**Združenie DEUS**

**Dátové centrum obcí a miest**

**Integračný manuál pre ISM:**

**Časť Integrácia na registre**

**Register adries**

|  |  |
| --- | --- |
| **VERZIA DOKUMEnTU** | 1.2 |
| **dáTUM VYDANIA**  **id DOKUMENTU**  **aUTOR**  **ZHOTOVITEĽ** | 25. 07. 2019  DCOM-ISM-IM\_RA  PosAm, s r.o., DATALAN, a.s.  PosAm s.r.o., DATALAN, a.s. |

# Obsah

[Obsah 2](#_Toc14946265)

[Záznam o zmenách 3](#_Toc14946266)

[1 Úvod 4](#_Toc14946267)

[2 Skratky 4](#_Toc14946268)

[3 Register adries 5](#_Toc14946269)

[3.1 Iniciálna dávka 5](#_Toc14946270)

[3.2 Informácie o zmenách – zmenové dávky 5](#_Toc14946271)

[4 Integrácia ISM na messaging 6](#_Toc14946272)

[4.1 Načítanie zmenových správ 6](#_Toc14946273)

[4.2 Správa - dávka 7](#_Toc14946274)

[4.3 Vysvetlenie pojmov IS RA 14](#_Toc14946275)

[4.3.1 Dátum účinnosti (effectiveDate) 14](#_Toc14946276)

[4.3.2 Dátum platnosti (validFrom) 14](#_Toc14946277)

[4.3.3 Dátum a čas zápisu (createdAt) 14](#_Toc14946278)

[4.3.4 Objekt 14](#_Toc14946279)

[4.3.4.1 Vznik objektu 15](#_Toc14946280)

[4.3.4.2 Zmena objektu 15](#_Toc14946281)

[4.3.4.3 Zánik objektu 15](#_Toc14946282)

[4.3.5 Záznam 16](#_Toc14946283)

[4.3.5.1 Oprava záznamu 16](#_Toc14946284)

[4.3.5.2 Storno záznamu 16](#_Toc14946285)

[4.3.6 Verifikácia adresy (verifiedAt) 16](#_Toc14946286)

[4.4 Logický model entít RA 17](#_Toc14946287)

[4.5 IS Mesta 19](#_Toc14946288)

[4.5.1 Vzorová implementácia konzumenta na strane ISM 19](#_Toc14946289)

[5 Prílohy 21](#_Toc14946290)

# Záznam o zmenách

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzia** | **Popis zmien** | **Autor zmeny** | **Dátum** |
| 1.0 | Prvá verzia dokumentu | Tomáš Mihalovič | 22.3.2019 |
| 1.1 | Doplnenie informácií o iniciálnej dávke (v kapitole 3.1) a rozšírený popis zmenovej dávky (kapitola 4.2) | Ivan Krištek | 19.7.2019 |
| 1.2 | Doplnenie vysvetlenia pojmov IS RA (kapitola 4.3), logicky model entit IS RA (kapitola 4.4) | Ivan Krištek | 25.7.2019 |
|  |  |  |  |

# Úvod

Dokument typu „integračný manuál pre ISM“ je určený dodávateľom IS mesta (ISM), ktorí sa budú integrovať na DCOM. Tento konkrétny integračný manuál popisuje integráciu na register adries.

Tento dokument nadväzuje na dokument s názvom *Integračný manuál pre ISM: Princípy integrácie*, ktorý popisuje základné princípy integrácie ISM a IS DCOM.

# Skratky

|  |  |
| --- | --- |
| Skratka | Význam |
| ISM | Informačný systém mesta |
| IS VS | informačný systém verejnej správy |
| RA | Register adries |
|  |  |
|  |  |

# Register adries

Integračné rozhranie RA bude k dispozícii mestu, ktoré nebude využívať modul Evidencia osôb v plnom rozsahu. IS mesta si bude udržiavať vlastnú kópiu dát RA.

Integračný komponent v IS DCOM bude vykonávať v pravidelných intervaloch preberanie zmenových dávok z registra adries a prostredníctvom asynchrónneho messagingu ich bude poskytovať integrovaným konzumentom ISM.

## Iniciálna dávka

V prípade potreby je možné prostredníctvom IS DCOM importovať iniciálnu dávku zo súborov, ktoré obsahujú údaje z registra adries. Žiadosť o vytvorenie a následný import iniciálnej alebo ľubovoľnej inej dávky (snapshot-u) z RA bude riešiť priamo dodávateľ mesta so správcom IS RA. Celá iniciálna dávka bude prenášaná a spracovaná cez messaging platformu rovnakým spôsobom ako zmenové dávky. Každý súbor môže obsahovať 1 až N zmenových dávok pričom každá bude v separátnej správe.

**Proces:** na základe špeciálnej požiadavky IS DCOM načíta sadu súborov, ktoré obsahujú aktuálne údaje adries evidovaných v RA, vrátane všetkých príslušných číselníkov. Tieto dáta transformuje do interných štruktúr, uloží ich ako referenčné a sprístupní ich podľa potreby cez messaging platformu jednotlivým ISM.

