



Java alkalmazások gyakorlat beadandó

Holyba Attila Patrik

FLX1D8

Git User: Beantech0

A feladatot egyedül készítettem el de két Git usert használtam fel.

Github: https://github.com/Beantech0/Java_gyakorlat_beadando_spring.git

2024.11.30.

Tartalom

Fejlesztési és tesztelési környezet	3
Feladat - Adatbázis	3
Választott adatbázis	3
1. Feladat látványos weboldal	4
2. Regisztráció-Bejelentkezés-menüsor	4
3. Felhasználók	5
4. Adatmegjelenítés adatbázisból	6
5. Kapcsolat űrlap, üzenetküldés	6
6. Üzenetek megjelenítése	7
7. RESTful API elkészítés és tesztelés	7

Fejlesztési és tesztelési környezet

IntelliJ IDEA 2024.2.3 Ultimate Edition

Java verzió: OpenJDK '17.0.12' 2024-07-16

A jar fájl a PDF alapján lett elkészítve és a tesztkörnyezeten tesztelve.

Feladat - Adatbázis

Választott adatbázis

A választott adatbázis: 1-20-Városok-3 táblás.

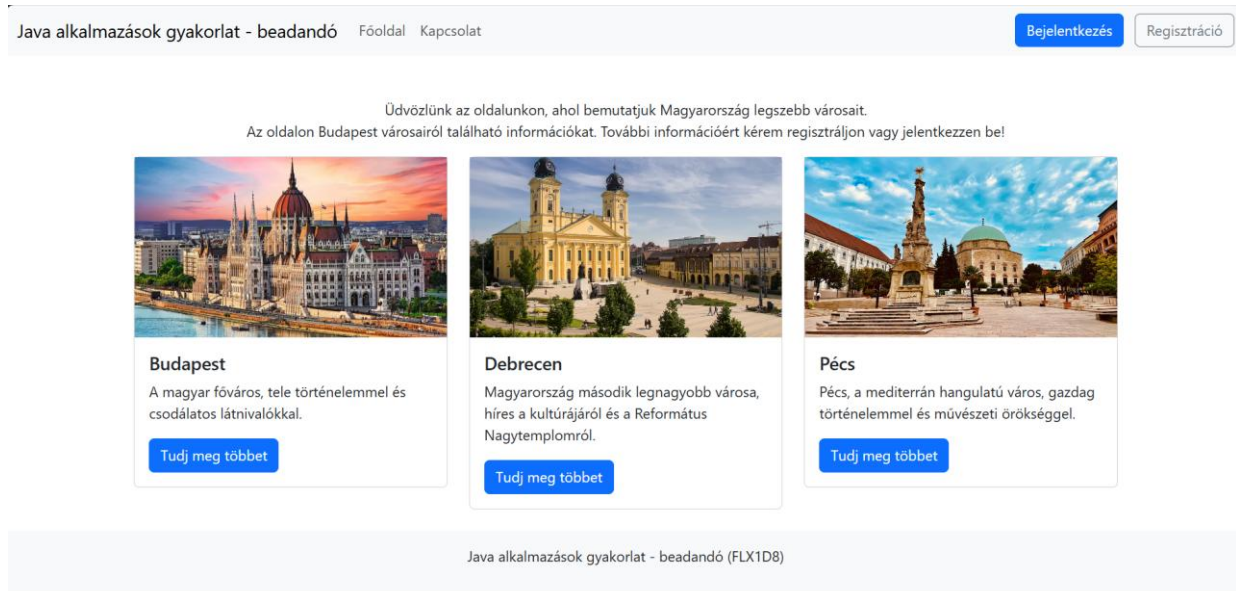
Az adatbázisban hazánk városairól találhatóak meg népességi és egyéb adatok.

A táblaszerkezet így néz ki.

feladat megye id : int(11) nev : text	feladat varos id : int(11) nev : text # megyeid : int(11) # megyeszekhely : tinyint(1) # megyeijogu : tinyint(1)	feladat lelekszam varosid : int(11) ev : int(11) # no : int(11) # osszesen : int(11)
--	--	---

1. Feladat látványos weboldal

A weboldalhoz bootstraet használtam a reszponzív felület elérése érdekében.



