

Mérési jegyzőkönyv – Szoftver Laboratórium 5

4. mérés: SOA

Név:	Szabó Bence Farkas
Neptun kód:	RF57V5
Feladat kódja:	26 – SZALL
Mérésvezető neve:	Bőjti Bence
Mérés időpontja:	2017-04-06 16:15
Mérés helyszíne:	R4K
A működő alkalmazás elérhetősége:	
Megoldott feladatok:	1,2,3,4,5,6,
Elérhető pontszám (plusz pontok nélkül):	40p

Mérési feladatok megoldása

1. feladat: kötelező REST szolgáltatás készítése

A megoldáshoz használt SQL utasítás

Ha a feladat megoldásához szükség volt SQL utasításra, azt itt tüntessük fel, a Kód stílus segítségével!

1. lekérdezés:

```
SELECT order_id, description, quantity, deadline_date FROM orders
```

2. lekérdezés:

```
'SELECT description, vehicle_type, quantity, origin, destination,
order_date, deadline_date, comment_text
FROM orders
WHERE order_id = :order_id', order_id=order_id
```

Magyarázat

Az első feladatban két egyszerű Python kódot kellett írni, amelyeknek a „./orders.json” és a „./orders/<order_id>.json” címeknél kellett a megfelelő adatokat kilistázniuk, JSON formátumban. A megoldáshoz a „@app.route(./orders.json)” és az „@app.route(./orders/<order_id>.json)” kezdetű metódusok tartoznak.

Elsőként a megrendeléseket lekérdező metódust valósítottam meg. Az adatok eléréséhez a fentebb látható 1. lekérdezéssel jelzett SQL parancsot használtam.

Ennek az SQL lekérdezésnek a segítségével és a cur.execute() metódussal megkapjuk az eredménytáblát, amelynek a sorait egy JSON tömbbe másoljuk, melynek sorai szótárak. Ezt a tömböt adjuk vissza. A dátum tárolásához először sima, időpont nélküli dátummá konvertáljuk a date() metódussal, majd ezt az isoformat()-tal a kellő formátumra hozzuk.

Az adott order_id-jű megrendelés adatait lekérdező metódust a 2. lekérdezéssel valósítottam meg. Ha ennek eredménytáblája üres (None) akkor a felhasználónak abort(404) üzenetet adunk. Ha nem üres, a lekért adatokat az előző metódusnál látott módon JSON formátumra hozzuk. Itt nem kell tömböt visszaadni, mivel csak egyetlen megrendelés adatait adjuk vissza.

Tesztelés menete

/orders.json tesztelése:

```
$ curl localhost:20918/orders.json
```

```
{
  "orders": [
    {
      "deadline_date": "2017-05-10",
      "description": "bútor",
      "order_id": "2017/000001",
      "quantity": 10000
    },
    {
      "deadline_date": "2017-04-11",
      "description": "építőanyag",
      "order_id": "2017/000002",
      "quantity": 8000
    },
    {
      "deadline_date": "2017-03-21",
      "description": "építőanyag",
      "order_id": "2017/000003",
      "quantity": 20000
    },
    {
      "deadline_date": "2017-04-12",
      "description": "élelmiszer <tartós>",
      "order_id": "2017/000004",
      "quantity": 15000
    },
    {
      "deadline_date": "2017-02-22",
      "description": "bútoralapanyag",
      "order_id": "2017/000005",
      "quantity": 30000
    },
    {
      "deadline_date": "2017-05-01",
      "description": "élelmiszer <ital>",
      "order_id": "2017/000006",
      "quantity": 20000
    },
    {
      "deadline_date": "2017-02-10",
      "description": "műszaki cikk",
      "order_id": "2017/000011",
      "quantity": 20000
    },
    {
      "deadline_date": "2017-03-08",
      "description": "papíráru, <nyers>",
      "order_id": "2017/000012",
      "quantity": 34000
    },
    {
      "deadline_date": "2017-04-10",
```

```

    "description": "vegyszer (festék&hígító)",
    "order_id": "2017/000013",
    "quantity": 12000
  },
  {
    "deadline_date": "2017-03-16",
    "description": "üzemanyag",
    "order_id": "2017/000021",
    "quantity": 16000
  },
  {
    "deadline_date": "2017-03-12",
    "description": "üzemanyag",
    "order_id": "2017/000022",
    "quantity": 42000
  },
  {
    "deadline_date": "2017-04-12",
    "description": "baromfi",
    "order_id": "2017/000031",
    "quantity": 16000
  },
  {
    "deadline_date": "2017-02-28",
    "description": "versenyló",
    "order_id": "2017/000032",
    "quantity": 4000
  },
  {
    "deadline_date": "2017-04-13",
    "description": "sertés",
    "order_id": "2017/000033",
    "quantity": 20000
  },
  {
    "deadline_date": "2017-04-24",
    "description": "használt autó",
    "order_id": "2017/000041",
    "quantity": 10
  },
  {
    "deadline_date": "2017-03-12",
    "description": "hatóságilag lefoglalt járművek",
    "order_id": "2017/000042",
    "quantity": 15
  }
]
}

