



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY

Bakalářská práce

Implementace modulu pro import údajů RÚIAN

Martin Schön



FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY

Bakalářská práce

Implementace modulu pro import údajů RÚIAN

Martin Schön

Vedoucí práce

Martin Bíkl, Ing. Petr Přibyl, Ing. Martin Zíma Ph.D.

© Martin Schön, 2024.

Všechna práva vyhrazena. Žádná část tohoto dokumentu nesmí být reprodukována ani rozšiřována jakoukoli formou, elektronicky či mechanicky, fotokopírováním, nahráváním nebo jiným způsobem, nebo uložena v systému pro ukládání a vyhledávání informací bez písemného souhlasu držitelů autorských práv.

Citace v seznamu literatury:

SCHÖN, Martin. *Implementace modulu pro import údajů RÚIAN*. Plzeň, 2024. Bakalářská práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd, Katedra informatiky a výpočetní techniky. Vedoucí práce Martin Bíkl, Ing. Petr Příbyl, Ing. Martin Zíma Ph.D.

Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení: **Martin SCHÖN**
Osobní číslo: **A22B0144P**
Adresa: **Rpety 42, Rpety, 26801 Hořovice, Česká republika**

Téma práce: **Implementace modulu pro import údajů RÚIAN**
Téma práce anglicky:
Jazyk práce: **Čeština**

Související osoby: **Ing. Martin Zíma, Ph.D. (Konzultant z univerzity)**
Katedra informatiky a výpočetní techniky
Ing. Martin Bíkl (Konzultant mimo univerzitu)
Katedra informatiky a výpočetní techniky
Ing. Petr Příbyl (Vedoucí)
Katedra informatiky a výpočetní techniky

Zásady pro vypracování:

1. Prostudujte datové schéma registru RÚIAN a možnosti získávání dat prostřednictvím datových služeb.
2. Prozkoumejte možnosti konfigurace řešení s přihlédnutím na mapování datových struktur.
3. Navrhněte konfigurační soubor, který bude umožňovat nastavení úrovně přenášených územních objektů a nastavení cílové databáze a cílových struktur.
4. Vytvořte aplikaci, která bude pravidelně synchronizovat veřejnou databázi RÚIAN do databázových struktur podle konfiguračního souboru. Synchronizace bude probíhat buď jako kompletní sada dat nebo přírůstkově. Jako úložiště využijte databázi Oracle, Microsoft SQL Server a PostgreSQL v posledních verzích.
5. Vytvořenou aplikaci ověřte na 3 konfiguračních souborech, zhodnoťte využitelnost daného řešení pro další databázové enginy a otestujte rychlost daného řešení pro kompletní i přírůtkovou sadu dat.

Seznam doporučené literatury:

Dodá vedoucí bakalářské práce.

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum:

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů. Tato práce nebyla využita k získání jiného nebo stejného akademického titulu.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Západočeská univerzita v Plzni má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Plzni dne 31. prosince 2024

.....

Martin Schön

V textu jsou použity názvy produktů, technologií, služeb, aplikací, společností apod., které mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Abstrakt

Text abstraktu v jazyce práce, tj. zde česky.

Abstract

The abstract text in a secondary language, here in English.

Klíčová slova

Poděkování

Text poděkování.

Obsah

1	Úvod	3
2	RÚIAN	5
2.1	Co je to RÚIAN	5
2.2	VFR	5
2.3	Tabulky	6
2.4	Uložení dat	7
3	Databáze	9
3.1	Co je to databáze	9
3.2	Microsoft SQL Server	10
3.3	PostgreSQL	10
3.4	Oracle	10
3.5	Komunikace s databází	11
4	Konfigurační soubor	13
4.1	Co je to konfigurační soubor	13
4.2	Formát	13
5	Technologie	15
5.1	Rest API	15
5.2	Spring Framework	15
5.3	Plánovač	16
5.4	Docker	16
A	První příloha	17
	Bibliografie	19

Problematika správy prostorových dat a jejich synchronizace mezi různými systémy nabývá na významu s rostoucí digitalizací státní správy a soukromých sektorů. Jedním z klíčových zdrojů těchto dat v České republice je Registr územní identifikace, adres a nemovitostí (RÚIAN), který poskytuje rozsáhlé a aktuální informace o územních objektech, adresách a dalších klíčových entitách. Efektivní využití dat z RÚIAN vyžaduje nejen jejich přístup prostřednictvím datových služeb, ale také robustní řešení pro mapování, konfiguraci a synchronizaci datových struktur.