2. Regisztráció-Bejelentkezés-menüsor

A bejelentkezés-regisztráció és az egyéni menüsor a springbootsecurity3-al lett megoldva, az órán tanult megoldások alapján. A DB bővítve lett egy Users és egy Messages táblával, amikhez a repot is létrehoztam.

The image shows a "Please sign in" form on a light gray background. It consists of two input fields: "Username" and "Password". Below these fields is a blue button labeled "Sign in".

Regisztráció

Név

E-mail cím

Jelszó

Regisztráció

A regisztráció validálás a **“/registration_validation”** url alatt történik meg amihez a UserDetailsService springboot funkciók implementáltam.

3. Felhasználók

Bejelentkezés után Admin esetében elérhető a “Kapcsolati üzenetek” menüpont is.

Java alkalmazások gyakorlat - beadandó

Főoldal

Kapcsolat

Felhasználói Főoldal

Városok

Kapcsolati üzenetek

Bejelentkezve: admin@admin.hu

Logout

Örülök hogy bejelentkeztél, köszöntelek az oldalon!

Java alkalmazások gyakorlat - beadandó (FLX1D8)

Bejelentkezett felhasználó és admin esetében a Felhasználói Főoldal és a Városok menüpont is elérhető.

Java alkalmazások gyakorlat - beadandó

Főoldal

Kapcsolat

Felhasználói Főoldal

Városok

Bejelentkezve: user@user.hu

Logout

Örülök hogy bejelentkeztél, köszöntelek az oldalon!

Java alkalmazások gyakorlat - beadandó (FLX1D8)

Ezek mind a layout.html-ben a security modullal lettek megoldva és a WebSecurityConfig osztályban.

Létrehozott felhasználók:

Jogkör	Felhasználónév	Jelszó
User	user@user.hu	asd
Admin	admin@admin.hu	admin

4. Adatmegjelenítés adatbázisból

Java alkalmazások gyakorlat - beadandó

Főoldal

Kapcsolat

Felhasználói Főoldal

Városok

Bejelentkezve: user@user.hu

Logout

Város	Megyeszékhely	Megyei jogu?	Megye	Összes lakos átlag	Női lakos átlag
Mindszent2	Csongrád	false	false	7272.3158	3728.5263
Komádi	Hajdú-Bihar	false	false	6006.2222	3012.0556
Tiszacsege	Hajdú-Bihar	false	false	4945.7895	2501.5789
Nyíradony	Hajdú-Bihar	false	false	8006.8947	4093.7368
Rákóczi falva	Jász-Nagykun-Szolnok	false	false	5491.4	2784.6
Mezőkeresztes	Borsod-Abaúj-Zemplén	false	false	3954.3	2039.1
Nagyatád	Somogy	false	false	11539.7895	6190.0
Tatabánya	Komárom-Esztergom	true	true	71048.8947	37251.4737
Tótkomlós	Békés	false	false	6404.7368	3322.8947
Lenti	Zala	false	false	8434.7368	4366.5263
Körösladány	Békés	false	false	4753.4167	2457.0833
Barcs	Somogy	false	false	11695.6316	6118.2632

Az adatbázisból való adatmegjelenítést a “Városok” menüpont alatt oldottam meg. Ami meghívja a “VarosokData” függvényt ami egy formázott stringet ad vissza. Itt a varosRepo alapján lekérem az összes várost és kiíratom a város és megye adatokat. Továbbá a lélekszám táblára készítettem egy egyedi lekérdezést a LelekszamRepo interface-ben ami visszaadja az adott varosid alapján hogy mennyi volt az átlagos összes lakosság és a női lakosok átlaga az összes évet tekintve.

```
public interface LelekszamRepo extends CrudRepository<Lelekszam, Integer> {
    1 usage  Beantech0

    @Query("SELECT AVG(l.osszesen) FROM Lelekszam l WHERE l.varosid = :varosid GROUP BY l.varosid")
    String getAveragePopulationPerCity(@Param("varosid") int varosid);

    @Query("SELECT AVG(l.no) FROM Lelekszam l WHERE l.varosid = :varosid GROUP BY l.varosid")
    String getAverageWomenPerCity(@Param("varosid") int varosid);
}
```

Utána ezt a stringet adja vissza a varosok.html ahol a <p th:text="\$ {str}"/> résznél kiíratom a str értékét.