```

```
/orders/<order_id>.json tesztelése:
$ curl localhost:20918/orders/2017/000001.json
{
  "comment_text": null,
  "deadline_date": "2017-05-10",
  "description": "bútor",
  "destination": "Budapest (HU)",
  "destination_currency": "HUF",
  "order_date": "2017-01-02",
  "origin": "Berlin (DE)",
  "origin_currency": "EUR",
  "quantity": 10000,
  "vehicle_type": "D"
}
```

2. feladat: kötelező REST fogyasztó készítése

A megoldáshoz használt SQL utasítás

Magyarázat

A második feladatnál a Request távoli szolgáltatás-elérő funkciót kellett használni. Az adott ország valutáját az alábbi linken tudtam lekérdezni: http://rapid.eik.bme.hu:9080/currency_ws/currencies/<country>.json, ahol a „country” az adott ország hivatalos országcódja. Ezt az előző feladat második metódusából, a lekérdezés eredményeként kapott response változójából kapjuk meg, melyből kiolvassuk a forrás és cél nevét. Ez tartalmazza az ország azonosítóját is, melyet kivágunk a string-ből. Ezt adjuk hozzá a GET metódus által használt url-hez. Így már lekérdezhető az országok valutája, melyet az előző feladat második metódusának kiegészítéseként adunk hozzá a visszaadott eredményhez.

Tesztelés menete

```
$ curl http://rapid.eik.bme.hu:9080/currency_ws/currencies/hu.json
{
  "country": "en"
  "currency": "HUF"
}
```

Példaadatok

Ha a feladat teszteléséhez szükség van példaadatokra, azt itt tüntessük fel

3. feladat: kötelező AJAX megoldás készítése

Magyarázat

A harmadik feladatnál egy statikus .html weblapot kellett készíteni, amely tartalmazta a segédlet összes \$-al kapcsolatos funkcióját. A weblap betöltődésekor a loadData() metódus hívódik meg, mely egy \$.ajax aszinkron hívással a „../orders.json” címről lekérdezi a táblázatba az adatokat, majd az orders_into_table() metódussal lehet ezeket az adatokat a táblázatba felvenni. Ezek mind <script> és a </script> közé kerültek, amelyet a <body></body>-ba tettem. Ezen kívül még felvettem az oldalra a táblázatot, „orders” ID-vel, hogy hivatkozni lehessen rá, és felvettem a táblázat fejlécét. A táblázatot kicsit kiszépítettem, hogy vonalakkal legyenek a cellák elválasztva.

Tesztelés menete

<http://localhost:20918/static/orders.html>

Order ID	Description	Quantity	Deadline Date
2017/000001	bütor	10000	2017-05-10
2017/000002	építőanyag	8000	2017-04-11
2017/000003	építőanyag	20000	2017-03-21
2017/000004	élelmiszer <tartós>	15000	2017-04-12
2017/000005	bütoralapanyag	30000	2017-02-22
2017/000006	élelmiszer <ital>	20000	2017-05-01
2017/000011	műszaki cikk	20000	2017-02-10
2017/000012	papíráru, <nyers>	34000	2017-03-08
2017/000013	vegyszer (festék&hígító)	12000	2017-04-10
2017/000021	üzemanyag	16000	2017-03-16
2017/000022	üzemanyag	42000	2017-03-12
2017/000031	baromfi	16000	2017-04-12
2017/000032	versenylo	4000	2017-02-28
2017/000033	sértés	20000	2017-04-13
2017/000041	használt autó	10	2017-04-24
2017/000042	hatóságilag lefoglalt járművek	15	2017-03-12

4. feladat: választható REST szolgáltatás bővítés

A megoldáshoz használt SQL utasítás

1. lekérdezés

```
"SELECT order_id, description, quantity, deadline_date FROM orders
WHERE description LIKE :description ESCAPE '@',
description=description"
```

2. lekérdezés

```
SELECT order_id, description, quantity, deadline_date FROM orders WHERE
vehicle_type
LIKE :vehicle_type ESCAPE '@', vehicle_type=vehicle_type
```