Cílem této bakalářské práce je analyzovat datové schéma registru RÚIAN a možnosti získávání dat prostřednictvím nabízených datových služeb. Dále bude provedena analýza a návrh konfiguračního řešení, které umožní nastavit úroveň přenášených územních objektů a cílové databázové struktury. V rámci práce bude navržena a implementována aplikace, která umožní pravidelnou synchronizaci dat z veřejné databáze RÚIAN do cílových databázových struktur s podporou databází Oracle, Microsoft SQL Server a PostgreSQL. Aplikace bude schopna provádět synchronizaci kompletních datových sad i přírůstkových změn podle zadané konfigurace.

2.1 Co je to RÚIAN

RÚIAN je zkratkou pro Registr územní identifikace adres a nemovitostí. Jedná se o státní informační systém v České republice, který obsahuje informace o adresách, budovách, parcelách a dalších objektech. Systém je spravován Českým úřadem zeměměřickým a katastrálním (ČÚZK). Data jsou využívána v mnoha oblastech, například v urbanistickém plánování, geodézii, nebo při správě nemovitostí. Jednotlivé prvky jsou zobrazovány na mapách státního mapového díla a digitální mapě veřejné správy.

Data z RÚIAN jsou veřejně dostupná a lze je získat z webové služby na adrese <https://vdp.cuzk.cz/vdp/ruian>. V této aplikaci je možné vyhledávat konkrétní prvky nebo ověřit jejich existenci. Data jsou poskytována ve formátu XML a lze je automaticky stahovat pomocí API. Tyto informace jsou pravidelně aktualizovány, což zajišťuje jejich aktuálnost a přesnost.

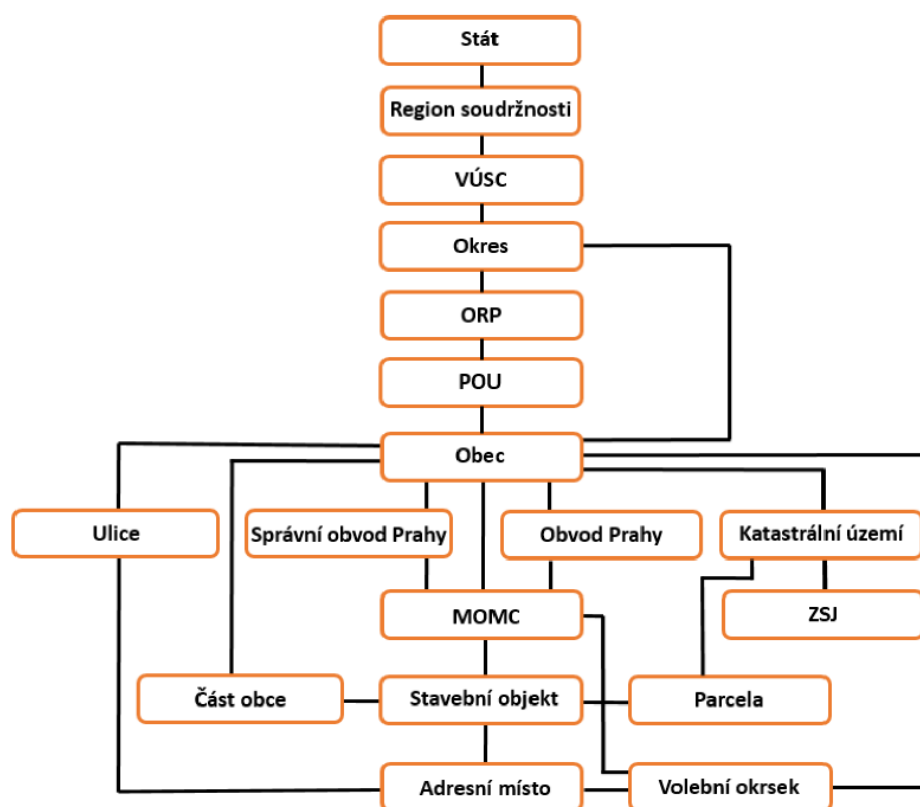
2.2 VFR

VFR neboli Výměnný formát RÚIAN je služba poskytující data. Je možné stahovat data dle zadaných formátů: Standardní, Historický a Speciální. Dále je možné si vybrat přírůstková data nebo úplnou kopii. Každý formát obsahuje dodatečně parametry, které je možné nastavit. Data z VFR jsou ve formátu XML. Každý element obsahuje atributy, které obsahují informace o dané entitě (Tabulce).

