



109 lines (81 loc) · 6.45 KB

Preview

Code

Blame

Raw



Zadání úkolu 2

Tvým úkolem je vytvořit program, který bude získávat data z obchodního rejstříku s využitím jeho REST API.

Část 1

V této části vyhledej informace o konkrétním subjektu na základě jeho identifikačního čísla (IČO). Toto číslo je jedinečným identifikátorem subjektu, pro každé číslo tedy rejstřík vrátí informace pouze o jednom subjektu. Nejprve se pomocí funkce `input()` zeptej uživatele nebo uživatky, o kterém subjektu chce získat informace. S využitím modulu `requests` odešli GET požadavek na adresu <https://ares.gov.cz/ekonomicke-subjekty-v-be/rest/ekonomicke-subjekty/ICO>, kde `ICO` nahraď číslem, které zadal(ka) uživatel(ka) (např. <https://ares.gov.cz/ekonomicke-subjekty-v-be/rest/ekonomicke-subjekty/22834958>). S adresou pracuj jako s obyčejným řetězcem, tj. můžeš využívat formátované řetězce, metodu `.replace()`, operátor `+` atd. Text, který API vrátí, převed' na JSON a zjisti z něj obchodní jméno subjektu a adresu jeho sídla (můžeš využít podle `textovaAdresa`). Získané informace vypiš na obrazovku.

Například pro IČO 22834958 by tvůj program měl vypsát následující text.

Czechitas z.ú.

Krakovská 583/9, Nové Město, 110 00 Praha 1



Část 2

Často se stane, že neznáme IČO subjektu, ale známe například jeho název nebo alespoň část názvu. Napiš program, který se zeptá uživatele(ky) na název subjektu, který chce vyhledat. Následně vypiš všechny nalezené subjekty, které ti API vrátí.

V případě vyhledávání musíme odeslat požadavek typu POST na adresu

<https://ares.gov.cz/ekonomicke-subjekty-v-be/rest/ekonomicke-subjekty/vyhledat>.

Request typu POST pošleme tak, že namísto funkce `requests.get()` použijeme funkci `requests.post()`. K requestu musíme přidat hlavičku (parametr `headers`), který určí formát výstupních dat. Použij slovník níže.

```
headers = {  
    "accept": "application/json",  
    "Content-Type": "application/json",  
}
```



Dále přidáme parametr `data`, do kterého vložíme řetězec, který definuje, co chceme vyhledávat. Data vkládáme jako řetězec, který má JSON formát. Pokud chceme například vyhledat všechny subjekty, které mají v názvu řetězec `"moneta"`, použijeme následující řetězec.

```
data = '{"obchodniJmeno": "moneta"}'
```



Níže je příklad odeslání requestu:

```
headers = {  
    "accept": "application/json",  
    "Content-Type": "application/json",  
}  
data = '{"obchodniJmeno": "moneta"}'  
res = requests.post("https://ares.gov.cz/ekonomicke-subjekty-v-be/rest/ekon
```



Tentokrát API vrátí počet nalezených subjektů (`pocetCelkem`) a seznam nalezených subjektů `ekonomickeSubjekty`. Tvůj program by měl vypsát obchodní jména všech nalezených subjektů a jejich identifikační čísla, výstupy odděluj čárkou. Příklad výstupu pro `"moneta"` je níže.

```
Nalezeno subjektů: 13  
MONETA PARTNERS s.r.o., 01590952  
Moneta Sinkovská, 05170443  
Nadace MONETA Clementia, 10730443  
Juno Moneta, z.s., 22741461
```



```
Moneta Investment, s.r.o., 24227625
Moneta SPV, s. r. o. "v likvidaci", 25355163
MONETA Money Bank, a.s., 25672720
Moneta Praha s.r.o., 26424720
Moneta holding s.r.o., 28660463
JK MONETA, s.r.o., 29242746
MONETA Stavební Spořitelna, a.s., 47115289
MONETA Auto, s.r.o., 60112743
MONETA Leasing, s.r.o., 60751606
```

Ve tvém programu musíš nahradit řetězec `moneta` proměnnou, která obsahuje řetězec zadaný uživatelem.

Bonus

Ke každému subjektu je v databázi uložena jeho právní forma. Ta se nachází pod klíčem `pravniForma`. Není tam přímo název subjektu, ale číselný kód, jehož význam je uložený v tzv. číselníku. Pomocí požadavku na adresu <https://ares.gov.cz/ekonomicke-subjekty-v-be/rest/ciselniky-nazevniky/vyhledat> si můžeme stáhnout celý číselník a poté tam příslušný kód vyhledat. Přidej do programu požadavek na tuto adresu. Půjde o požadavek typu POST, parametr `headers` zůstane stejný a jako parametr `data` zadej:

```
data = '{"kodCiselniku": "PravniForma", "zdrojCiselniku": "res"}'
```



Číselník je v seznamu pod klíčem `ciselniky`. Dále použij počáteční hodnotu ze seznamu (dotaz vrátí pouze jeden číselník, v seznamu je tedy pouze jedna položka). Touto hodnotou je opět slovník, ve kterém je pod klíčem `polozkyCiselniku` seznam všech kódů a jejich hodnot.

Poté napiš funkci `find_legal_form`, která bude přijímat dva parametry - hledaný kód a seznam `polozkyCiselniku`. Například pro kód `"112"` by funkce měla vrátit řetězec `"Společnost s ručením omezeným"`.

Nyní uprav část programu, která vypisuje všechny aplikace podle názvu, aby spolu s obchodním jménem a identifikačním číslem vypsala i právní normu. Například pro `"moneta"` by výstup mohl vypadat takto:

```
Nalezeno subjektů: 13
MONETA PARTNERS s.r.o., 01590952, Společnost s ručením omezeným
Moneta Sinkovská, 05170443, Fyzická osoba podnikající dle živnostenského
zákona
Nadace MONETA Clementia, 10730443, Nadace
Juno Moneta, z.s., 22741461, Spolek
Moneta Investment, s.r.o., 24227625, Společnost s ručením omezeným
Moneta SPV, s. r. o. "v likvidaci", 25355163, Společnost s ručením
omezeným
```



MONETA Money Bank, a.s., 25672720, Akciová společnost
Moneta Praha s.r.o., 26424720, Společnost s ručením omezeným
Moneta holding s.r.o., 28660463, Společnost s ručením omezeným
JK MONETA, s.r.o., 29242746, Společnost s ručením omezeným
MONETA Stavební Spořitelna, a.s., 47115289, Akciová společnost
MONETA Auto, s.r.o., 60112743, Společnost s ručením omezeným
MONETA Leasing, s.r.o., 60751606, Společnost s ručením omezeným

Diakritika

Na splnění úkolu stačí, aby tvůj program dobře fungoval pro vyhledávání řetězců bez diakritiky.

Pokud bys chtěl(a) vyhledat nějaký název, který obsahuje diakritiku, je nutné řetězec zakódovat. K tomu slouží metoda `encode()`, Protože chceme použít kódování UTF-8, je třeba toto kódování doplnit do volání metody. Pokud název diakritiku neobsahuje, není to nutné.

```
data = '{"obchodniJmeno": "škoda"}'  
data = data.encode("utf-8")
```



Tip: Protože v hodnotě `data` jsou složené závorky, namísto formátovaných řetězců je jednodušší spojit řetězec dohromady z jednotlivých částí s využitím `+`. Alternativně můžeš využít metodu `.dump()`, která slovník uloží jako řetězec.