Magyarázat

A negyedik feladathoz két Python kódot írtam meg, melyek a „@app.route(\"/orders/by_description/<desc_pfx>.json\")” és az „@app.route(\"/orders/by_type/<type>.json\")” kezdetű metódusokhoz tartoznak. A kettő igencsak hasonló volt, mivel mind a kettő egy-egy paraméteres SQL lekérdezést hajtottak végre, és ez a paraméter a keresendő szó volt. A megadott paramétert ahhoz, hogy a speciális karakterek kereshetőségét megteremtsem, a Python replace() string-metódusával változtattam meg, minden speciális karakter elé egy „@” jel került, majd az SQL lekérdezést annyival módosítottam, hogy a „@” jelet „escape-elje”, vagyis a mögötte lévő karakter karakterként értelmeződjön. Így a fent látható paraméteres SQL lekérdezéseket használtam. Ahhoz, hogy a csonkolt szavak is kereshetők legyen, a keresendő szavakat kiegészítettem: az elejére és végére is egy-egy „%” karaktert került, így már bármelyik olyan szót, amely tartalmazza a keresett kifejezést, visszaadja a kereső. Ha az SQL lekérdezés által adott eredménytábla üres (None), akkor 404-es kódú hibaüzenetet adunk a felhasználónak, amiből tudja, hogy a keresett szöveg nem szerepel a táblázat adott oszlopában. A választ az első feladathoz hasonló JSON formátumban adjuk vissza.

Tesztelés menete

Láthatjuk, hogy a 'b'-re keresve kilistázza az összes megrendelést, melynek leírásában szerepel b-betű:

```
$ curl http://localhost:20918/orders/by_description/b.json
{
  "orders": [
    {
      "deadline_date": "2017-05-10",
      "description": "bútor",
      "order_id": "2017/000001",
      "quantity": 10000
    },
    {
      "deadline_date": "2017-02-22",
      "description": "bútoralapanyag",
      "order_id": "2017/000005",
      "quantity": 30000
    },
    {
      "deadline_date": "2017-04-12",
      "description": "baromfi",
      "order_id": "2017/000031",
      "quantity": 16000
    }
  ]
}
```

Hasonlóan működik ez a jármű típusánál is:

```
& curl http://localhost:20918/orders/by_type/D.json
{
  "orders": [
    {
      "deadline_date": "2017-05-10",
      "description": "bútor",
      "order_id": "2017/000001",
      "quantity": 10000
    },
    {
      "deadline_date": "2017-04-11",
      "description": "építőanyag",
      "order_id": "2017/000002",
      "quantity": 8000
    },
    {
      "deadline_date": "2017-03-21",
      "description": "építőanyag",
      "order_id": "2017/000003",
      "quantity": 20000
    }
  ]
}
```

```

    },
    {
      "deadline_date": "2017-04-12",
      "description": "élelmiszer <tartós>",
      "order_id": "2017/000004",
      "quantity": 15000
    },
    {
      "deadline_date": "2017-02-22",
      "description": "bútoralapanyag",
      "order_id": "2017/000005",
      "quantity": 30000
    },
    {
      "deadline_date": "2017-05-01",
      "description": "élelmiszer <ital>",
      "order_id": "2017/000006",
      "quantity": 20000
    }
  ]
}

```

Illetve a % karakterre való kereséssel nem listáz ki mindent a kérés:

```

& curl http://localhost:20918/orders/by_type/%.json
{
  "orders": []
}

```

5. feladat: választható REST szolgáltatás bővítés

A megoldáshoz használt SQL utasítás

Magyarázat

Az ötödik feladatot a DELETE verb megírásával kezdtem. A service.py –ba vettem fel egy új metódust, amely akkor hívódik meg, hogyha a „/orders/<order_id>” címen szeretnének valamilyen műveletet végrehajtani, a megengedett műveletek: DELETE és PUT.

A egy vizsgálattal kezdem, mellyel megnézem hogy létezik-e az adott order_id-val rendelkező megrendelés az adatbázisban. Ha nem akkor 404-es hibával jelezzük a felhasználónak, ha pedig igen, akkor a megadott utasítástól függően hajtsa végre a DELETE vagy PUT szálát. A DELETE szál SQL utasítása:

```
"DELETE FROM orders WHERE order_id = :order_id", order_id=order_id
```

Ha nem sikerült a törlés, akkor 500-as kódú hibát dob a program.