Iniciálna dávka z RA je vo formáte XML súborov, XSD schéma je zhodná s XSD schémou pre zmenové dávky:

* 00 – sumár inicializačnej dávky obsahujúci ID dávky (changesID) s výsledným kódom generovania inicializačnej dávky (resultCode), dátumom a časom vygenerovania inicializačnej dávky,
* 01 – inicializačná dávka krajov,
* 02 – inicializačná dávka okresov,
* 03 – inicializačná dávka obcí,
* 04 – inicializačná dávka častí obcí,
* 05 – inicializačná dávka ulíc,
* 06 – inicializačná dávka súpisných čísel (budovy),
* 07 – inicializačná dávka orientačných čísel (vchody s adresným bodom),
* 08 – inicializačná dávka bytov.

## Informácie o zmenách – zmenové dávky

IS DCOM preberá pravidelne zmenové dávky z RA. Tieto zmenové dávky budú sprístupnené na prevzatie jednotlivým ISM. Obsah zmenovej dávky bude IS DCOM zasielať prostredníctvom queues v messaging platforme.

V súvislosti so zmenovými dávakami IS DCOM využíva nasledovné biznis služby RA:

* Ra\_changes - poskytnutie zoznamu identifikátorov adresy so zmenenými referenčnými údajmi - poskytnutie zoznamu jednoznačných identifikátorov údajových entít (zoznam identifikátora adresy), ktorých referenčné údaje boli zmenené.
  + Operácie:
    - GetChanges
    - ConfirmChanges

**Proces:** v definovanom časovom intervale (zvyčajne jeden krát za deň) si systém DCOM vyžiada z RA zoznam zmien pre všetky adresy. Získané dáta transformuje do interných štruktúr a aktualizuje údaje v lokálnej kópii v DCOM.

# Integrácia ISM na messaging



Obrázok 1: Integračná schéma

Na obrázku je vidieť logickú schému jednotlivých integračných komponentov pre zmenové dávky registra adries. Zmenové dávky sa spracovávajú v backende IS DCOM a výsledné správy sú cez exchange typu fanout rozposielané do všetkých queues ISM.

Na komunikáciu sa v rámci IS DCOM ale aj pre externé ISM používa asynchrónna messaging platforma RabbitMQ. Jednotlivé zmenové dávky RA sú publikované formou XML správ do exchange ra, ktorá ich následne fanout algoritmom distribuuje identicky do jednotlivých queues ISM. Jedna queue sa interne používa aj na spracovanie zmenových dávok v IS DCOM.

Integrácia ISM a IS DCOM je jednoduchou klientskou integráciou na messaging platformu RabbitMQ v prostredí integračnej zbernice pre externé systémy. Integračný modul IS DCOM vykonáva preberanie zmenových dávok z registra adries a prostredníctvom asynchrónneho messagingu ich poskytuje konzumentom (ISM#1 až ISM#N).

Každý konzument bude mať dedikovanú queue, odkiaľ ich môže preberať. Všetky queues registra adries budú umiestnené na vhost-e “/ra”.

## Načítanie zmenových správ

Odporúčaný proces načítania a spracovania správ pre ISM:

1. ISM sa pripojí na svoje queue messaging platformy a stiahne správu

apmq://messaging.intra.dcom.sk:5671/ra/<nazov\_queue\_mesta>

1. ISM správu interne spracuje uloží si ju a odošle potvrdenie o úspešnom prevzatí
2. IS DCOM Messaging platforma príjme potvrdenie o prevzatí a správu vymaže z queue

Spôsob potvrdzovania prevzatia správ v messaging platforme je možné konfiguračne nastavovať na strane klienta ISM. Zodpovednosťou ISM je potvrdiť správu až vtedy, keď ju má spracovanú a údaje spoľahlivo uložené.

**Pozor:** v prípade potvrdenia prevzatia správy messaging platforme je správa z nej zmazaná a už nie je možné ju opakovane načítať.

## Správa - dávka

Štruktúra správ jednotlivých dávok je totožná s XSD štruktúrou, ktorú definuje register adries (viď príloha - **RA\_Changes\_v20.xsd**) ako odpoveď na vyžiadanie si zmenovej dávky.

Konkrétne je to element **GetChangesResponse.Return** ktorý je zloženým root elementom

<xsd:complexType name="getChangesResponse">  
 <xsd:sequence>  
 <xsd:element minOccurs="0" name="return">  
 <xsd:complexType>  
 <xsd:complexContent>  
 <xsd:extension base="ns0:resultCodeResponse">  
 <xsd:sequence>  
 <xsd:element minOccurs="0" name="changesId" type="xsd:long"/>  
 <xsd:element minOccurs="0" name="changesGenerated" type="xsd:dateTime"/>  
 <xsd:element maxOccurs="unbounded" minOccurs="0" name="register"  
 type="ns0:ChangeRegisterCType"/>  
 </xsd:sequence>  
 </xsd:extension>  
 </xsd:complexContent>  
 </xsd:complexType>  
 </xsd:element>  
 </xsd:sequence>  
</xsd:complexType>

Príklad XML správy, ktorú tvorí zmenová dávka (je uvedená len jej časť)