5. Kapcsolat űrlap, üzenetküldés

A kapcsolat menüpont alatt van lehetőség üzenetet küldeni az adminnak az alábbi formál.

Lépjen kapcsolatba velünk!

Írja ide üzenetét*

Submit Reset

Ha nincs bejelentkezve senki akkor Vendég néven fogja elmenteni az üzenetet egyébként pedig a user felhasználóneve alapján. A /kapcsolat url-ről a Submit gomb átküld a /kapcsolat_success URL-re ahol az urlapSubmit elmenti a Messages táblába az üzenetet. Az felhasználói adatok kiolvasása után a messagesRepo.save módszerével kerül mentésre az üzenet.

6. Üzenetek megjelenítése

Az üzenetek megtekintésére az adminnak van lehetősége a “Kapcsolati üzenetek” menüpont alatt.

Java alkalmazások gyakorlat - beadandó Főoldal Kapcsolat Felhasználói Főoldal Városok Kapcsolati üzenetek Bejelentkezve: admin@admin.hu Logout

Üzenetek:		
User	Dátum	Szöveg
Vendég	2024-12-01T15:28:09	Szia, vendég vagyok.
user@user.hu	2024-12-01T15:33:22	Szia, asd vagyok Monorról.
admin@admin.hu	2024-12-01T16:11:14	Szia, Admin vagyok.

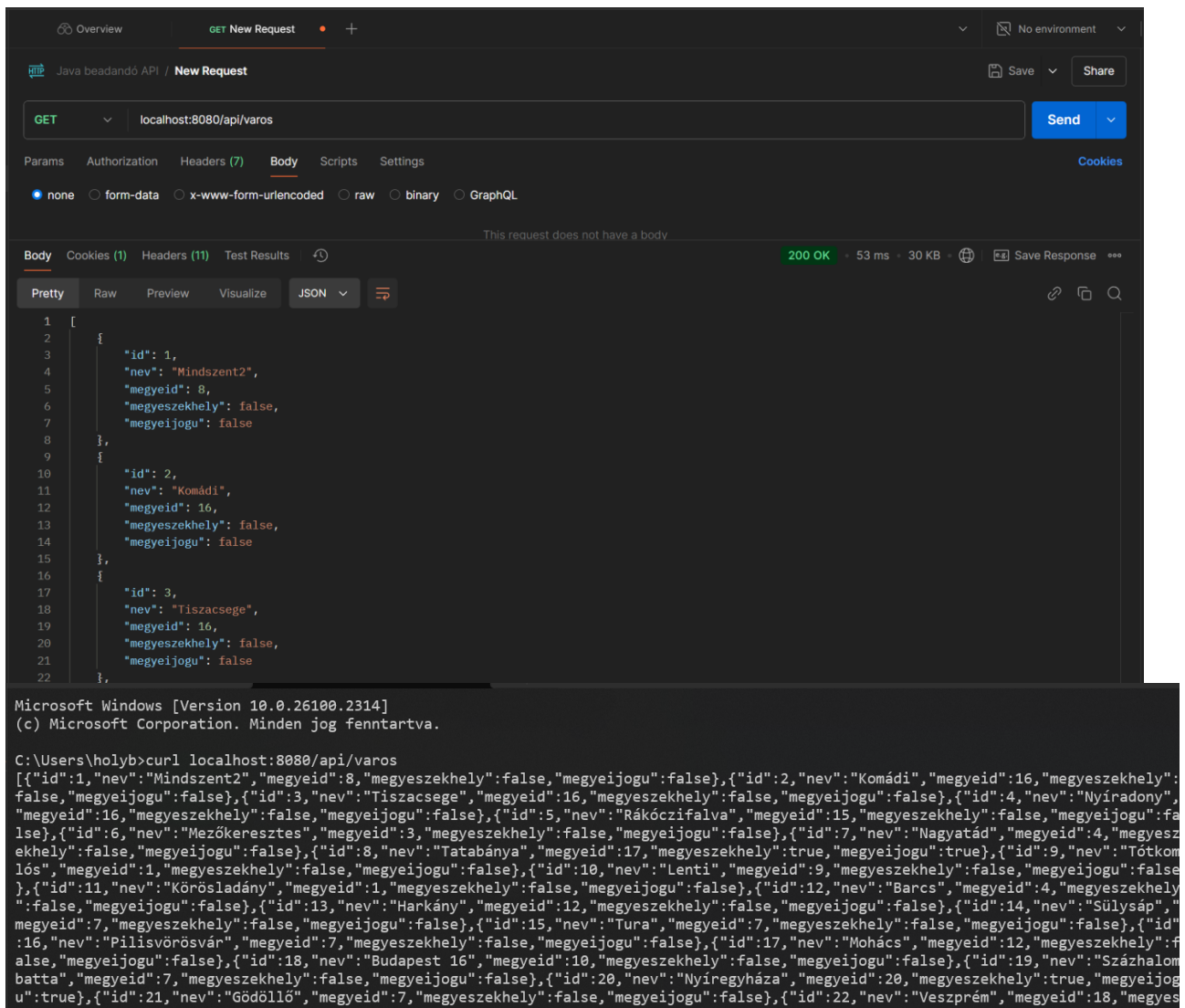
Java alkalmazások gyakorlat - beadandó (FLX1D8)