A PUT művelethez az alábbi SQL kódot használtam fel:

```
"UPDATE orders
SET description = :description,
vehicle_type = :vehicle_type,
quantity = :quantity,
origin = :origin,
destination = :destination,
order_date = TO_DATE(:order_date, 'yyyy-MM-dd'),
deadline_date = TO_DATE(:deadline_date, 'yyyy-MM-dd'),
comment_text = :comment_text
WHERE order_id = :order_id",
description=data['description'],
vehicle_type=data['vehicle_type'],
quantity=data['quantity'],
origin=data['origin'],
destination=data['destination'],
order_date=data['order_date'],
deadline_date=data['deadline_date'],
comment_text=data['comment_text'],
order_id=order_id)
```

Ha nem sikerült a beillesztés, 500-as kódú hibát dob a program.

A POST művelethez egy külön metódust írtam meg, amely az „/orders.json” hivatkozásakor elérhető. A metódus részletes magyarázata megtalálható a service.py file kommentjei között, itt csak a lényegesebb dolgokra térek ki. Ahhoz, hogy az újonnan felvett eszközhöz azonosítót tudjon rendelni a függvény, generálni kell egy ilyen azonosítót. Ez a generate_order_id() metódus feladata, amely végigiterál az azonosítókön, és a legnagyobbhoz hozzáadva egyet, megteremti az új, egyedi azonosítót. Miután ez megvan, már csak végre kell hajtani az alábbi SQL utasítást:


```

"INSERT INTO orders
VALUES
(:order_id,
:description,
:vehicle_type,
:quantity, :origin,
:destination,
TO_DATE(:order_date, 'yyyy-MM-dd')),
TO_DATE(:deadline_date, 'yyyy-MM-dd')),
:comment_text",
order_id=order_id,
description=data['description'],
vehicle_type=data['vehicle_type'],
quantity=data['quantity'],
origin=data['origin'],
destination=data['destination'],
order_date=data['order_date'],
deadline_date=data['deadline_date'],
comment_text=data['comment_text'])

```

Tesztelés menete

```

$ curl -request DELETE -header „Content-Type:
application/json”localhost:20918/orders.json

```

Példaadatok

Ha a feladat teszteléséhez szükség van példaadatokra, azt itt tüntessük fel

6. feladat: választható AJAX megoldás bővítés

Magyarázat

A hatodik feladathoz az orders.html oldalt kellett kiegészíteni gombokkal, valamint az adatok frissítésével. Először a gombokat vettem fel a táblába, az ajaxdemo.html file mintájára, kiegészítve az ID megadásával a táblafeltöltő-ciklusban. A gombok nevének a sorhoz tartozó megrendelés order_id mezőjét adtam meg, hogy arra később könnyedén hivatkozhaszak. A gomb clicked() metódusát a úgy írtam meg, hogy egy \$.ajax() lekérdezéssel a korábban már megírt „orders/order_id.json” paraméteres oldalról kérje le az adatokat, és ha ez sikeres volt, akkor azokat egy alert ablakban jelenítse meg. A \$.ajax()-ban úgy adtam meg a megrendelés azonosítóját, hogy lekérdeztem a megnyomott button nevét.

Ez után következett az adatok frissítése. Ehhez a window.setInterval() metódust használtam, ami a paraméterként megadott függvényt a (másik) paraméterként megadott időközönként meghívja. Ez a metódus ismét a loadData() volt, mivel ez tölti be az adatokat. A success() függvényét még ki kellett egészíteni, hogy a már meglévő sorokat törölje ki, azokra már nem lesz szükség. Ezt egy for ciklussal oldottam meg. Ahhoz hogy az eredeti, még frissítés előtti, sorszámot eltároljam, felvettem egy globális tableData változót.

A változás jelzéséhez összehasonlítottam a tábla frissítés előtti és frissítés utáni változatát a compare() függvényvel. Ez a függvény ellenőrzi, hogy a különbözik e a táblának bármely cellája. Ha valahol eltérést találunk, akkor az eltérésnek megfelelően alert()-et küldünk a felhasználónak a változásokról.

Tesztelés menete

Az weblapon lévő gombok működése az első képen, míg az oldal frissítését végző metódusok üzenetei a további képeken láthatók.