- **Standardní** – obsahuje úplná nebo přírůstková data.
 - Časový rozsah: Přírůstky od data / Úplná kopie
 - Územní prvky: Stát až ZJS / Obec a podřadné
 - Datová sada: Základní / Kompletní
 - Výběr z údajů: Základní údaje / Gen. hranice, Originální hranice, Vlajky a znaky
 - Územní omezení: ČR / Kraj (VÚSC) / ORP / Obec
- **Historický** – obsahuje historická data.
 - Časový rozsah: Přírůstky od data / Úplná kopie
 - Územní prvky: Stát až ZJS / Obec a podřadné
 - Územní omezení: ČR / Kraj (VÚSC) / ORP / Obec
- **Speciální** – obsahuje speciální data.
 - Časový rozsah: Přírůstky od data / Úplná kopie
 - Výběr z údajů: Číselníky / Vazby / Vazby a číselníky
 - Kategorie: Všechny / Geodetické body / Nerostné bohatství

2.3 Tabulky

Data z RÚIAN jsou rozdělena do několika tabulek. Jak je vidět na obrázku 2.1 [1], každá tabulka obsahuje jiné informace. Některé tabulky obsažené v RÚIAN jsou nepotřebné. Příkladem může být tabulka *Stát*, která obsahuje informace o státu Česká republika. Ovšem RÚIAN obsahuje pouze informace o České republice, tudíž tato tabulka je nadbytečná. Které tabulky budou zpracovány a které budou ignorovány, záleží na specifikaci v XSD (XML Schema Definition) souborech v dokumentaci VFR.



Obrázek 2.1: Tabulky RÚIAN

2.4 Uložení dat

Vzhledem k formátu dat získávaných z VFR je nezbytné zvolit vhodný způsob jejich ukládání, který zajistí efektivní organizaci, správu a manipulaci s daty. Nejlepší možností se jeví využití relační databáze, která poskytuje robustní nástroje pro ukládání dat s pevně definovanou strukturou a jejich vzájemné propojení prostřednictvím klíčů.

3.1 Co je to databáze

Databáze je softwarový nástroj, který slouží k efektivnímu ukládání, organizaci a vyhledávání dat. Díky své strukturované povaze umožňuje správu velkých objemů dat a poskytuje funkcionality pro zajištění konzistence, bezpečnosti a rychlého přístupu k uloženým informacím.

Databáze lze obecně rozdělit do dvou hlavních kategorií: **Relační databáze** a **Objektové databáze**. Každý z těchto přístupů má své specifické vlastnosti a je vhodný pro odlišné typy aplikací.

- Relační databáze
 - Data jsou uložena v tabulkách
 - Každý řádek tabulky obsahuje jeden záznam
 - Každý sloupec tabulky obsahuje jeden atribut
 - Vztahy mezi tabulkami jsou definovány klíči
 - Využití SQL jazyka

Relační databáze jsou vhodné pro strukturovaná data, která mají pevnou strukturu. Jedná se o nejčastěji používaný typ databáze.

- Objektové databáze
 - Data jsou uložena jako objekty
 - Každý objekt obsahuje atributy a metody
 - Vztahy mezi objekty jsou definovány referencemi
 - Využití objektově orientovaného jazyka

Objektové databáze jsou vhodné pro nestrukturovaná data, která mají složitou strukturu. Jedná se o novější typ databáze, který je vhodný pro moderní aplikace.

Jaká je tedy vhodná databáze pro uložení dat z RÚIAN VFR? Vzhledem k tomu, že data z VFR mají pevnou strukturu a jsou vzájemně propojena klíči, je nejlepší volbou relační databáze. Pro zpracování dat z RÚIAN VFR je tedy vhodná relační databáze. Na relační databázi bude třeba vytvořit schéma dle dříve zmíněných tabulek a zvolení databázového systému. Pro účely této práce byly zvoleny 3 následující databázové systémy: Microsoft SQL Server, PostgreSQL a Oracle.

3.2 Microsoft SQL Server

Microsoft SQL Server je relační databázový systém, který vyvinula společnost Microsoft a který se stal jedním z předních nástrojů pro ukládání, správu a analýzu dat. SQL Server je robustní a výkonný systém, který nabízí širokou škálu funkcí a možností přizpůsobení pro různé typy aplikací. Díky své dlouhodobé podpoře a integraci s dalšími produkty společnosti Microsoft, jako je Azure nebo Power BI, je SQL Server oblíbenou volbou pro velké a střední podniky.

Edice SQL Serveru zahrnují Standard, Enterprise a Express, které se liší funkcemi a cenou. Standard Edition je nejčastěji používanou edicí, která obsahuje všechny základní funkce a je vhodná pro většinu aplikací.

