

Zadanie

Systém uchováva informácie o poistencoch a ich zmluvách v SQL databáze. Systém musí zabezpečiť aby sa do databázy ukladali len validné a úplné dáta. Pokiaľ nie je uvedené inak sú všetky dáta povinné. Na testovanie som používal MySQL databázu.

O každom poistencovi systém eviduje nasledovné dáta: unikátny číselný identifikátor, meno, priezvisko, rodné číslo, email, adresa trvalého pobytu, korešpondenčná adresa, zoznam zmlúv.

Pokiaľ korešpondenčná adresa poistenca nie je definovaná, použije sa jeho adresa trvalého pobytu. Adresa sa skladá z PSČ, názvu obce, názvu ulice a orientačného čísla domu. Systém eviduje dva typy zmlúv: poistenie majetku a cestovné poistenie. Každá zmluva má priradené unikátne číslo zmluvy a dátum vzniku. Zmluvy poistenia majetku obsahujú informácie o type nehnuteľnosti, adrese nehnuteľnosti a hodnote nehnuteľnosti v eurách. Typ nehnuteľnosti je číselníková hodnota, povolené hodnoty sú: "Byt", "Rodinný dom - murovaný", "Rodinný dom - drevený". Zmluvy cestovného poistenia obsahujú dátumy začiatku a konca poistenia, či bola poistená zodpovednosť za škodu, či bol poistený úraz. Systém publikuje nasledovné REST webové služby:

- zoznam poistencov
 - metóda: GET,
 - url: /insured
 - response: zoznam poistencov zoradený vzostupne podľa priezviska, jednotlivé objekty poistencov neobsahujú zoznam zmlúv
- zaevidovanie nového poistenca
 - metóda: POST,
 - url: /insured,
 - body: informácie o evidovanom poistencovi
 - response: v prípade úspešného zaevidovania vráti vygenerovaný číselný identifikátor poistenca
- detail poistenca,
 - metóda: GET,
 - url: /insured/{id poistenca}
 - response: objekt poistenca ktorý obsahuje aj zoznam zmlúv

Projekt je možné vytvoriť pomocou [Spring Initializr](#) azvolením závislostí: [Spring Web](#), [H2 Database](#), [Spring Data JPA](#).

Konfigurácie

Na pripojenie databázy som používal MySQL.

username: root

password: root

Port: 3306

spring.datasource.url=jdbc:mysql://localhost:3306/poistenie?useSSL=false

spring.datasource.username=root

spring.datasource.password=root

#Hibernate properties

#dialect is for generate sql queries

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL5Dialect

spring.jpa.hibernate.ddl-auto=update

Riešenie

Architektúra programu, je rozdelená na balíky: controller, model, repository a service.

Balík *controller*

Obsahuje triedu *PoistenecController*. Táto trieda je zodpovedná za všetky http requesty. Obsahuje funkcie na POST (uloženie nového poistenca do databázy) na GET (vyhľadávanie všetkých alebo

konkrétneho poistenca podľa ID).

```
@PostMapping
public ResponseEntity savePoistenec(@RequestBody Poistenec poistenec) {
    List<String> errors = new ArrayList<>();
    errors = poistenecService.checkJson(poistenec);

    ResponseEntity response;
    if (errors.isEmpty()) {
        Poistenec saved = poistenecService.savePoistenec(poistenec);
        response = new ResponseEntity(saved.getId(), HttpStatus.OK);
    } else
        response = new ResponseEntity(errors, HttpStatus.UNPROCESSABLE_ENTITY);

    return response;
}
```

Pokiaľ chceme uložiť nového poistenca, musíme zadať všetky údaje poistenca. Pokiaľ niektoré údaje vynecháme, tak poistenec sa neuloží a API vráti response, ktorý obsahuje zoznam chýbajúcich dát, ktoré treba doplniť. Ak všetky údaje boli správne zadané, tak API vráti ID uloženého poistenca.

Logiku a validáciu som spravil triede *PoistenecServiceImpl*.

```
public List<String> checkJson(Poistenec poistenec){
    List<String> errors = new ArrayList<String>();

    if (poistenec.getMeno().length() == 0)
        errors.add("Name has to be filled! ");
    if (poistenec.getPriezvisko().length() == 0)
        errors.add("Last name has to be filled! ");
    if (poistenec.getEmail().length() == 0 || poistenec.getEmail().contains("@") == false)
        errors.add("Email has to be filled! ");
    if (poistenec.getRc().toString().length() != 10)
        errors.add("RC has to have 10 digits! ");
    if (poistenec.getObec().length() == 0)
        errors.add("Settlement has to be filled! ");
    if (poistenec.getUlica().length() == 0)
        errors.add("Street has to be filled! ");
    if (poistenec.getPsc().toString().length() != 5)
        errors.add("PSC has to have 5 digits! ");
    if (poistenec.getOrientacneCislo().toString().length() == 0)
        errors.add("Street number has to be filled! ");

    return errors;
}
```