Itt hasonlóan a városok listázásához az AdminUzenetek függvény egy stringet ad vissza ahol lekérem az üzeneteket és formázott stringként megjelenítem a táblázat bodyban.

7. RESTful API elkészítés és tesztelés

A REST API hívásokat az órán tanult módszerrel készítettem el. Csináltam egy VarosokRestController osztályt aminél megadtam a @RestController kapcsolót. Az API az /API/ url alatt érhető el.

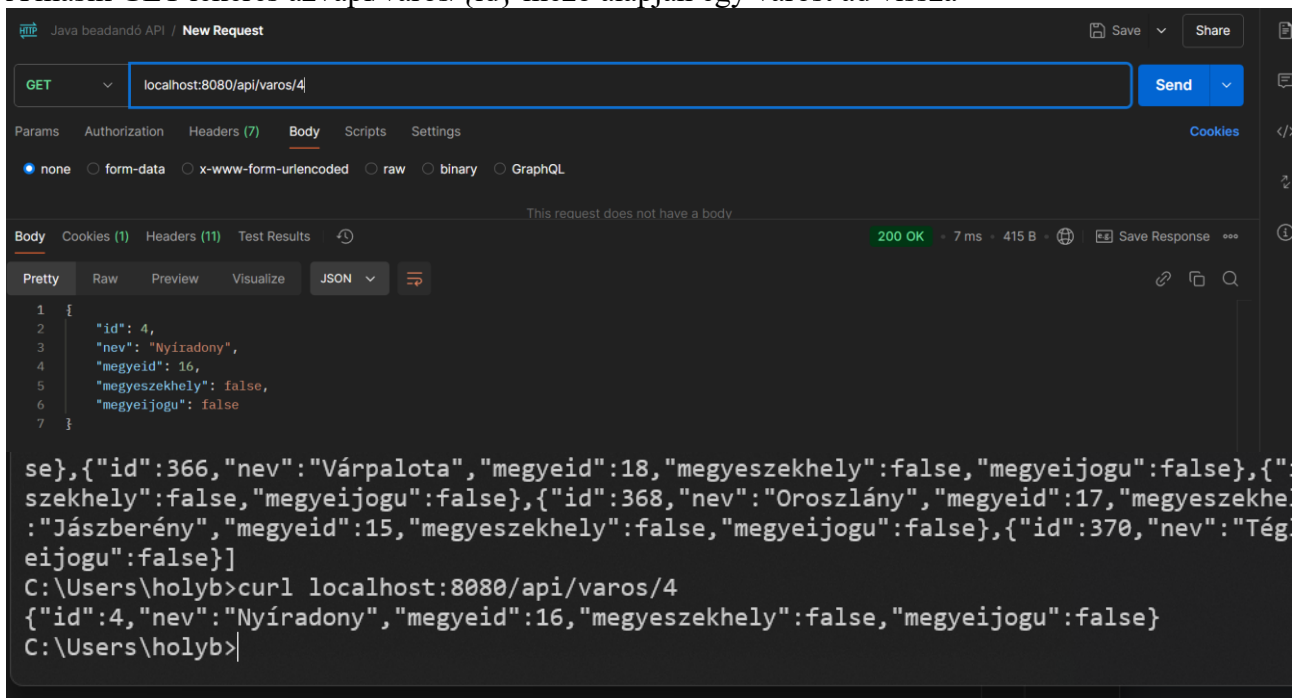
Elsőként az GET /api/varosok URI alatt listázom az összes várost a VarosRepo segítségével. A tesztelésről képernyőképeket Postmanről és CURL-ről beletettem.



