodom 200 bez dodatnog sadržaja.

U slučaju da zaglavlje nedostaje, rezultat je odgovor sa statusnim kodom 401 i JSON objektom sledećeg formata i sadržaja:

```
{
  "msg": "Missing Authorization Header"
}
```

U slučaju da neko polje tela nedostaje ili je nekorektno, rezultat zahteva je odgovor sa statusnim kodom 400 i JSON objektom sledećeg formata:

```
{
  "message": "....."
}
```

Sadržaj polja **message** je:

- **"Field <FIELD\_NAME> is missing."** ukoliko neko od polja nije prisutno ili je vrednost polja string dužine 0, **<FIELD\_NAME>** je ime polja koje je očekivano u telu zahteva;
- **"Invalid id."** ukoliko polje **id** ne predstavlja validan MongoDB **ObjectId** identifikator ili u bazi ne postoji imovina sa datim identifikatorom;
- **"Invalid selling price."** ukoliko polje **selling\_price** ne predstavlja broj ili je u pitanju broj 0 ili manji.

Odgovarajuće provere se vrše u navedenom redosledu.

Funkcionalnosti kontejnera koji je namenjen za rad sa direktorom su date u nastavku.

- **Pregled zahteva**

Adresa /pending\_orders

Tip GET

Zaglavlje Zaglavlje i njihov sadržaj su:

```
{
  "Authorization": "Bearer <ACCESS_TOKEN>"
}
```

Vrednost **<ACCESS\_TOKEN>** je string koji predstavlja JSON veb token za pristup koji je izdat direktoru prilikom prijave.

Telo -

Odgovor Ukoliko su sva tražena zaglavlja prisutna, rezultat zahteva je odgovor sa

statusnim kodom 200 čiji je sadržaj JSON objekat sledećeg formata:

```
{
  "orders": [
    {
      "uuid": "550e8400-e29b-41d4-a716-446655440000",
      "order_type": "BUY",
      "name": "...",
      "categories": [
        "category0",
        "category1"
      ],
      "info": {...},
      "buying_price": 10000
    },
    {
      "uuid": "550e8400-e29b-41d4-a716-446655440001",
      "order_type": "SELL",
      "id": "66688f0f4b6f2d6a2f7c9a11",
      "selling_price": 20000
    },
    ...
  ]
}
```

Polje **orders** predstavlja niz JSON objekata. Svaki objekat predstavlja jedan zahtev koji čeka na odobrenje.

Svaki JSON objekat u nizu **orders** sadrži sledeća polja:

- **uuid** – string koji predstavlja jedinstveni identifikator zahteva u Redis servisu;
- **order\_type** – string koji predstavlja tip zahteva. Vrednost može biti **"BUY"** za zahtev za kupovinu ili **"SELL"** za zahtev za prodaju.

Ukoliko je vrednost polja **order\_type** jednaka **"BUY"**, objekat dodatno sadrži sledeća polja:

- **name** – string koji predstavlja naziv imovine koja se kupuje;
- **categories** – lista stringova koji predstavljaju kategorije kojima imovina pripada;
- **info** – JSON objekat proizvoljnog sadržaja koji predstavlja dodatne informacije o imovini;
- **buying\_price** – broj koji predstavlja cenu po kojoj se imovina kupuje.

Ukoliko je vrednost polja **order\_type** jednaka **"SELL"**, objekat dodatno sadrži sledeća polja:

- **id** – string koji predstavlja jedinstveni identifikator imovine u MongoDB bazi podataka. Vrednost ovog polja odgovara vrednosti MongoDB **ObjectId** identifikatora;
- **selling\_price** – broj koji predstavlja cenu po kojoj se predlaže

prodaja imovine.

U slučaju da zaglavlje nedostaje, rezultat je odgovor sa statusnim kodom 401 i JSON objektom sledećeg formata i sadržaja:

```
{  
  "msg": "Missing Authorization Header"  
}
```

- **Odlučivanje o ishodu zahteva**

Adresa     /decision

Tip        POST

Zaglavlje   Zaglavlje i njihov sadržaj su:

```
{  
  "Authorization": "Bearer <ACCESS_TOKEN>"  
}
```

Vrednost **<ACCESS\_TOKEN>** je string koji predstavlja JSON veb token za pristup koji je izdat direktoru prilikom prijave.

