**Združenie DEUS**

**Dátové centrum obcí a miest**

**Integračný manuál pre ISM:**

**Časť Integrácia na registre**

**Register právnických osôb**

|  |  |
| --- | --- |
| **VERZIA DOKUMEnTU** | 1.0 |
| **dáTUM VYDANIA**  **id DOKUMENTU**  **aUTOR**  **ZHOTOVITEĽ** | 22. 03. 2019  DCOM-ISM-IM\_RPO\_v1.0  PosAm, s r.o., DATALAN, a.s.  PosAm s.r.o., DATALAN, a.s. |

# Obsah

[Obsah 2](#_Toc4600199)

[Záznam o zmenách 3](#_Toc4600200)

[1 Úvod 4](#_Toc4600201)

[2 Skratky 5](#_Toc4600202)

[3 Register právnických osôb 6](#_Toc4600203)

[3.1 Iniciálna dávka 6](#_Toc4600204)

[3.2 Informácie o zmenách – zmenové dávky 6](#_Toc4600205)

[4 Integrácia ISM na messaging 7](#_Toc4600206)

[4.1 Načítanie zmenových správ 7](#_Toc4600207)

[4.2 Správa - dávka 8](#_Toc4600208)

[4.3 Logický dátový model 10](#_Toc4600209)

[4.4 IS Mesta 11](#_Toc4600210)

[4.4.1 Vzorová implementácia konzumenta na strane ISM 11](#_Toc4600211)

[5 Prílohy 14](#_Toc4600212)

# Záznam o zmenách

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzia** | **Popis zmien** | **Autor zmeny** | **Dátum** |
| 1.0 | Prvá verzia dokumentu | Tomáš Mihalovič | 22.3.2019 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 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 právnických osôb (RPO).

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 |
| RPO | Register právnických osôb |
|  |  |
|  |  |

# Register právnických osôb

Integračné rozhranie RPO bude využívať IS mesta, ktorý si bude udržiavať vlastnú kópiu dát RPO.

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 právnických osôb. Žiadosť o vytvorenie iniciálnej dávky z RPO sprostredkuje prevádzka DCOM, následný spracovanie do IS mesta rieši dodávateľ ISM. 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 zaslaná v separátnej správe.

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

## Informácie o zmenách – zmenové dávky



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

Zmenové dávky sa skladajú zo sady zmenových súborov z RPO (novo pridané alebo zmenené osoby). Každý súbor zmenovej dávky bude zasielaný ako separátna správa.

Zoznam súborov sa získava volaním WS RPOChangedSubjectsService metóda listChangedSubjectsFiles služby RPO. Následne obsah jednotlivých súborov, teda jednotlivé XML sa uložia do messaging platformy a postupne sa spracúvajú.

Integračný modul RPO publikuje v messaging platforme jednotlivé súbory zmenovej dávky ako separátne správy resp. súbory vo formáte XML resp. JSON, ktoré si ďalej ISM spracováva a ukladá už vo svojej réžii.

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

# Integrácia ISM na messaging



Obrázok č.1: integračná schéma

Na obrázku je vidieť logickú schému jednotlivých integračných komponentov pre zmenové dávky registra právnických osôb. Zmenové dávky sa spracovávajú v backende IS DCOM a výsledné správy sú cez exchange 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 RPO sú publikované formou XML správ do exchange rpo, 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 právnických osôb 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 virtuálnom hoste “/rpo”.

## Načítanie zmenových správ

Odporúčaný proces načítania a spracovania správ pre ISM:

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

apmq://messaging.intra.dcom.sk:5671/rpo/<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 a súborov je totožná s XSD štruktúrou, ktorú definuje register právnických osôb aktuálne vo verzii 2.4.

Kvôli rozsahu jednotlivých schém ich pripájame vo forme príloh a pre ich pochopenie je potrebné sa obrátiť priamo na integračnú dokumentáciu systému RPO.

Hlavné xsd schémy dátovej štruktúry sú:

* <http://rpo.statistics.sk/RPO/Datatypes/rpo_core_schema-v2.4>
* <http://rpo.statistics.sk/RPO/Datatypes/Poskytnutie/Ext/Poskytnutie_udajov_PO_in-v2.4>
* <http://rpo.statistics.sk/RPO/Datatypes/Poskytnutie/Ext/Poskytnutie_udajov_PO_out-v2.4>
* <http://rpo.statistics.sk/RPO/Datatypes/Poskytnutie/Ext/Poskytnutie_IPO_out-v2.4>
* <http://rpo.statistics.sk/RPO/Datatypes/Poskytnutie/Ext/Poskytnutie_IPO_in-v2.4>
* <http://rpo.statistics.sk/RPO/Datatypes/Poskytnutie/Ext/Poskytnutie_IPO_rozsirene_out-v2.4>
* <http://rpo.statistics.sk/RPO/Datatypes/Poskytnutie/Ext/Poskytnutie_suborov_zmien_PO_out-v2.4>