Táto funkcia kontroluje, spraví jednoduchú validáciu, či povinné

dáta sú zadané alebo nie. Na konci vráti zoznam chýb ktorý je vrátený ako *http response*. Na začiatku som chcel používať *inbuilt* validačnú funkcionality ktorú ponúka *spring* a *hibernate*, ale z nejakého dôvodu nefungovali. Potrebné *dependency anotácie* som implementoval. Tieto validačné podmienky som nechal v modeloch ako komentár.

```
public void checkKorespondencna(Poistenec poistenec){  
  
    if (poistenec.getKorespondencnaAdresa() == null) {  
        KorespondencnaAdresa korespondencnaAdresa = new KorespondencnaAdresa();  
        korespondencnaAdresa.setObec(poistenec.getObec());  
        korespondencnaAdresa.setPsc(poistenec.getPsc());  
        korespondencnaAdresa.setUlica(poistenec.getUlica());  
        korespondencnaAdresa.setOrientacneCisloDomu(poistenec.getOrientacneCislo());  
        korespondencnaAdresaRepository.save(korespondencnaAdresa);  
        poistenec.setKorespondencnaAdresa(korespondencnaAdresa);  
    }  
}  
  
@Override  
public Poistenec savePoistenec(Poistenec poistenec) {  
  
    checkKorespondencna(poistenec);  
    return poistenecRepository.save(poistenec);  
}
```

Pred uložením nového poistenca do databázy, kontrolujem či má zadanú korešpondenčnú adresu. Ak nie, tak vytvorím novú korešpondenčnú adresu podľa trvalého pobytu poistenca.

Definoval som nasledovné modely: Poistenec, KorešpondenčnáAdresa, PoistenieCestovne, PoistenieMajetku.

```

@Data
@Entity
@Table(name = "poisteneci")
public class Poistenec {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id_poistenec")
    private Long id;

    // @Pattern(regexp = "[a-zA-Z]+", message = "Name can contain only letters!")
    @NotBlank(message = "name required!")
    private String meno;

    // @Pattern(regexp = "[a-zA-Z]+", message = "Lastname can contain only letters!")
    // @Column(name = "Priezvisko", nullable = false)
    @NotBlank(message = "LastName required!")
    private String priezvisko;

    // @Pattern(regexp = "^[\\w-\\.]+@([\\w-]+\\.){1,4}$", message = "must have valid email format")
    // @Column(name = "Email", nullable = false, unique = true)
    @NotBlank(message = "email required!")
    private String email;

    // @Pattern(regexp="^[\\d]{10}$", message="RC should have 6 digits!")
    @Column(name = "Rc", nullable = false, unique = true)
    @NotNull
    private Long rc;
}

```

Testovacie vstupy:

1.

```

{
  "meno": "David",
  "priezvisko": "Cibula",
  "email": "cibula@gmail.com",
  "rc": 2323212345,
  "psc": 12345,
  "obec": "Sala",
  "ulica": "Hviezdoslavova",
  "orientacneCislo": 33,
  "poisteniaMajetku":
    [{
      "datumVzniku": "07/06/2023",

```

```
"hodnota":123,
"typNehnutelnosti":"Rodinny_dom_murovany",
"psc":22956,
"obec": "Kosice",
"ulica": "Sturova",
"orientacne_domu": 2
},
{
"datumVzniku":"01/16/2020",
"hodnota":555555,
"typNehnutelnosti":"Rodinny_dom_dreveny",
"psc":11111,
"obec": "Banska Bystrica",
"ulica": "Sturova",
"orientacne_domu":12
},
{
"datumVzniku":"11/05/2023",
"hodnota":985963,
"typNehnutelnosti":"Byt",
"psc":12356,
"obec": "Losice",
"ulica": "Krajna",
"orientacne_domu": 30
}},
"korespondencnaAdresa":
{
"psc":94444,
"obec": "Presov",
"ulica": "Hlavna",
"orientacneCisloDomu": 12
```

```
}  
}
```