Telo        Telo zahteva je JSON objekat sledećeg formata:

```
{  
  "uuid": "...",  
  "approved": True  
}
```

Sva polja su obavezna i njihov sadržaj je sledeći:

- **uuid** – string koji predstavlja jedinstveni identifikator zahteva u Redis servisu koji treba odobriti.
- **approved** – logička vrednost koja određuje da li je zahtev prihvaćen ili odbijen.

Odgovor Ukoliko su sva tražena zaglavlja prisutna i sva polja u telu zahteva prisutna i odgovarajućeg su formata, zahtev sa prosleđenim identifikatorom se obrađuje i uklanja iz Redis servisa.

Ukoliko je odobren zahtev za kupovinu, u MongoDB bazu podataka se upisuje nova imovina. Za novu imovinu se čuvaju naziv, kategorije, cena kupovine, datum kupovine i dodatne informacije. Datum kupovine se postavlja na trenutak odobravanja zahteva.

Ukoliko je odobren zahtev za prodaju, u MongoDB bazi podataka se ažurira postojeća imovina. Za datu imovinu se postavljaju cena prodaje i datum prodaje. Datum prodaje se postavlja na trenutak odobravanja zahteva.

Ukoliko je bilo koji tip zahteva odbijen, on se briše iz Redis servisa.

Rezultat uspešne obrade je odgovor sa statusnim kodom 200 bez dodatnog sadržaja.

U slučaju da zaglavlje nedostaje, rezultat je odgovor sa statusnim kodom 401 i JSON objektom sledećeg formata i sadržaja:

```
{
  "msg": "Missing Authorization Header"
}
```

U slučaju da neko polje tela nedostaje ili je nekorektno, rezultat zahteva je odgovor sa statusnim kodom 400 i JSON objektom sledećeg formata:

```
{
  "message": "....."
}
```

Sadržaj polja **message** je:

- **"Field uuid is missing."** ukoliko polje **uuid** nije prisutno ili je vrednost polja string dužine 0;
- **"Invalid uuid."** ukoliko polje **uuid** nije string odgovarajućeg UUID formata ili ukoliko u Redis servisu ne postoji zahtev sa prosleđenim identifikatorom.
- **"Field approved is missing."** ukoliko polje **approved** nije prisutno;
- **"Invalid decision."** ukoliko polje **approved** nije logička vrednost.

Odgovarajuće provere se vrše u navedenom redosledu.

- **Pregled poslovanja fonda**

Adresa /report

Tip GET

Zaglavlje Zaglavlje i njihov sadržaj su:

```
{
  "Authorization": "Bearer <ACCESS_TOKEN>"
}
```

Vrednost **<ACCESS\_TOKEN>** je string koji predstavlja JSON veb token za pristup koji je izdat direktoru prilikom prijave.

Telo

-

Odgovor

Ukoliko su sva tražena zaglavlja prisutna, rezultat zahteva je odgovor sa statusnim kodom 200 čiji je sadržaj JSON objekat sledećeg formata:

```
{
  "statistics": [
    {
      "category": "category0",
      "spent": 10000,
      "earned": 15000
    },
    {
      "category": "category1",
      "spent": 20000,
      "earned": 26000
    },
    ...
  ]
}
```

Polje **statistics** predstavlja niz JSON objekata. Svaki objekat predstavlja statistiku prodaje za jednu kategoriju.

Svaki JSON objekat u nizu **statistics** sadrži sledeća polja:

- **category** – string koji predstavlja naziv kategorije;
- **spent** – ukupan iznos potrošen za kupovinu imovine koja pripada datoj kategoriji;
- **earned** – ukupan iznos zarađen prodajom imovine koja pripada datoj kategoriji;

U obračun zarade ulazi samo imovine koje su prodate, odnosno imovina koja ima definisanu cenu prodaje i datum prodaje.

Ukoliko jedna imovina pripada većem broju kategorija, ona se računa u statistiku svake od tih kategorija. To znači da se njena kupovna cena dodaje vrednosti **spent** za svaku kategoriju kojoj pripada, a njena prodajna cena dodaje vrednosti **earned** za svaku kategoriju kojoj pripada.

