

LinkSoft - Java test

Popis projektu:

Projekt je založen na databázi rozsahů IP adres přiřazených k městům v ČR a SK. Tato data jsou u projektu přibalena ve formě CSV. Aplikace by měla při spuštění data z přiložených CSV souborů přečíst a uložit do své embedded databáze.

Projekt dále obsahuje package `cz.linksoft.hr.test.api`, který obsahuje definici rozhraní aplikace včetně vlastního doménového modelu. Vaším úkolem bude vytvořit implementaci zde umístěných rozhraní.

K dispozici je REST služba (RestFacade), která v jednoduchosti zprostředkovává přístup k implementaci rozhraní zmíněného výše. Požadováno je, aby JSON výstup z této služby zůstal i po vašich veškerých změnách naprosto nezměněn - je povoleno činit úpravy na modelu, ale ne na serializovaném JSON výstupu z něj.

Pro spuštění aplikace budete potřebovat Javu 8 a Maven. Spustit jí lze buď přes IDE a nebo přes Maven a to pomocí příkazu `mvn spring-boot:run`. Po nastatování je možné k REST službám přistoupit přes adresu <http://localhost:8080>.

Závislosti deklarované v `pom.xml` jsou nezávazné. Projekt si můžete pozměnit dle vaší potřeby.

Zadání

1. Vytvoření JPA entit a databázového modelu

- aplikace obsahuje embedded databázi H2, po startu je k dispozici konzole na adrese <http://localhost:8080/h2-console/>
- aplikace při startu automaticky vytvoří prázdnou databázi dle JPA entit
(`hibernate.hbm2ddl.auto=create-drop`)
- je na vašem zvážení, zdali pro namapování použijete model v `cz.linksoft.hr.test.api.model` , nebo vytvořte vlastní
- rozhraní z pohledu serializace do JSON u REST služeb by mělo za každých okolností zůstat zachováno

2. Přečtení přibalených CSV souborů a naplnění embedded databáze po spuštění aplikace

- CSV soubory se nachází v komprimované podobě (GZIP) v `src/main/resources/data/`
- pro vylistování souborů ke čtení můžete použít třídu `ApplicationDataResourceProvider`
- pro čtení CSV prosím nepoužívejte žádnou knihovnu třetí strany
- data jsou uložena v denormalizované podobě, při čtení je tedy nutné data svépomocí namapovat do hierarchické struktury `IpAddressRange -> City ->`

Region -> Country

- formát CSV souborů je shrnut v sekci “Popis formátu CSV souborů”

3. Implementace služeb

- package `cz.linksoft.hr.test.core.service` již obsahuje připravené třídy, které čekají na implementaci dle dokumentace uvedené v rozhraní
- LocationService zprostředkovává jednoduchý přístup k datům
- IPAddressService pracuje s rozsahy IP adres asociovanými s městy

4. Traffic limit pro RESTové služby

- přístup k RESTovým službám je třeba omezit a to následovně
 - pro vyhodnocení se použije IP adresa pro každý příchozí požadavek ~
`ServletRequest.getRemoteAddr()`
 - pokud bude překročen povolený traffic, je požadováno odeslání prázdné odpovědi s HTTP statusem `429 Too Many Requests`
 - akceptováno smí být pouze:
 - X požadavků z daného města za poslední hodinu
 - Y požadavků z daného regionu za poslední hodinu

- Z požadavků z daného státu za poslední hodinu
- význam spojení “za poslední hodinu” smí být v rámci optimalizace přiměřeně zjednodušen
- proměnné X, Y a Z mohou být nastaveny “na pevně” a bez rozdílu mezi jednotlivými městy, regiony a státy

5. Další nepovinné úkoly

1. REST služby - korektní řešení chybových stavů
2. čtení CSV - podpora umístění oddělovačů řádků a hodnot v textu ohraničeném uvozovkami (v tomto případě neplní úlohu oddělovače, ale jedná se o součást textu)
3. čtení CSV - podpora escapování uvozovek zpětným lomítkem (zpětné lomítko v textu ohraničeném uvozovkami znamená zrušení případného speciálního významu následujícího znaku, tj. “a”\b” by se mělo přečíst jako: a”b)
4. čtení CSV - chyba vzniklá při čtení z daného řádku by měla odkazovat na číslo daného řádku/záznamu

Popis formátu CSV souborů

CSV soubory jsou v ASCII kódování, pro jistotu však pro čtení použijte UTF-8. Soubory neobsahují žádnou “hlavičku” a na konci mohou obsahovat libovolné množství prázdných řádků.

CSV má tyto parametry:

- oddělovač řádků je \n
- oddělovač hodnot je ,
- text může být obalen v uvozovkách (znak ")

Popis sloupců:

Název	Datový typ	Popis
ip_from	INT (10)	First IP address in netblock.
ip_to	INT (10)	Last IP address in netblock.
country_code	CHAR(2)	Two-character country code based on ISO 3166.
country_name	VARCHAR(64)	Country name based on ISO 3166.
region_name	VARCHAR(128)	Region or state name.
city_name	VARCHAR(128)	City name.
GPS latitude	DOUBLE	City latitude. Default to capital city latitude if city is unknown.
GPS longitude	DOUBLE	City longitude. Default to capital city longitude if city is unknown.

IP adresy jsou uloženy jako "IP Number" takto:

IP Address = w.x.y.z

IP Number = $256^3 \cdot w + 256^2 \cdot x + 256 \cdot y + z$

$w = \text{int} (\text{IP Number} / 256^3) \% 256$

```
x = int ( IP Number / 256^2 ) % 256  
y = int ( IP Number / 256 ) % 256  
z = int ( IP Number ) % 256
```

(kde ^ znamená umocnění a % zbytek po vydělení)

Rozsah IP adres je v CSV uložen ve formě intervalu `<ip_from;`
`ip_to)`.

Zdrojová data CSV souborů byla stažena z adresy
<http://lite.ip2location.com/>, databáze `DB5.LITE`, typ `IPV4 CSV`.

Na závěr...

Úloha bude vyhodnocena na základě těchto kritérií:

- dodržení funkčních požadavků
- míra dodržení zadání
- kvalita napsaného kódu
- kvalita otestování

Pokud byste se rozhodl/a některé požadavky ze zadání nesplnit a nebo je splnit jen částečně, popište nám prosím co konkrétně a proč.

Děkujeme a přejeme vám hodně štěstí :-)