2.

```
{  
  "meno": "David",  
  "priezvisko": "Sabo",  
  "email": "sabo@gmail.com",  
  "rc": 1234567888,  
  "psc": 11111,  
  "obec": "Saľa",  
  "ulica": "Hviezdoslavova",  
  "orientacneCislo": 33,  
  "poisteniaMajetku":  
    [{  
      "datumVzniku": "07/06/2023",  
      "hodnota": 123,  
      "typNehnutelnosti": "Byt",  
      "psc": 22956,  
      "obec": "Trnava",  
      "ulica": "Hlavna",  
      "orientacne_domu": 2  
    }],  
  "korespondencnaAdresa":  
    {  
      "psc": 94444,  
      "obec": "Prešov",  
      "ulica": "Hlavna",  
      "orientacneCisloDomu": 2  
    }  
}
```

3.

```
{  
  "meno": "Adam",  
  "priezvisko": "Zabzell",  
  "email": "zabzel@gmail.com",  
  "rc": 5612354447,  
  "psc": 94444,  
  "obec": "Bratislava",  
  "ulica": "Hlavne namestie",  
  "orientacneCislo": 5,  
  "poisteniaMajetku":  
    [{  
      "datumVzniku": "02/03/2023",  
      "hodnota": 12325,  
      "typNehnutelnosti": "Byt",  
      "psc": 15956,  
      "obec": "Bratislava",  
      "ulica": "Hlavna",  
      "orientacne_domu": 7  
    }]  
}
```


4.

```
{  
  "meno": "Matej",  
  "priezvisko": "Kukučín",  
  "email": "matej@gmail.com",  
  "rc": 2782631125,  
  "psc": 93005,  
  "obec": "Gabčíkovo",  
  "ulica": "Letná",  
  "orientacneCislo": 32  
}
```

5.

```
{  
  "meno": "Andrej",  
  "priezvisko": "Sládkovič",  
  "email": "marína@gmail.com",  
  "rc": 2782631111,  
  "psc": 93005,  
  "obec": "Senec",  
  "ulica": "Jarná",  
  "orientacneCislo": 12  
  "cestovnePoistenia":  
    [{  
      "zaciatok": "02/03/2023" ,  
      "koniec": "02/03/2024",  
      "typPoistenia": "zodpovednosť za škodu"  
    }]  
}
```

Výstupy

Poistenci

1	marína@gmail.com	Andrej	Senec	12	Sládkovič	93005	2782631111	Jarná	1
2	matej@gmail.com	Matej	Gabčikovo	32	Kukučín	93005	2782631125	Letná	2
3	zabzel@gmail.com	Adam	Bratislava	5	Zabzell	94444	5612354447	Hlavné námestie	3
4	sabo@gmail.com	David	Saľa	33	Sabo	11111	1234567888	Hviezdoslavova	4
5	cibula@gmail.com	David	Sala	33	Cibula	12345	2323212345	Hviezdoslavova	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

poisteniaMajetku

1	02/03/2023	12325	Bratislava	7	15956	Byt	Hlavná	3
2	07/06/2023	123	Trnava	2	22956	Byt	Hlavná	4
3	07/06/2023	123	Kosice	2	22956	Rodinný dom murovaný	Sturova	5
4	01/16/2020	555555	Banská Bystrica	12	11111	Rodinný dom drevený	Sturova	5
5	11/05/2023	985963	Losice	30	12356	Byt	Krajná	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Korešpondenčná adresa

id	obec	orientacne_domu	psc	ulica
1	Senec	12	93005	Jarná
2	Gabčikovo	32	93005	Letná
3	Bratislava	5	94444	Hlavné námestie
4	Prešov	2	94444	Hlavná
5	Presov	12	94444	Hlavná
NULL	NULL	NULL	NULL	NULL

Cestovne poistenia

id_cestovne_poistenie	koniec	typ_poistenia	zaciatok	fk_poistenec_id
1	02/03/2024	zodpovednosť za škodu	02/03/2023	1
NULL	NULL	NULL	NULL	NULL