Niz **statistics** treba da bude sortiran opadajuće po vrednosti polja **earned**, zatim rastuće po vrednosti polja **spent** i na kraju rastuće po imenu kategorije.

U slučaju da zaglavlje nedostaje, rezultat je odgovor sa statusnim kodom 401 i JSON objektom sledećeg formata i sadržaja:

```
{
  "msg": "Missing Authorization Header"
}
```



# Pokretanje sistema

Potrebno je napisati konfiguracioni fajl pomoću kojeg se sistem može pokrenuti korišćenjem Kubernetes alata. Sve informacije neophodne za rad kontejnera, kao što su adrese servisa, URL baze podataka, nazivi baza, korisnička imena, lozinke i ostali parametri, potrebno je proslediti kontejnerima putem varijabli okruženja.

Svi konfiguracioni podaci moraju biti izdvojeni u Kubernetes **ConfigMap** objektima, kako bi se omogućilo jednostavno podešavanje i izmena konfiguracije. Sve lozinke i ostale osetljive podatke izdvojiti u komponentu **Secret**.

Prilikom pokretanja sistema potrebno je obezbediti sledeće:

- automatsku inicijalizaciju relacione baze podataka, koja podrazumeva kreiranje svih neophodnih tabela i dodavanje jednog početnog direktora sa sledećim informacijama:

```
{  
  "forename": "Scrooge",  
  "surname": "McDuck",  
  "email": "onlymoney@gmail.com",  
  "password": "evenmoremoney"  
}
```

- trajno čuvanje podataka u bazama podataka, tako da podaci ostanu sačuvani i nakon prestanka rada ili ponovnog pokretanja odgovarajućih kontejnera;
- pokretanje servisa koji obrađuje zahteve zaposlenih u tri replike, kako bi se omogućila veća dostupnost i paralelna obrada zahteva;

Na samoj odbrani, od studenata se očekuje da pomoću ovog fajla pokrenu sistem i demonstriraju njegov rad.

# Odobrovanje zahteva glasanjem

Potrebno je obezbediti da zaposleni mogu da glasaju za ili protiv zahteva za prodaju ili kupovinu. Glasanje je potrebno implementirati pomoću pametnih ugovora **Ethereum Blockchain** platforme. Za simulaciju ove platforme potrebno je iskoristi **Docker Image** artefakt dostupan na [Docker Hub \(https://hub.docker.com/r/trufflesuite/ganache-cli/\)](https://hub.docker.com/r/trufflesuite/ganache-cli/) repozitorijumu.

Kada direktor odobri zahtev potrebno je kreirati jedan objekat pametnog ugovora koji će dozvoliti zaposlenima da glasaju za ili protiv tog zahteva. Samo unapred odabrani zaposleni mogu da glasaju i to samo jednom po pametnom ugovoru. Zahtev se smatra odobrenim ukoliko za njega glasa većina unapred definisanih glasača, odnosno najmanje  $n / 2 + 1$  glasača, gde je n ukupan broj adresa kojima je dozvoljeno glasanje.

Prilikom realizacije potrebno je izmeniti funkcionalnost kojom se odlučuje o ishodu zahteva koja se nalazi u kontejneru koji je namenjen za rad sa direktorom:

- **Odobrovanje zahteva**

Adresa     /decision

Tip         POST

Zaglavlje   Zaglavlje i njihov sadržaj su:

```
{
  "Authorization": "Bearer <ACCESS_TOKEN>"
}
```

Vrednost **<ACCESS\_TOKEN>** je string koji predstavlja JSON veb token za pristup koji je izdat direktoru prilikom prijave.

Telo         Telo zahteva je JSON objekat sledećeg formata:

```
{
  "uuid": "550e8400-e29b-41d4-a716-446655440000",
  "voters": [
    "0x1111111111111111111111111111111111111111111111111111111111111111",
    "0x2222222222222222222222222222222222222222222222222222222222222222",
    "0x3333333333333333333333333333333333333333333333333333333333333333",
    ...
  ]
}
```

Sva polja su obavezna i njihov sadržaj je sledeći:

- **uuid** – string koji predstavlja jedinstveni identifikator zahteva u Redis servisu koji treba odobriti.
- **voters** – lista stringova koji predstavljaju validne adrese Ethereum računa kojima je dozvoljeno glasanje nad datim zahtevom.