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<ra:GetChangesResponse.Return xmlns:ra="http://www.minv.sk/ra">  
 <changesGenerated>2018-12-11T18:03:46.550+01:00</changesGenerated>  
 <changesId>8865931</changesId>  
 <register>  
 <type>PROPERTY\_REGISTRATION\_NUMBER</type>  
 <count>117</count>  
 <propertyRegistrationNumberChange>  
 <changeId>201772003</changeId>  
 <changedAt>2018-12-10T09:05:08.711+01:00</changedAt>  
 <databaseOperation>INSERT</databaseOperation>  
 <objectId>4285409</objectId>  
 <versionId>6036737</versionId>  
 <createdReason>CREATE</createdReason>  
 <validFrom>2018-12-10T09:05:08.695+01:00</validFrom>  
 <validTo>3000-12-31T23:59:59+01:00</validTo>  
 <effectiveDate>2018-12-10+01:00</effectiveDate>  
 <PropertyRegistrationNumber>448</PropertyRegistrationNumber>  
 <Building ContainsFlats="false">  
 <BuildingPurpose>  
 <Codelist>  
 <CodelistCode>CL010142</CodelistCode>  
 <CodelistItem>  
 <ItemCode>residentialBuilding</ItemCode>  
 <ItemName>Bytová budova</ItemName>  
 </CodelistItem>  
 </Codelist>  
 </BuildingPurpose>  
 <BuildingTypeCode>  
 <Codelist>  
 <CodelistCode>CL010143</CodelistCode>  
 <CodelistItem>  
 <ItemCode>10</ItemCode>  
 <ItemName>Rodinný dom</ItemName>  
 </CodelistItem>  
 </Codelist>  
 </BuildingTypeCode>  
 </Building>  
 <municipalityIdentifier>2737</municipalityIdentifier>  
 </propertyRegistrationNumberChange>

........................................................................

<buildingUnitChange>  
 <changeId>201790133</changeId>  
 <changedAt>2018-12-10T12:47:21.259+01:00</changedAt>  
 <databaseOperation>INSERT</databaseOperation>  
 <objectId>4285627</objectId>  
 <versionId>3225573</versionId>  
 <createdReason>CREATE</createdReason>  
 <validFrom>2018-12-10T12:47:21.197+01:00</validFrom>  
 <validTo>3000-12-31T23:59:59+01:00</validTo>  
 <effectiveDate>2018-12-10+01:00</effectiveDate>  
 <BuildingUnit UnitNumber="26.5.2006" Floor="4">byt</BuildingUnit>  
 <buildingNumberIdentifier>1772957</buildingNumberIdentifier>  
 </buildingUnitChange>  
 </register>  
</ra:GetChangesResponse.Return>

Zmenové dávky poskytujú údaje v rozsahu:

* kraje
* okresy
* obce
* časti obce
* ulice
* súpisné čísla
* orientačné čísla
* byty

Rozsah poskytovaných údajov nie je možné meniť. V jednej zmenovej dávke je maximálny počet zmenových záznamov 1000.

Hlavné elementy dátovej štruktúry sú:

* changesGenerated – dátum a čas kedy bola dávka vygenerovaná [**xsd:dateTime**]
* changesId – identifikátor zmenovej dávky [**xsd:long**]
* register – zložený typ, ktorý obsahuje samotné zmeny

Popis dátovej štruktúry v elemente **register**:

**register.type** - názov typu registra, enumerácia "RegisterType"

**register.count** - počet zmien pre daný register v danej dávke

**register.**[**register**]**Change** - zmenový záznam pre objekt daného registra

***regionChange*** - zmenový záznam pre objekt kraja

**changeId** - identifikátor zmeny, long

**changedAt** - dátumčas vykonania zmeny

**databaseOperation** - vykonaná databázová operácia, enumerácia "ChangeType"

**objectId** - identifikátor objektu, long

**versionId** - identifikátor verzie záznamu, long

**createdReason** - dôvod vytvorenia záznamu, enumerácia "CreatedReason"

**validFrom** - dátumčas platnosti od

**validTo** - dátumčas platnosti do

**effectiveDate** - dátum účinnosti, date

[**Region**] - zmenový záznam pre objekt základného číselníka

**Codelist** - základný číselník

**CodelistCode** - kód základného číselníka

**CodelistItem** - položka základného číselníka

**itemName** - názov položky základného číselníka, string

**itemCode** - kód položky základného číselníka, string

***countyChange*** - zmenový záznam pre objekt okresu

**changeId** - identifikátor zmeny, long

**changedAt** - dátumčas vykonania zmeny

**databaseOperation** - vykonaná databázová operácia, enumerácia "ChangeType"

**objectId** - identifikátor objektu, long

**versionId** - identifikátor verzie záznamu, long

**createdReason** - dôvod vytvorenia záznamu, enumerácia "CreatedReason"

**validFrom** - dátumčas platnosti od

**validTo** - dátumčas platnosti do

**effectiveDate** - dátum účinnosti, date

[**County**] - zmenový záznam pre objekt základného číselníka

**Codelist** - základný číselník

**CodelistCode** - kód základného číselníka

**CodelistItem** - položka základného číselníka

**itemName** - názov položky základného číselníka, string

**itemCode** - kód položky základného číselníka, string

**regionIdentifier** - identifikátor nadradeného kraja, long

***municipalityChange*** - zmenový záznam pre objekt obce

**changeId** - identifikátor zmeny, long

**changedAt** - dátumčas vykonania zmeny

**databaseOperation** - vykonaná databázová operácia, enumerácia "ChangeType"

**objectId** - identifikátor objektu, long

**versionId** - identifikátor verzie záznamu, long

**createdReason** - dôvod vytvorenia záznamu, enumerácia "CreatedReason"

**validFrom** - dátumčas platnosti od

**validTo** - dátumčas platnosti do

**effectiveDate** - dátum účinnosti, date

[**Municipality**] - zmenový záznam pre objekt základného číselníka

**Codelist** - základný číselník

**CodelistCode** - kód základného číselníka

**CodelistItem** - položka základného číselníka

**itemName** - názov položky základného číselníka, string

**itemCode** - kód položky základného číselníka, string

**countyIdentifier** - identifikátor nadradeného okresu, long

**status** - štatút obce, enumerácia "MunicipalityStatus"

**cityIdentifier** - identifikátor nadradeného mesta pre mestskú časť, long

***districtChange*** - zmenový záznam pre objekt časti obce

**changeId** - identifikátor zmeny, long

**changedAt** - dátumčas vykonania zmeny

**databaseOperation** - vykonaná databázová operácia, enumerácia "ChangeType"

**objectId** - identifikátor objektu, long

**versionId** - identifikátor verzie záznamu, long

**createdReason** - dôvod vytvorenia záznamu, enumerácia "CreatedReason"

**validFrom** - dátumčas platnosti od

**validTo** - dátumčas platnosti do

**effectiveDate** - dátum účinnosti, date

[**District**] - zmenový záznam pre objekt základného číselníka

@**uniqueNumbering** - atribút pre jedinečné číslovanie budov v časti obce

**Codelist** - základný číselník

**CodelistCode** - kód základného číselníka

**CodelistItem** - položka základného číselníka

**itemName** - názov položky základného číselníka, string

**itemCode** - kód položky základného číselníka, string

**municipalityIdentifier** - identifikátor nadradenej obce, long

***streetNameChange*** - zmenový záznam pre objekt ulice

**changeId** - identifikátor zmeny, long

**changedAt** - dátumčas vykonania zmeny

**databaseOperation** - vykonaná databázová operácia, enumerácia "ChangeType"

**objectId** - identifikátor objektu, long

**versionId** - identifikátor verzie záznamu, long

**createdReason** - dôvod vytvorenia záznamu, enumerácia "CreatedReason"

**validFrom** - dátumčas platnosti od

**validTo** - dátumčas platnosti do

**effectiveDate** - dátum účinnosti, date

**StreetName** - názov objektu, string

**municipalityIdentifiers** - zoznam longov, buď identifikátor obce alebo identifikátory mestských častí, cez ktoré ulica prechádza

**districtIdentifiers** - zoznam longov, zoznam identifikátorov častí obce, cez ktoré ulica prechádza

***propertyRegistrationNumberChange*** - zmenový záznam pre objekt súpisné číslo

**changeId** - identifikátor zmeny, long

**changedAt** - dátumčas vykonania zmeny

**databaseOperation** - vykonaná databázová operácia, enumerácia "ChangeType"

**objectId** - identifikátor objektu, long

**versionId** - identifikátor verzie záznamu, long

**createdReason** - dôvod vytvorenia záznamu, enumerácia "CreatedReason"

**validFrom** - dátumčas platnosti od

**validTo** - dátumčas platnosti do

**effectiveDate** - dátum účinnosti, date

**propertyRegistrationNumber** - súpisné číslo, long

**Building** - údaje budovy

@**containsFlats** - atribút, bytový dom/rodinný dom

[**BuildingPurpose**] - zmenový záznam pre objekt základného číselníka

**Codelist** - základný číselník

**CodelistCode** - kód základného číselníka

**CodelistItem** - položka základného číselníka

**itemName** - názov položky základného číselníka, string

**itemCode** - kód položky základného číselníka, string

[**BuildingTypeCode**] - zmenový záznam pre objekt základného číselníka

**Codelist** - základný číselník

**CodelistCode** - kód základného číselníka

**CodelistItem** - položka základného číselníka

**itemName** - názov položky základného číselníka, string

**itemCode** - kód položky základného číselníka, string

**municipalityIdentifier** - identifikátor nadradenej obce, long

**districtIdentifier** - identifikátor nadradenej časti obce, long

***buildingNumberChange*** - zmenový záznam pre objekt orientačné číslo

**changeId** - identifikátor zmeny, long

**changedAt** - dátumčas vykonania zmeny

**databaseOperation** - vykonaná databázová operácia, enumerácia "ChangeType"

**objectId** - identifikátor objektu, long

**versionId** - identifikátor verzie záznamu, long

**createdReason** - dôvod vytvorenia záznamu, enumerácia "CreatedReason"

**validFrom** - dátumčas platnosti od

**validTo** - dátumčas platnosti do

**effectiveDate** - dátum účinnosti, date

**BuildingNumber** - orientačné číslo, long

**BuildingIndex** - identifikátor adresy, string

**AddressPoint** - adresný bod, zložený atribút

**BLH** - súradnicový systém

**AxisB** - súradnice zemepisnej šírky

**AxisL** - súradnice zemepisnej dĺžky

**propertyRegistrationNumberIdentifier** - identifikátor súpisného čísla, long

**streetNameIdentifier** - identifikátor ulice, long

**verifiedAt** - dátumčas verifikovania adresy

***buildingUnitChange*** - zmenový záznam pre objekt časť budovy

**changeId** - identifikátor zmeny, long

**changedAt** - dátumčas vykonania zmeny

**databaseOperation** - vykonaná databázová operácia, enumerácia "ChangeType"

**objectId** - identifikátor objektu, long

**versionId** - identifikátor verzie záznamu, long

**createdReason** - dôvod vytvorenia záznamu, enumerácia "CreatedReason"

**validFrom** - dátumčas platnosti od

**validTo** - dátumčas platnosti do

**effectiveDate** - dátum účinnosti, date

**BuildingUnit** - časť budovy, byt

@**UnitNumber** - atribút číslo bytu, string

@**Floor** - atribút podlažie, long

**buildingNumberIdentifier** - identifikátor orientačného čísla, long

Príklad zmenovej dávky:



Na prenos správ cez RabbitMQ je použitá kompresia gzip, čo je špecifikované v hlavičke správy atribútom **content\_encoding** a jeho hodnotou **gzip.** Routing key je totožný s identifikátorom dávky. Správa tiež v hlavičke obsahuje sadu identifikátorov z distribuovaného tracingu (X3-\*).

Príklad obsahu správy z RabbitMQ admin UI:



## Vysvetlenie pojmov IS RA

### Dátum účinnosti (effectiveDate)

Dátum účinnosti je atribútom záznamov v Registri adries. Dátum účinnosti **je určený zákonom, nariadením alebo rozhodnutím**.

Dátum účinnosti do IS RA zadáva oprávnený používateľ.

Dátum účinnosti môže byť:

* v minulosti (DÚ < SYSDATE),
* v prítomnosti (DÚ = SYSDATE),
* v budúcnosti (DÚ > SYSDATE).

### Dátum platnosti (validFrom)

Dátum platnosti je atribútom záznamov v Registri adries. Ak existujú viaceré záznamy toho istého objektu, platný môže byť vždy iba jeden záznam.

Dátum platnosti je v IS RA needitovateľný, spravuje a zapisuje ho systém automaticky.

Dátum platnosti (DP) vyplýva z Dátumu účinnosti (DÚ) a z toho, kedy je údaj zapisovaný do IS RA:

* ak je DÚ v minulosti (DÚ < SYSDATE), údaj sa okamžite zapíše do systému a je okamžite platný (DP = SYSDATE),
* ak je DÚ v prítomnosti (DÚ = SYSDATE), údaj sa okamžite zapíše do systému a je okamžite platný (DP = SYSDATE),
* ak je DÚ v budúcnosti (DÚ > SYSDATE), údaj sa okamžite zapíše do systému a jeho platnosť sa nastaví podľa DÚ (DP = DÚ).

Podľa DP rozlišujeme štyri stavy záznamov:

* **historický** (DP v minulosti, bol opravený alebo zmenený, a ním nasleduje ďalší záznam o objekte),
* **platný** (DP je v minulosti, je to v súčasnosti platný záznam o objekte),
* **neúčinný** (DP je v budúcnosti, objekt ešte nenadobudol účinnosť),
* **neplatný** (DP v minulosti, objekt bol zrušený).

### Dátum a čas zápisu (createdAt)

Pre každý údaj zapísaný do IS RA sa ukladá presný dátum a čas jeho zápisu. Tento dátum a čas nie je možné zmeniť.

Dátum a čas zápisu sa posiela externým systémom a je ponechané na nich, ako ho budú interpretovať.

### Objekt

Objekty v aplikácii IS RA sú prvky adresy – kraj, okres, obec, časť obce, ulica, súpisné číslo, orientačné číslo a byty. IS RA slúži na evidenciu týchto objektov a správu záznamov o nich. Objekty reprezentované v IS RA vychádzajú z business logiky informačného systému Registra adries, ktorá vychádza z návrhu zákona:

* **Adresa** je súbor údajov, identifikujúcich umiestnenie budovy a vchodov do budovy na území SR. Súbor údajov adresy obsahuje:
  + - názov kraja
    - názov okresu
    - názov obce (štatúty obce: mesto, mestská časť, obec, vojenský obvod)
    - názov časti obce, ak sa obec člení na časti
    - názov ulice alebo verejného priestranstva, ak sú v obci zavedené
    - súpisné číslo
    - orientačné číslo, ak je určené
    - adresný bod
    - byty, ak sú evidované
  + **Kraj** je najvyššia administratívna a územnosprávna jednotka a jednotka štátnej správy na Slovensku.
  + **Okres** je druhá najmenšia administratívna jednotka Slovenska. Niekoľko okresov tvorí kraj.
  + **Obec** je základný územný samosprávny a správny celok na Slovensku.
  + **Časť obce** je nižšia územná jednotka obce, ktorá má spravidla vlastný názov a územie, môže byť osídlená alebo ju môže tvoriť len pôda.
  + **Ulica** je verejné priestranstvo s definovanými hranicami a názvom jedinečným v rámci obce.
  + **Súpisné číslo** je na Slovensku číslo povinne prideľované ako unikátne každej budove alebo stavbe v rámci obce alebo časti obce.
  + **Orientačné číslo** je na Slovensku číslo, ktoré môže obec prideliť budove ako unikátne v rámci ulice, ak je v obci zavedený uličný systém s názvami ulíc.
  + **Adresný bod** je priestorový údaj, ktorý v priestore označuje polohu budovy alebo vchodu. Adresný bod je vyjadrený v priestorových súradniciach záväzného geodetického referenčného systému, ktorý je Európsky terestrický referenčný systém 1989 s alfanumerickým kódom ETRS89, alebo Súradnicový systém Jednotnej trigonometrickej siete katastrálnej s alfabetickým kódom S-JTSK.