The screenshot shows the Postman interface with a GET request to `localhost:8080/api/varos`. The response is a JSON array of 22 city objects, each containing `id`, `nev` (name), `megyeid` (county ID), `megyeszekhely` (capital), and `megyeijogu` (county seat) fields. The response status is 200 OK.

```
1 [
2   {
3     "id": 1,
4     "nev": "Mindszent2",
5     "megyeid": 8,
6     "megyeszekhely": false,
7     "megyeijogu": false
8   },
9   {
10    "id": 2,
11    "nev": "Komádi",
12    "megyeid": 16,
13    "megyeszekhely": false,
14    "megyeijogu": false
15  },
16  {
17    "id": 3,
18    "nev": "Tiszacsege",
19    "megyeid": 16,
20    "megyeszekhely": false,
21    "megyeijogu": false
22  },
23  {
24    "id": 4,
25    "nev": "Nyíradony",
26    "megyeid": 16,
27    "megyeszekhely": false,
28    "megyeijogu": false
29  },
30  {
31    "id": 5,
32    "nev": "Rákóczi",
33    "megyeid": 15,
34    "megyeszekhely": false,
35    "megyeijogu": false
36  },
37  {
38    "id": 6,
39    "nev": "Mezőkeresztes",
40    "megyeid": 3,
41    "megyeszekhely": false,
42    "megyeijogu": false
43  },
44  {
45    "id": 7,
46    "nev": "Nagyatád",
47    "megyeid": 4,
48    "megyeszekhely": false,
49    "megyeijogu": false
50  },
51  {
52    "id": 8,
53    "nev": "Tatabánya",
54    "megyeid": 17,
55    "megyeszekhely": true,
56    "megyeijogu": true
57  },
58  {
59    "id": 9,
60    "nev": "Tótkomlós",
61    "megyeid": 1,
62    "megyeszekhely": false,
63    "megyeijogu": false
64  },
65  {
66    "id": 10,
67    "nev": "Lenti",
68    "megyeid": 9,
69    "megyeszekhely": false,
70    "megyeijogu": false
71  },
72  {
73    "id": 11,
74    "nev": "Körösladány",
75    "megyeid": 1,
76    "megyeszekhely": false,
77    "megyeijogu": false
78  },
79  {
80    "id": 12,
81    "nev": "Barcs",
82    "megyeid": 4,
83    "megyeszekhely": false,
84    "megyeijogu": false
85  },
86  {
87    "id": 13,
88    "nev": "Harkány",
89    "megyeid": 12,
90    "megyeszekhely": false,
91    "megyeijogu": false
92  },
93  {
94    "id": 14,
95    "nev": "Sülysáp",
96    "megyeid": 7,
97    "megyeszekhely": false,
98    "megyeijogu": false
99  },
100 {
101   "id": 15,
102   "nev": "Tura",
103   "megyeid": 7,
104   "megyeszekhely": false,
105   "megyeijogu": false
106 },
107 {
108   "id": 16,
109   "nev": "Pilisvörösvár",
110   "megyeid": 7,
111   "megyeszekhely": false,
112   "megyeijogu": false
113 },
114 {
115   "id": 17,
116   "nev": "Mohács",
117   "megyeid": 12,
118   "megyeszekhely": false,
119   "megyeijogu": false
120 },
121 {
122   "id": 18,
123   "nev": "Budapest 16",
124   "megyeid": 10,
125   "megyeszekhely": false,
126   "megyeijogu": false
127 },
128 {
129   "id": 19,
130   "nev": "Százhalombatta",
131   "megyeid": 7,
132   "megyeszekhely": false,
133   "megyeijogu": false
134 },
135 {
136   "id": 20,
137   "nev": "Nyíregyháza",
138   "megyeid": 20,
139   "megyeszekhely": true,
140   "megyeijogu": true
141 },
142 {
143   "id": 21,
144   "nev": "Gödöllő",
145   "megyeid": 7,
146   "megyeszekhely": false,
147   "megyeijogu": false
148 },
149 {
150   "id": 22,
151   "nev": "Veszprém",
152   "megyeid": 18,
153   "megyeszekhely": true,
154   "megyeijogu": true
155 }
```