Odgovor     Ukoliko su sva tražena zaglavlja prisutna i ukoliko su sva tražena polja prisutna u telu zahteva i odgovarajućeg su formata, za zahtev sa

prosleđenim identifikatorom kreira se pametni ugovor na Ethereum blockchain platformi. Naknadu za kreiranje ugovora platiti sa nekog od računa koji se kreiraju prilikom pokretanja simulatora.

Pametni ugovor implementira mehanizam većinskog glasanja. Ugovor mora da sadrži listu dozvoljenih glasača i mora da obezbedi sledeće:

- samo Ethereum računi čije se adrese nalaze u listi **voters** mogu da glasaju;
- svaki dozvoljeni glasač može da glasa najviše jednom;
- glas može biti glas za potvrdu zahteva ili glas za odbijanje zahteva;
- broj glasača mora biti neparan;
- zahtev se smatra prihvaćenim kada broj glasova za potvrdu dostigne većinu, odnosno  $n / 2 + 1$ , gde je n broj dozvoljenih glasača;
- zahtev se smatra odbijenim kada broj glasova za odbijanje dostigne većinu, odnosno  $n / 2 + 1$ .

Rezultat uspešnog zahteva je odgovor sa statusnim kodom 200 čiji je sadržaj JSON objekat sledećeg formata:

```
{
  "approve_transaction": {...},
  "reject_transaction": {...}
}
```

Polje **approve\_transaction** predstavlja JSON objekat koji predstavlja transakciju kojom glasač glasa za potvrdu zahteva.

Polje **reject\_transaction** predstavlja JSON objekat koji predstavlja transakciju kojom glasač glasa za odbijanje zahteva.

Glasanje se izvršava slanjem jedne od vraćenih transakcija sa Ethereum računa koji se nalazi u listi dozvoljenih glasača. Ukoliko transakciju pokuša da pošalje račun koji nije u listi **voters**, pametni ugovor mora da odbije izvršavanje transakcije uz poruku **"Invalid address."**

Zaposleni glasaju u nekom trenutku koji nije unapred poznat. Kada pametni ugovor prikupi potrebnu većinu glasova za potvrdu, zahtev se smatra odobrenim. Tada se zahtev uklanja iz Redis servisa i evidentira u MongoDB bazi podataka. Ukoliko se zahtev odbije on se samo uklanja iz Redis servisa.

Nakon ovog trenutka glasanje je završeno i svaki pokušaj glasanja nakon ovog trenutka treba da se odbije uz poruku **"Voting ended."**

Ukoliko je odobren zahtev za kupovinu, u MongoDB bazu podataka se upisuje nova imovina. Za novu imovinu se čuvaju naziv, kategorije, cena kupovine, datum kupovine i dodatne informacije o imovini. Datum kupovine se postavlja na trenutak odobravanja zahteva.

Ukoliko je odobren zahtev za prodaju, u MongoDB bazi podataka se ažurira postojeća imovina. Za datu imovinu se postavljaju cena prodaje i datum prodaje. Datum prodaje se postavlja na trenutak odobravanja zahteva.

U slučaju da zaglavlje nedostaje, rezultat je odgovor sa statusnim kodom 401 i JSON objektom sledećeg formata i sadržaja:

```
{  
  "msg": "Missing Authorization Header"  
}
```

U slučaju da neko polje tela nedostaje ili je nekorektno, rezultat zahteva je odgovor sa statusnim kodom 400 i JSON objektom sledećeg formata:

```
{  
  "message": "....."  
}
```

Sadržaj polja **message** je:

- **"Field uuid is missing."** ukoliko polje **uuid** nije prisutno ili je vrednost polja string dužine 0;
- **"Invalid uuid."** ukoliko polje **uuid** nije string odgovarajućeg UUID formata ili ukoliko u Redis servisu ne postoji zahtev sa prosleđenim identifikatorom;
- **"Field voters is missing."** ukoliko polje **voters** nije prisutno ili ukoliko je lista prazna;
- **"Invalid voter address."** ukoliko makar jedna vrednost iz liste **voters** ne predstavlja validnu adresu Ethereum računa.
- **"Even number of voters."** ukoliko je broj elemenata liste **voters** paran.

Odgovarajuće provere se vrše u navedenom redosledu.