#### Vznik objektu

Vznik objektu je možný, iba ak nie je pre nadradený objekt evidovaný zánik.

V DB vznikne nový záznam s príznakom VZNIK

* ak je DÚ vzniku v minulosti, DP je aktuálny dátum a čas,
* ak je DÚ vzniku v prítomnosti, DP je aktuálny dátum a čas,
* ak je DÚ vzniku v budúcnosti, DP je rovnaký deň ako DÚ.

#### Zmena objektu

Zmena objektu je možná, iba ak neexistuje záznam objektu, ktorý ešte nenadobudol účinnosť. DÚ zmeny musí byť väčší ako DÚ vzniku, alebo posledného účinného záznamu.

Predchádzajúci záznam bude mať ukončenú účinnosť aj platnosť dňom platnosti zmeny.

Vznikne nový záznam v DB s príznakom ZMENA

* ak je DÚ zmeny v minulosti, DP je aktuálny dátum a čas,
* ak je DÚ zmeny v prítomnosti, DP je aktuálny dátum a čas,
* ak je DÚ zmeny v budúcnosti, DP je rovnaký deň ako DÚ.

#### Zánik objektu

Zánik je možný, iba ak neexistuje záznam objektu, ktorý ešte nenadobudol účinnosť. Zánik je možný, iba ak neexistujú podradené objekty.

DÚ zániku musí byť väčší ako DÚ vzniku, alebo posledného účinného záznamu.

Vznikne nový záznam v DB s príznakom ZÁNIK

* ak je DÚ zániku v minulosti, DP je aktuálny dátum a čas,
* ak je DÚ zániku v prítomnosti, DP je aktuálny dátum a čas,
* ak je DÚ zániku v budúcnosti, DP je rovnaký deň ako DÚ.

Predchádzajúci záznam bude mať ukončenú účinnosť aj platnosť dňom platnosti zániku.

Záznam o zániku má ukončenú platnosť dňom platnosti zániku (zánik nie je sám o sebe nikdy platný). Objekt, ktorý zanikol má stav „neplatný“ – je to stav posledného záznamu o tomto objekte.

### Záznam

O každom objekte evidovanom v IS RA môže existovať viacero záznamov – napr. ak bol objekt založený, a potom zmenený, tento objekt má jeden záznam o vzniku a jeden záznam o zmene. Záznam predstavuje časovú verziu objektu a obsahuje všetky údaje o objekte (napr. názov, dátum účinnosti a pod.).

Všetky záznamy o danom objekte je možné zobraziť v detaile objektu.

#### Oprava záznamu

Oprava slúži na vykonanie malých zmien na zázitemName, tak aby záznam presne zodpovedal stavu vecí (napr. oprava preklepu v názve).

Opraviť je možné iba posledný záznam. Pri oprave je možné opraviť všetky údaje, okrem dátumových údajov. Záznamy, ktoré majú ukončenú platnosť nie je možné opraviť.

Ak je záznam ešte neúčinný - opravou sa záznam v DB bez následkov prepíše.

Ak je záznam platný - opravou sa vytvorí nový záznam s príznakom OPRAVA.

Oprava nadobúda platnosť okamžite po zápise.

#### Storno záznamu

Storno slúži na odstránenie chybného záznamu, ktorý nie je možné opraviť (napr. bol chybne zadaný dátum účinnosti).

Storno nadobúda platnosť okamžite po zápise.

Môžu nastať tri prípady:

1. Ak je posledný záznam už účinný (má stav „platný“ alebo „neplatný“) a ak zároveň existuje predposledný záznam (má stav „historický“), vytvorí sa kópia predposledného záznamu. Táto kópia bude mať stav „platný“ a hodnota atribútu „Typ záznamu v evidencii“ bude nastavená na hodnotu „Storno záznamu v evidencii“. Záznam, ktorý bol pôvodne posledný, bude označený za historický.
2. Ak je záznam už účinný (má stav „platný“) a ak neexistuje predposledný záznam (ide o storno záznamu o vzniku objektu) vytvorí sa kópia posledného záznamu (kde hodnota atribútu „Typ záznamu v evidencii“ je nastavená na „Storno záznamu v evidencii“) a ďalšia kópia posledného záznamu (kde hodnota atribútu „Typ záznamu v evidencii“ je nastavená na „Zrušenie záznamu v evidencii“). Objekt, ktorému bol týmto spôsobom vystornovaný záznam o vzniku, je „neplatný“.
3. Ak je stornovaný záznam ešte neúčinný (má stav „neúčinný“), tento záznam sa vymaže.

**Podmienky:**

* je možné stornovať iba posledný záznam v evidencii daného objektu,
* nesmie existovať celoštátne moratórium,
* nesmie existovať moratórium s územnou platnosťou pre daný objekt.

### Verifikácia adresy (verifiedAt)

Verifikácia adresy slúži na potvrdenie správnosti evidovaných údajov adresy. Údaje adresy verifikuje zodpovedný pracovník obce, t.j. používateľ s oprávnením na evidenciu adries.