Výstup na GET všetkých poistencov

```
[
  {
    "id": 5,
    "meno": "David",
    "priezvisko": "Cibula",
    "email": "cibula@gmail.com",
    "rc": 2323212345,
    "psc": 12345,
    "obec": "Sala",
    "ulica": "Hviezdoslavova",
    "orientacneCislo": 33,
```

```
"poisteniaMajetku": [
  {
    "id": 3,
    "datumVzniku": "07/06/2023",
    "hodnota": 123,
    "typNehnutelnosti": "Rodinny_dom_murovany",
    "psc": 22956,
    "obec": "Kosice",
    "ulica": "Sturova",
    "orientacne_domu": 2
  },
  {
    "id": 4,
    "datumVzniku": "01/16/2020",
    "hodnota": 555555,
    "typNehnutelnosti": "Rodinny_dom_dreveny",
    "psc": 11111,
    "obec": "Banska Bystrica",
    "ulica": "Sturova",
    "orientacne_domu": 12
  },
  {
    "id": 5,
    "datumVzniku": "11/05/2023",
    "hodnota": 985963,
    "typNehnutelnosti": "Byt",
    "psc": 12356,
    "obec": "Losice",
    "ulica": "Krajna",
    "orientacne_domu": 30
  }
],
"cestovnePoistenia": [],
"korespondencnaAdresa": {
  "id": 5,
  "psc": 94444,
  "obec": "Presov",
  "ulica": "Hlavna",
  "orientacneCisloDomu": 12
},
{
  "id": 2,
  "meno": "Matej",
  "priezvisko": "Kukučín",
  "email": "matej@gmail.com",
  "rc": 2782631125,
  "psc": 93005,
  "obec": "Gabčíkovo",
  "ulica": "Letná",
  "orientacneCislo": 32,
```

```
"poisteniaMajetku": [],
"cestovnePoistenia": [],
"korespondencnaAdresa": {
  "id": 2,
  "psc": 93005,
  "obec": "Gabčíkovo",
  "ulica": "Letná",
  "orientacneCisloDomu": 32
},
{
  "id": 4,
  "meno": "David",
  "priezvisko": "Sabo",
  "email": "sabo@gmail.com",
  "rc": 1234567888,
  "psc": 11111,
  "obec": "Sala",
  "ulica": "Hviezdoslavova",
  "orientacneCislo": 33,
  "poisteniaMajetku": [
    {
      "id": 2,
      "datumVzniku": "07/06/2023",
      "hodnota": 123,
      "typNehnutelnosti": "Byt",
      "psc": 22956,
      "obec": "Trnava",
      "ulica": "Hlavna",
      "orientacne_domu": 2
    }
  ],
  "cestovnePoistenia": [],
  "korespondencnaAdresa": {
    "id": 4,
    "psc": 94444,
    "obec": "Prešov",
    "ulica": "Hlavna",
    "orientacneCisloDomu": 2
  }
},
{
  "id": 1,
  "meno": "Andrej",
  "priezvisko": "Sládkovič",
  "email": "marína@gmail.com",
  "rc": 2782631111,
  "psc": 93005,
  "obec": "Senec",
  "ulica": "Jarná",
  "orientacneCislo": 12,
```

```
"poisteniaMajetku": [],
"cestovnePoistenia": [
  {
    "idCestovnePoistenie": 1,
    "zaciatok": "02/03/2023",
    "koniec": "02/03/2024",
    "typPoistenia": "zodpovednosť za škodu"
  }
],
"korespondencnaAdresa": {
  "id": 1,
  "psc": 93005,
  "obec": "Senec",
  "ulica": "Jarná",
  "orientacneCisloDomu": 12
},
{
  "id": 3,
  "meno": "Adam",
  "priezvisko": "Zabzell",
  "email": "zabzel@gmail.com",
  "rc": 5612354447,
  "psc": 94444,
  "obec": "Bratislava",
  "ulica": "Hlavne namestie",
  "orientacneCislo": 5,
  "poisteniaMajetku": [
    {
      "id": 1,
      "datumVzniku": "02/03/2023",
      "hodnota": 12325,
      "typNehnutelnosti": "Byt",
      "psc": 15956,
      "obec": "Bratislava",
      "ulica": "Hlavna",
      "orientacne_domu": 7
    }
  ],
  "cestovnePoistenia": [],
  "korespondencnaAdresa": {
    "id": 3,
    "psc": 94444,
    "obec": "Bratislava",
    "ulica": "Hlavne namestie",
    "orientacneCisloDomu": 5
  }
}
]
```

Výstup na GET poistenca s ID 3

```
{
  "id": 3,
  "meno": "Adam",
  "priezvisko": "Zabzell",
  "email": "zabzel@gmail.com",
  "rc": 5612354447,
  "psc": 94444,
  "obec": "Bratislava",
  "ulica": "Hlavne namestie",
  "orientacneCislo": 5,
  "poisteniaMajetku": [
    {
      "id": 1,
      "datumVzniku": "02/03/2023",
      "hodnota": 12325,
      "typNehnutelnosti": "Byt",
      "psc": 15956,
      "obec": "Bratislava",
      "ulica": "Hlavna",
      "orientacne_domu": 7
    }
  ],
  "cestovnePoistenia": [],
  "korespondencnaAdresa": {
    "id": 3,
    "psc": 94444,
    "obec": "Bratislava",
```

```
"ulica": "Hlavne namestie",  
"orientacneCisloDomu": 5  
}  
}
```

Chybný input

```
[  
    "Name has to be filled! ",  
    "Last name has to be filled! ",  
    "RC has to have 10 digits! ",  
    "Street has to be filled! "  
]
```

Databázový model