<http://localhost:20918/static/orders.html>

Order ID	Description	Quantity	Deadline Date	
2017/000001	bütor	10000	2017-05-10	>
2017/000002	építőanyag	8000	2017-04-11	>
2017/000003	építőanyag	20000	2017-03-21	>
2017/000004	élelmiszer <tartós>	15000	2017-04-12	>
2017/000005	bütoralapanyag		2017-02-22	>
2017/000006	élelmiszer <ital>		2017-05-01	>
2017/000011	műszaki cikk		2017-02-10	>
2017/000012	papíraru, <nyers>		2017-03-08	>
2017/000013	vegyi áru (festék&hígító)		2017-04-10	>
2017/000021	üzemanyag		2017-03-16	>
2017/000022	üzemanyag		2017-03-12	>
2017/000031	baromfi		2017-04-12	>
2017/000032	versenylo		2017-02-28	>
2017/000033	sértés		2017-04-13	>
2017/000041	használt autó		2017-04-24	>
2017/000042	hatóságilag lefoglalt járművek		2017-03-12	>

A(z) localhost:20918 közlendője:
Description: bütor
Vehicle Type: D
Quantity: 10000
Origin: Berlin (DE)
Destination: Budapest (HU)
Order Date: 2017-01-02
Deadline Date: 2017-05-10
Comments: null
Origin country currency: EUR
Destination country currency: HUF
OK

Az oldal egy gomb megnyomása után.

A teszteléshez az SQL developer-ben módosítottam az adatbázis tartalmát, melynek hatására az alábbi eredmények láthatók.

Order ID	Description	Quantity	Deadline Date	
2017/000001	bütor	10000	2017-05-10	>
2017/000002	építőanyag	8000	2017-04-11	>
2017/000003	építőanyag	20000	2017-03-21	>
2017/000004	élelmiszer <tartós>	15000	2017-04-12	>
2017/000005	bütoralapanyag		2017-02-22	>
2017/000006	élelmiszer <ital>		2017-05-01	>
2017/000011	műszaki cikk		2017-02-10	>
2017/000012	papíraru, <nyers>		2017-03-08	>
2017/000013	vegyi áru (festék&hígító)		2017-04-10	>
2017/000021	üzemanyag		2017-03-16	>
2017/000022	üzemanyag		2017-03-12	>
2017/000031	baromfi	16000	2017-04-12	>
2017/000032	versenylo	4000	2017-02-28	>
2017/000033	sértés	20000	2017-04-13	>
2017/000041	használt autó	10	2017-04-24	>
2017/000042	hatóságilag lefoglalt járművek	15	2017-03-12	>

A(z) localhost:20918 közlendője:
Új megrendelés az adatbázisban: 2017/000090
☐ Ez az oldal ne nyisson meg további párbeszédablakokat.
OK

Order ID	Description	Quantity	Deadline Date	
2017/000090	valami	100	2017-05-20	>
2017/000001	bütor	10000	2017-05-10	>
2017/000002	építőanyag	8000	2017-04-11	>

Az első képen az oldal üzenete látható insert után, a második képen pedig az új megrendelés a táblázatban.

A(z) localhost:20918 közlendője:
Ezt a megrendelést törölték: 2017/000090
☐ Ez az oldal ne nyisson meg további párbeszédablakokat.
OK

Order ID	Description	Quantity	Deadline Date
2017/000001	bútor	10000	2017-05-10
2017/000002	építőanyag	8000	2017-04-11
2017/000003	építőanyag	20000	2017-03-21
2017/000004	élelmiszer <tartós>	15000	2017-04-12
2017/000005	bútoralapanyag	30000	2017-02-22
2017/000006	élelmiszer <ital>	20000	2017-05-01
2017/000011	műszaki cikk	20000	2017-02-10
2017/000012	papíraru, <nyers>	34000	2017-03-08
2017/000013	vegyi áru (festék&higító)	12000	2017-04-10
2017/000021	üzemanyag	16000	2017-03-16
2017/000022	üzemanyag	42000	2017-03-12
2017/000031	baromfi	16000	2017-04-12
2017/000032	versenylo	4000	2017-02-28
2017/000033	sértés	20000	2017-04-13
2017/000041	használt autó	10	2017-04-24
2017/000042	hatóságilag lefoglalt járművek	15	2017-03-12

Az első képen az oldal üzenete delete után, a másodikon a már megváltozott táblázat látható.

A(z) localhost:20918 közlendője:

2017/000001 Megnevezese bútor-rol bútor 2-re változott

☐ Ez az oldal ne nyisson meg további párbeszédablakokat.

OK

Order ID	Description
2017/000001	<u>bútor 2</u>
2017/000002	építőanyag
2017/000003	építőanyag

Az első képen az oldal üzenete update után, a másodikon a megváltoztatott adatok látszódnak.

Vélemény(ek) a mérésről

Vélemény, építő jellegű kritika.