Dátum a čas verifikácie zapisuje systém po potvrdení stavu verifikácie používateľom.

Dátum a čas verifikácie sa poskytujú externým systémom v zmenových dávkach.

## Logický model entít RA



Obrázok 2: Logický model entít IS RA

Logický model popisuje previazanosť medzi jednotlivými registrami evidovanými v IS RA a zároveň aj ich významové atribúty.

Každý register má v názve uvedený aj anglický ekvivalent, pod ktorým vystupuje vo webových službách.

**Adresa** je definovaná identifikátorom adresy a adresným bodom. RA eviduje atribúty adresy pre adresy na orientačnom alebo súpisnom čísle. Na orientačnom čísle je adresa evidovaná, ak je v obci zavedený uličný systém. Ak obec nemá zavedený uličný systém, adresa je evidovaná na súpisnom čísle. Držiteľom adresy je vchod (alebo "virtuálny vchod").

**Identifikátor adresy** je povinný jedinečný identifikátor adresy v celom registri adries, alfanumerický generovaný reťazec, v minulosti nazývaný index domu (pôvodne nazývaný počítačové číslo domu). Každá adresa musí mať identifikátor adresy. IS RA generuje identifikátor adresy pre každú novú adresu vytvorenú v RA, identifikátory existujúcich adries boli prevzaté z Registra obyvateľstva.

**Vchod** je reprezentovaný **orientačným číslom**. Atribúty vchodu sú:

* **identifikátor adresy**
* **adresný bod**
* **PSČ**

Každý vchod patrí práve jednej budove (súpisné číslo).

Každý vchod môže mať evidované byty.

Ak obec má definovaný uličný systém, tak každý vchod budovy je priradený práve jednej ulici. Ak má budova viac vchodov, je možné, že každý z nich môže byť priradený inej ulici.

**Virtuálny vchod** je vchod bez orientačného čísla, na ktorom sa eviduje identifikátor adresy a adresný bod, ale nemá naplnené atribúty orientačné číslo a znak orientačného čísla. Virtuálny vchod je nositeľom adresy v obciach bez uličného systému, kde adresu tvorí len súpisné číslo a názov obce (a okres, kraj, PSČ a pod.).

**Budova** je reprezentovaná **súpisným číslom**.

Budova (súpisné číslo) má vždy minimálne jeden vchod. Ak v obci, do ktorej budova patrí, nie je definovaný uličný systém, tak sa na budove eviduje tzv. "virtuálny vchod".

Každá budova je priradená práve jednej obci a zároveň môže byť priradená časti obce, ak je daná obec administratívne rozdelená na časti obce.

**Byt** je evidovaný na vchode.

**Ulica** vždy patrí minimálne jednej obci, prípadne jednej alebo viacerým mestským častiam.

Ulica môže byť priradená viacerým častiam obce, ak je obec rozdelená na časti obce.

**Cestný úsek / os ulice** popisuje geografické znázornenie ulice, pričom sú na ňom evidované atribúty označenie a trieda cesty.

**Obec** je základná územnosprávna jednotka.

Každá obec má definovaný **štatút** obce (ako enumeráciu), možné štatúty:

* obec - MUNICIPALITY
* mesto - CITY
* mestská časť - CITY\_DISTRICT
* vojenský obvod - MILITARY\_DISTRICT
* mesto zriadené osobitný zákonom - CITY\_MANAGED\_BY\_SPECIAL\_LAW (Bratislava a Košice)

Ak obec je mestskou častou, musí mať naplnený atribút mesto (city), ktorý definuje, ktorému mestu daná mestská časť patrí. Mestom sa rozumie každá obec so štatútom mesto alebo mesto zriadené osobitným zákonom (len Bratislava a Košice).

Každá obec patrí práve jednému okresu.

**Časť obce** je nižšia územná jednotka obce, menšia ako obec, ktorá má spravidla vlastné katastrálne územie a vlastný názov, určený zo zákona oprávneným orgánom štátu.

Časť obce patrí práve jednej obci.

**Okres** je menšia administratívna jednotka územného členenia SR. Okres patrí práve jednému kraju.

**Kraj** je územnosprávna jednotka štátnej správy.

**Registre** kraj, okres a obec sú tzv. štatistické číselníky, ktoré zverejňuje Štatistický úrad SR. Pojem **číselník** sa používa v zmysle Vyhlášky Štatistického úradu SR z 12. septembra 2002, ktorou sa vydáva štatistický číselník krajov, štatistický číselník okresov a štatistický číselník obcí.

Každý register udržuje svoju históriu záznamov. Jednotlivé časové verzie záznamu majú spoločný identifikátor (objectId resp. identifier), ale rozdielne ID verzie (versionId resp. version). Ak sa záznam jedného registra odkazuje na iný register (okres na kraj), tak daná časová verzia okresu sa odkazuje len na indentifikátor daného kraja. Ktorá časová verzia kraja k danej časovej verzii okresu bude priradená, závisí od časového rezu, v ktorom sa "pýtame". Každá časová verzia má dátumčasovú platnosť od - do (validFrom, validTo).

Tieto väzby sú v logickom modeli zakreslené so stereotypom <<proxy>>.

## IS Mesta

Konzumentov správ zmenových dávok sú jednotlivé ISM. Pripojenie konzumentov a spracovanie správ je ich zodpovednosťou pričom na úrovni messaging platformy je zabezpečený mechanizmus perzistencie správ a potvrdzovania ich prevzatia.