SQL Server byl původně navržen výhradně pro Windows, ale od verze SQL Server 2017 je dostupný také pro operační systém Linux. Tato multiplatformní podpora zvyšuje jeho použitelnost v různých IT prostředích. [2]

3.3 PostgreSQL

PostgreSQL je open-source relační databázový systém, který je známý svou spolehlivostí, výkonem a rozšiřitelností. Původně byl vyvinut jako alternativa k proprietárním řešením, jako je SQL Server, a dnes se řadí mezi nejpokročilejší relační databázové systémy na trhu. Díky své otevřené povaze a aktivní komunitě uživatelů a vývojářů se stal oblíbenou volbou nejen mezi malými firmami, ale také ve středních a velkých organizacích.

PostgreSQL je dostupný zdarma a podporuje všechny hlavní operační systémy, včetně Windows, Linux a macOS. Tato multiplatformní dostupnost umožňuje snadnou integraci PostgreSQL do různých vývojových prostředí. [3]

3.4 Oracle

Oracle Database, vyvinutá společností Oracle Corporation, patří mezi přední relační databázové systémy na světě. Tento systém je známý svou robustností, vysokým výkonem a schopností zvládat kritické podnikové aplikace a rozsáhlé datové sady.

Oracle Database je navržena tak, aby poskytovala spolehlivé a efektivní řešení pro ukládání, správu a analýzu dat ve velkých i středních organizacích.

Edice Oracle Database zahrnují Standard Edition, Enterprise Edition a Express Edition, které se liší funkcemi a cenou. Enterprise Edition je nejkomplexnější edicí, která obsahuje všechny pokročilé funkce a je vhodná pro velké podniky s náročnými požadavky.

Oracle Database je kompatibilní s většinou hlavních operačních systémů, včetně Windows, Linux a Unix. Díky tomu může být nasazena v různých IT prostředích podle požadavků organizace. [4]

3.5 Komunikace s databází

Všechny tři databázové systémy podporují komunikaci pomocí SQL jazyka. SQL (Structured Query Language) je standardizovaný jazyk pro práci s relačními databázemi, který umožňuje vytváření, čtení, aktualizaci a mazání dat. Pro komunikaci s databází je tedy třeba vytvořit SQL dotazy, které budou provádět operace nad daty.

Při implementaci je možné využít knihovny a frameworky, které usnadňují práci se SQL dotazy, jako například JDBC pro Javu, psycopg2 pro Python nebo Active Record v Ruby. Tyto nástroje nabízejí přístup k databázi, které usnadňují složitější operace.

Další možností je využití ORM (Object-Relational Mapping), který umožňuje mapování objektů na tabulky v databázi. Mezi populární ORM nástroje patří Hibernate pro Javu, SQLAlchemy pro Python nebo Entity Framework pro .NET. Tyto nástroje umožňují programátorům pracovat s databází více intuitivním způsobem, například pomocí objektů místo přímého psaní SQL dotazů. ORM také usnadňuje správu databázových transakcí a poskytuje nástroje pro migrace, které zajišťují snadnou aktualizaci struktury databáze v průběhu času.

Pro účely této práce bude využit ORM Hibernate, který umožňuje mapování objektů na tabulky v databázi a JDBC pro komunikaci s databází.

Konfigurační soubor

4

4.1 Co je to konfigurační soubor

Konfigurační soubor je soubor, který obsahuje nastavení aplikace. V konkrétním případě se jedná o nastavení:

- Databáze – Connection string
- Zdroj – Webová služba
- Výčet tabulek + mapování
- Výčet sloupců tabulek + mapování
- Plánovač – Interval stahování
- Vytvoření databáze – ANO/NE

Dále bude třeba vybrat vhodný formát pro konfigurační soubor.

4.2 Formát

Pro konfigurační soubor je možné zvolit několik formátů:

- XML(XSD) – Extensible Markup Language
- JSON – JavaScript Object Notation
- YAML – YAML Ain't Markup Language

Bude třeba víceúrovňový formát, který umožní snadné čtení a zápis. XML je zbytečně složité a nepřehledné. JSON je jednoduchý a přehledný, ale nepodporuje komentáře. YAML je jednoduchý a přehledný, podporuje komentáře, ale je méně rozšířený. Pro účely této práce byl zvolen formát JSON. [5].

Některé technologie pro tuto práci již byly zmíněny (Java, JDBC, Hibernate, Quartz, atd.). Pro účely této práce budou třeba ještě další technologie, které umožní tvorbu aplikace.