A másik GET lekérés az `/api/varos/{id}` mező alapján egy várost ad vissza



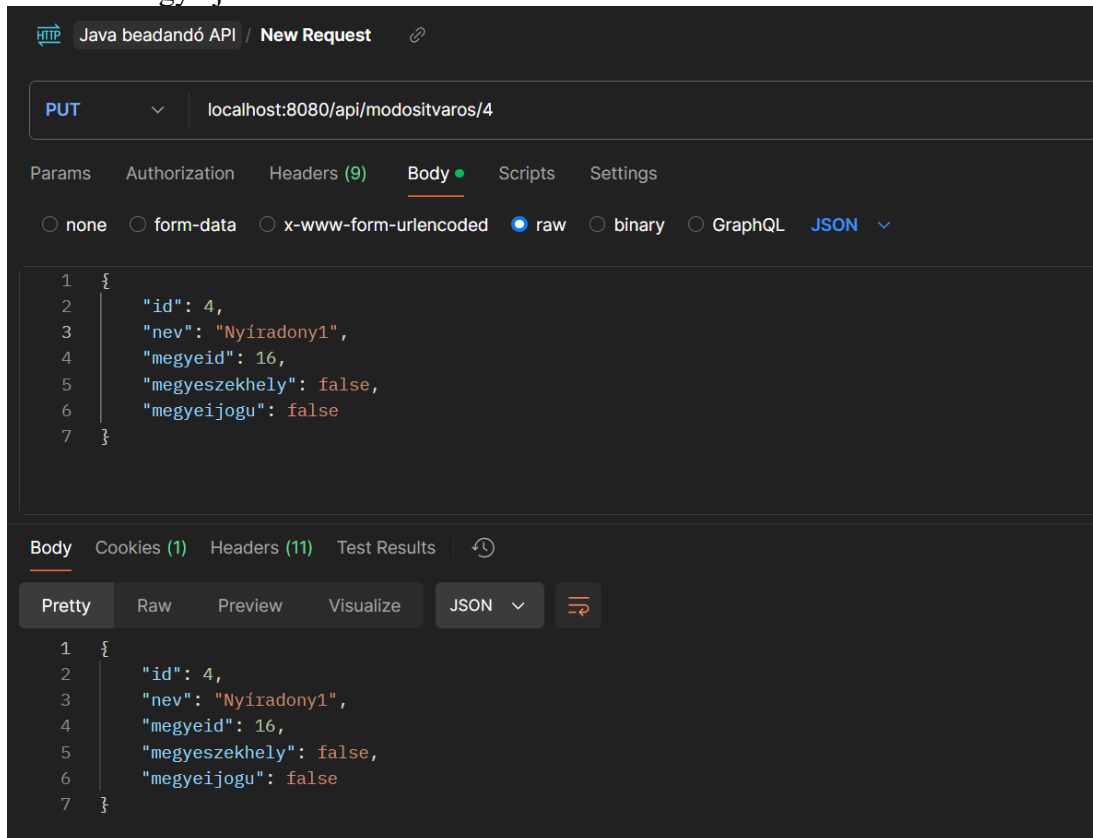
The screenshot shows the Postman interface with a GET request to `localhost:8080/api/varos/4`. The response is a JSON object for the city of Nyíradony, with `id`: 4, `nev`: "Nyíradony", `megyeid`: 16, `megyeszekhely`: false, and `megyeijogu`: false. The response status is 200 OK.

```
1 {
2   "id": 4,
3   "nev": "Nyíradony",
4   "megyeid": 16,
5   "megyeszekhely": false,
6   "megyeijogu": false
7 }
```