V nasledovnej kapitole je vidieť ukážku vzorovej implementácie jednoduchého konzumenta.

### Vzorová implementácia konzumenta na strane ISM

Príklad integrácie a spracovania jednotlivých dávok vidieť na vzorovej implementácii Java spring boot aplikácie, ktorá sa pripája na RabbitMQ server pomocou Spring AMQP knižnice.

Pre klientský prístup jedinou potrebnou závislosťou je RabbitMQ:



čo v prípade maven projektu znamená nasledujúcu závislosť:

<dependency>  
 <groupId>org.springframework.boot</groupId>  
 <artifactId>spring-boot-starter-amqp</artifactId>  
</dependency>

Základné atribúty v **application.properties**, potrebné pre prihlásenie klientom na RabbitMQ server sú:

# rabbit  
spring.rabbitmq.addresses=messaging.intra.dcom.sk:5671

spring.rabbitmq.username=ra-test-user  
spring.rabbitmq.password=ra-test-user  
spring.rabbitmq.virtual-host=/ra

# tls properties

spring.rabbitmq.ssl.algorithm=TLSv1.2

spring.rabbitmq.ssl.enabled=true

spring.rabbitmq.ssl.validate-server-certificate=true

spring.rabbitmq.ssl.verify-hostname=true

**spring.rabbitmq.addresses** - obsahuje všetky nody RabbitMQ v clusterovom zapojení oddelené čiarkou.

**spring.rabbitmq.username** a **spring.rabbitmq.password** – prihlasovacie údaje

**spring.rabbitmq.virtual-host** – určenie virtual hosta v rámci RabbitMQ servera (default je „/“)

**spring.rabbitmq.ssl.\*** - ssl parametre spojenia

Pre vytvorenie TLS spojenia je nutné pridať do Java trust-store certifikát CA DCOM, ktorá vydala certifikát servera.

Na základe parametrov v **application.properties** spring boot aplikácia automaticky vytvorí connection factory a pripojí sa pri štarte na RabbitMQ.

Ďaľším potrebným krokom je vytvorenie „listenera“, ktorý sa pripojí na queue a počúva na nové správy.

Jednoduchý príklad „listenera“, ktorý spracováva všetky prichádzajúce správy vo formáte XML, prípadne aj skomprimované v gzip encodingu.

package sk.dcom.integration.ext.rachangesconsumer;  
import org.slf4j.Logger;  
import org.slf4j.LoggerFactory;  
import org.springframework.amqp.core.Message;  
import org.springframework.amqp.core.MessageProperties;  
import org.springframework.amqp.rabbit.annotation.RabbitListener;  
import org.springframework.stereotype.Component;  
import sk.dcom.integration.ext.rachanges.service.ChangesResponse;  
  
import javax.xml.bind.JAXBContext;  
import javax.xml.bind.JAXBException;  
import javax.xml.bind.Unmarshaller;  
import java.io.ByteArrayInputStream;  
import java.io.IOException;  
import java.io.InputStream;  
import java.util.zip.GZIPInputStream;  
  
@Component  
public class RaChangeListener {

public static final String *QUEUE\_RA\_CHANGES* = "ra-ba-consumer";

static final Logger *logger* = LoggerFactory.*getLogger*(RaChangeListener.class);  
  
 public ChangesResponse unmarshallChangesResponse(InputStream from) {  
 try {  
 JAXBContext jaxbContext = JAXBContext.*newInstance*(ChangesResponse.class);  
 Unmarshaller jaxbUnmarshaller = jaxbContext.createUnmarshaller();  
  
 ChangesResponse rawResponse = (ChangesResponse)jaxbUnmarshaller.unmarshal(from);  
 return rawResponse;  
 } catch (JAXBException e) {  
 *logger*.error(e.getMessage(), e);  
 return null;  
 }  
 }  
  
 @RabbitListener(queues = *QUEUE\_RA\_CHANGES*)  
 public void processChanges(Message message) {  
 InputStream input = new ByteArrayInputStream(message.getBody());  
 MessageProperties msp = message.getMessageProperties();  
 if (msp != null && msp.getContentEncoding() != null && msp.getContentEncoding().equalsIgnoreCase("gzip"))  
 {  
 *logger*.info("RECEIVED gzipped Message: trying to unpack it.");  
 try {  
 input = new GZIPInputStream(input);  
 } catch (IOException e) {  
 *logger*.error(e.getMessage(), e);  
 return;  
 }  
 }  
 ChangesResponse cr = unmarshallChangesResponse(input);  
 if (cr == null) {  
 *logger*.info("RECEIVED Message: " + message.getMessageProperties().toString() + " -> but fail to unmarshal it.");  
 return;  
 }  
 *logger*.info("RECEIVED CHANGE RESPONSE: ID=" + cr.getChangesId() + " generated on=" + cr.getChangesGenerated() + " register size=" + cr.getRegister().size());  
 }  
}

Potvrdenie jednotlivých správ v queue je defaultne nastavené tak, že ak metóda listenera (v príklade **processChanges**) zbehne bez výnimky, správa sa považuje za doručenú. Ak metóda vyvolá výnimku správa sa znovu zaradí do queue.

# Prílohy