Konkrétne je to element **GeneratedFileCType** ktorý je zloženým root elementom

<xs:complexType name="GeneratedFileCType">  
 <xs:annotation>  
 <xs:documentation>Informacie o synchronizacnej davke</xs:documentation>  
 </xs:annotation>  
 <xs:all>  
 <xs:element name="GeneratedFileName" type="cs:GeneratedFileNameType">  
 <xs:annotation>  
 <xs:documentation>Nazov suboru</xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element name="GeneratedFileFullPath" type="cs:GeneratedFileFullPathType">  
 <xs:annotation>  
 <xs:documentation>URL adresa, na ktorej je subor pristupny</xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element name="GeneratedFileType" type="cs:GeneratedFileTypeType">  
 <xs:annotation>  
 <xs:documentation>Typ suboru (NEW – nove, UPDATE- zmeny)</xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 <xs:element name="GeneratedFileCreation" type="cs:GeneratedFileCreationType">  
 <xs:annotation>  
 <xs:documentation>Datum a cas vytvorenia suboru</xs:documentation>  
 </xs:annotation>  
 </xs:element>  
 </xs:all>  
</xs:complexType>

Hlavné elementy dátovej štruktúry sú:

* GeneratedFileName – názov súboru zmenovej dávky [**xs:string dĺžka 1 - 255**]
* GeneratedFileFullPath – celá cesta k súboru zmenovej dávky [**xs:string dĺžka 1 - 1023**]
* GeneratedFileTypeType – typ súboru zmenovej dávky [**cs:GeneratedFileTypeType**]
* GeneratedFileCreation - dátum a čas vytvorenia súboru zmenovej dávky [**xs:dateTime**]

Na prenos správ cez RabbitMQ je použitá kompresia zip, čo je špecifikované v hlavičke správy atribútom **content\_type** a hodnotou **application/zip.** Routing key je totožný s prvým znakom typu súboru (“**I**” – iniciálna dávka, “**U**” – zmena, “N” – nový). Správa tiež v hlavičke obsahuje identifikátory z distribuovaného tracingu (X3-\*).

Hlavička správy navyše obsahuje aj kľúče:

* x-creation-date – totožné s hodnotou atribútu GeneratedFileCreation
* x-file-type – totožné s hodnotou atribútu GeneratedFileTypeType
* x-file-name - totožné s hodnotou atribútu GeneratedFileName

Príklad obsahu správy z RabbitMQ admin UI:



## Logický dátový model

Logický dátový model predstavuje entity využívané interným modulom IS DCOM. Poskytuje základný logický pohľad na štruktúru dát, umožňujúcu vybudovanie aplikačnej logiky systému. Logický dátový model je zároveň základom pre vytvorenie fyzického (databázového) modelu aplikácie, preto je obohatený niektorými tzv. stereotypmi, použiteľnými pri tvorbe nižších (implementačne detailnejších) vrstiev aplikácie. Uvádzame ho ako potenciálnu pomôcku pre ISM, akým spôsobom sú dáta z RPO spracované do logického dátového modelu.



Obrázok č. 2: Logický dátový model

## 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=rpo-test-user  
spring.rabbitmq.password=rpo-test-user  
spring.rabbitmq.virtual-host=/rpo

# 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 zip 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\_RPO\_CHANGES* = "rpo-ba-consumer";

static final Logger *logger* = LoggerFactory.*getLogger*(RaChangeListener.class);  
  
 @RabbitListener(queues = RabbitConfig.*QUEUE\_RPO\_CHANGES*)  
 public void onMessage(Message message) {  
 InputStream input = new ByteArrayInputStream(message.getBody());  
 MessageProperties msp = message.getMessageProperties();  
 if (msp != null && msp.getContentType() != null && "application/zip".equalsIgnoreCase(msp.getContentType()))  
 {  
 *LOG*.info("RECEIVED gzipped Message: trying to unpack it.");  
  
 try  
 {  
 Map<String, Object> header = msp.getHeaders();  
  
 String fileCreationDate = (String)header.get("x-creation-date");  
 String fileType = (String)header.get("x-file-type");  
 String fileName = (String)header.get("x-file-name");  
  
 ZipInputStream zis = new ZipInputStream(input);  
  
 try {

// spracovanie davky  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 zis.close();  
 } catch (IOException e) {  
 *LOG*.error(e.getMessage());  
 }

} else {  
 *LOG*.info("RECEIVED invalid content type message: " + msp.getContentType());  
 }

}

}

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

# Prílohy - schémy