5.1 Rest API

REST API (Representational State Transfer Application Programming Interface) je architektonický styl pro návrh webových služeb, který umožňuje snadnou a efektivní komunikaci mezi klientem a serverem. Tento přístup se stal jedním z nejrozšířenějších způsobů pro integraci aplikací díky své jednoduchosti, flexibilitě a nezávislosti na platformě. REST API využívá standardní metody protokolu HTTP, jako jsou GET, POST, PUT a DELETE, k provádění různých operací s daty.

REST API poskytuje ideální prostředí pro implementaci přenosu dat díky své flexibilitě a schopnosti pracovat s různými datovými zdroji. V této práci bude kladen důraz na robustnost a spolehlivost API, což zahrnuje zpracování chybových stavů, zabezpečení komunikace (například prostřednictvím HTTPS) a optimalizaci výkonu. [6]

5.2 Spring Framework

Spring Framework je open-source framework pro vývoj aplikací v jazyce Java. V tomto frameworku bude vytvořena aplikace, která bude vše spojovat dohromady. Spring Framework obsahuje mnoho modulů, které usnadňují vývoj aplikací, jako například Spring Boot, Spring Data, Spring Security, atd. Pro účely této práce bude využit modul Spring Boot, který umožňuje rychlé vytvoření aplikace s minimální konfigurací. [7]

5.3 Plánovač

V popisu, co bude potřebovat konfigurační soubor bylo zmíněno, že bude třeba určit plánovač. Co je to plánovač? Plánovač je nástroj, který umožňuje spouštění úloh v pravidelných intervalech. Výhodou plánovače je, že umožňuje automatické spouštění úloh bez nutnosti manuálního zásahu. Je zde několik možných plánovačů, které lze použít:

- Cron – Unix
- Task Scheduler – Windows
- Quartz – Java
- Apache Airflow – Python

Všechny tyto plánovače umožňují spouštění úloh v pravidelných intervalech. Pro účely této práce byl zvolen plánovač Quartz, který je napsán v jazyce Java a umožňuje spouštění úloh v pravidelných intervalech. Intervaly mohou být nastaveny v cron notaci, což umožňuje velkou flexibilitu při plánování úloh (každý den v 3:00, každý týden v pondělí v 8:00, každý měsíc první den v 12:00 atd.).

5.4 Docker

Docker je open-source platforma pro vývoj, nasazení a provoz aplikací. Dále umožňuje vytváření kontejnerů, které obsahují všechny potřebné závislosti pro běh aplikace. Proč je třeba docker? Pro každou databázi je třeba vytvořit instanci, která bude obsahovat všechny potřebné tabulky, data a klienta pro komunikaci a práci s databází. Na to se hodí docker, který umožňuje vytvoření kontejneru s databází a klientem, který bude obsahovat všechny potřebné závislosti pro běh aplikace. Dále zde může běžet i plánovač, který bude spouštět úlohy v pravidelných intervalech.

[8]

První příloha

A

Bibliografie

1. CZBAP-127. VFR. 2023. Dostupné také z: [https://cuzk.gov.cz/ruian/Poskytovani-udaju-ISUI-RUIAN-VDP/Vymenny-format-RUIAN-\(VFR\)/DL058RR2-v5-0-Struktura-a-popis-VFR_final.aspx](https://cuzk.gov.cz/ruian/Poskytovani-udaju-ISUI-RUIAN-VDP/Vymenny-format-RUIAN-(VFR)/DL058RR2-v5-0-Struktura-a-popis-VFR_final.aspx).
2. MICROSOFT. Microsoft SQL Server. *Microsoft Documentation*. 2023. Dostupné také z: <https://docs.microsoft.com/en-us/sql/sql-server/>.
3. GROUP, PostgreSQL Global Development. PostgreSQL. *PostgreSQL Documentation*. 2023. Dostupné také z: <https://www.postgresql.org/docs/>.
4. ORACLE. Oracle Database. *Oracle Documentation*. 2023. Dostupné také z: <https://docs.oracle.com/en/database/>.
5. COMMUNITY, Cisco. *XML vs JSON vs YAML*. 2022. Dostupné také z: <https://community.cisco.com/t5/devnet-general-knowledge-base/xml-vs-json-vs-yaml/ta-p/4729758>.
6. FIELDING, Roy T. *Architectural Styles and the Design of Network-based Software Architectures*. 2000. Dostupné také z: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm.
7. SPRING. *Spring Framework Documentation*. 2023. Dostupné také z: <https://spring.io/projects/spring-framework>.
8. DOCKER. *What is Docker?* 2023. Dostupné také z: <https://www.docker.com/what-docker>.

1101001 1100001
10101100001110010 1100001
101011010101 10



11010011101101001
0110000110101
111000101011101