Below the Postman interface, a terminal window shows the following commands and output:

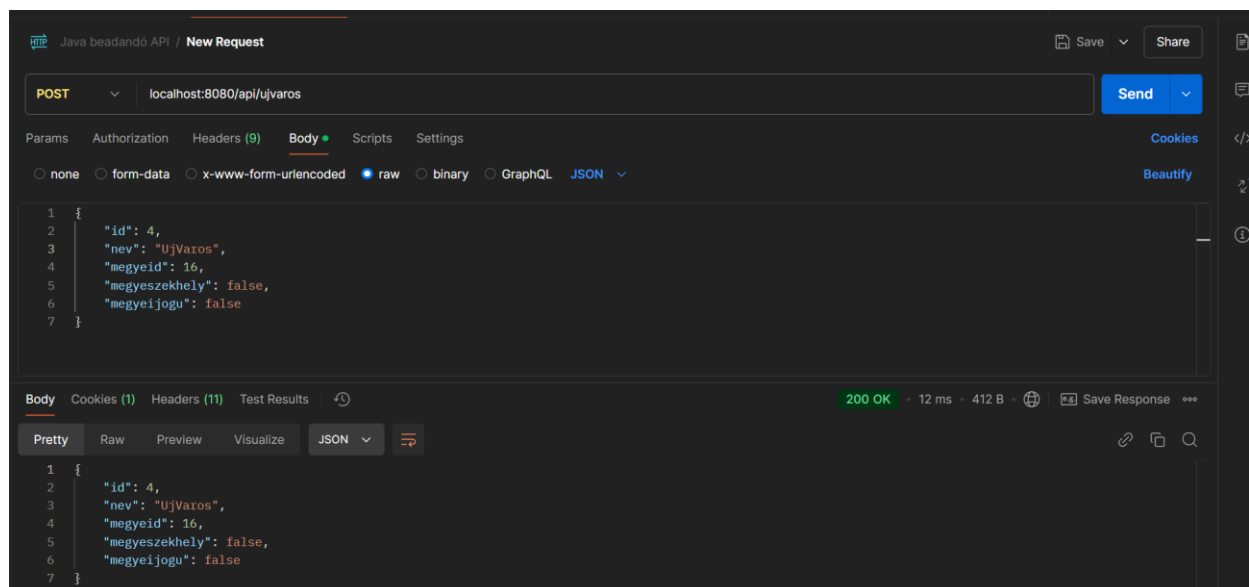
```
C:\Users\holyb>curl localhost:8080/api/varos/4
{"id":4,"nev":"Nyíradony","megyeid":16,"megyeszekhely":false,"megyeijogu":false}
C:\Users\holyb>
```


A PUT /api/modositvaros/{id} módosítja a megadott városid értékeit, itt json-ben kell megadni az új értékeket. Itt kiolvassuk az adatokat a kapott json-ből és ezeket elmentjük az id-hoz. Ha nincs ilyen ID akkor létrehoz egy új rekordot.



```
C:\Users\holyb>curl -X PUT localhost:8080/api/modositvaros/4 -H "Content-Type: application/json" -d '{"id": 4,"nev": "Nyíradony1","megyeid": 16,"megyeszekhely": false,"megyeijogu": false}'
```

A PUT-hoz hasonlóan a POST /api/ujvaros létrehoz egy új várost, itt nem kell megadni ID-t mert azt létrehozza majd a DB autoincrement miatt. Egy JSON fájlt kell megadni.



/api/varostorol/{id} DELETE method-al készült egy törlési API endpoint ahol ID alapján lehet várost törölni. Látható hogy 200-at ad vissza az API és törlődött a város.

