

Szia!

Ebben a dokumentumban a zeroToHero Programozás verseny első fordulójának a feladatsorát olvashatod.

Ne feledd, a feladatok megoldására egy hét áll a rendelkezésedre.

Beküldési határidő: 2020.07.26. 23:59

A feladatokat a következő lépések segítségével adhatod be:

Beadás módja

1. Töltsd le ezt a Google Docs dokumentumot a gépedre
2. Töltsd ki a **Sárgán aláhúzott helyeket!**
3. Miután kitöltötted a dokumentumot, mentsd le .pdf formátumban
4. Tedd elérhetővé az alkalmazásod bárki számára heroku segítségével. Segítséget a heroku ismertetőben kaphatsz hozzá!
5. Küldd el az így elkészült **feladatlapot** és a megoldott **zip fájlba csomagolt fájlokat**, mint csatolmányt emailben, a következő paraméterekkel:
 - a. Tárgy: **[1.forduló] <Teljes neved>** (pl. "[1.forduló] Kovács András")
 - b. Cím: programozas@pcf.hu
6. Ellenőrizd, hogy tényleg megfelelően elküldted-e a levelet és csatoltad-e a megoldott feladatok forráskódját is :)

Sok sikert kívánunk a versenyhez!

Az első kitöltendő kérdésként kérlek add meg az neved és az email címed, melyről a megoldásokat küldöd:

Név:

E-mail cím amiről a megoldást küldöd:

Üdvözöllek a zeroToHero programozás versenyének első fordulójában!

Add meg az általad választott nyelvet amivel a gyakorlati feladatokat fogod megoldani:

A beküldött projektnek fordíthatónak és futtathatónak kell lennie!

Technikai segítség

Python ismertető:

<http://www.zerotohero.hu/python>

Javascript ismertető:

<http://www.zerotohero.hu/javascript>

Java ismertető:

<http://www.zerotohero.hu/java>

A feladatmegoldáshoz szintén szükséges heroku ismertető:

<http://www.zerotohero.hu/heroku>

Előkészített keretrendszerek:

http://www.zerotohero.hu/elokeszített_keretrendszerek

Megoldásod heroku linkje:

Elméleti kérdések

Feladat 1

Mik azok a HTTP státusz kódok, mire használjuk őket: (1 pont)

Sorolj fel 3 HTTP státusz kódot és a hozzá tartozó állapotot szövegesen: (1 pont)

Minek a rövidítése a REST? Mit jelent az a gyakorlatban hogy egy backend szerver stateless? (2 pont)

Egy adatbázis táblában mit nevezünk egy tábla elsődleges kulcsának? Mit nevezünk külső kulcsnak? (2 pont)

Összesen: 6 pont

Alkalmazás készítése

Feladat 2 - HTTP státusz meghatározás

A feladatod létrehozni egy REST GET endpoint-ot, ami query paraméterben kap egy számértéket(HTTP státuszkód), az endpoint pedig visszaadja a hozzá tartozó HTTP státusz szöveges leírását. Figyeljünk a hibakezelésre is! Amennyiben az endpoint nem létező HTTP státuszkódot kap paraméterül, térjünk vissza 400-as hibakóddal!

A feladatot egy már előkészített példa programon belül kell megoldanod. Nem használhatsz új függőségeket sem. A query paraméter neve, amivel dolgoznod kell: `statusCode`. A rest endpoint a `/http-status/getStatusDescription` URL-en kell legyen elérhető.

Példa lekérdezés-válasz párok:

Query: 404, Response: NOT FOUND

Query: 300, Response: AMBIGUOUS

Összesen: 18 pont

Feladat 3 - CRUD

A Finom Falatozó Ételbár évek óta szolgálja ki a környezetükben élő embereket. Viszont most bővülni szeretnének, új helyszíneken terveznek megnyitni, ehhez viszont új berendezés és munkaerő is kell. A te feladatod egy olyan REST API készítése, mely meghívásával ezek a változások számon tarthatóak.

Séma:

1. Location:
 - a. Id (mentéskor generált)
 - b. Name (szöveg)
 - c. Address (szöveg)
2. Equipment:
 - a. Id (mentéskor generált)
 - b. Name (szöveg, lehet null)
 - c. Type (cash register, oven)
 - d. LocatedAt (Location)
3. Employee:
 - a. Id (mentéskor generált)
 - b. Name (szöveg)
 - c. Job (manager, cashier, cook)
 - d. WorksAt (Location)
 - e. Operates (Equipment)

A feladatod elkészíteni a sémát adatbázisban (12 pont) és implementálni a következő műveleteket mindegyik táblára:

- Create (8pont)
- Update (8pont)
- Delete (8pont)

A műveletek közben a következő dolgokra kell figyelni:

- Egnél több manager nem dolgozhat egy helyszínen
- Pénztárosból nem dolgozhat több egy helyszínen mint amennyi pénztárgép van. Szakácsból nem dolgozhat több egy helyszínen mint amennyi sütő van
- Nem használhat alkalmazott olyan gépet, ami nem azonos helyszínen van vele
- Minden műveletre igaz, hogy nem maradhat befejeztével üres függelék az adatbázisban (pl felszerelés törlés után már nem létező pénztárgéppel dolgozik egy pénztáros)

Az endpointokat a következő URL-eken ajánld ki:

- Create (POST):
 - /api/location/
 - /api/equipment/
 - /api/employee/
- Update (PUT) & Delete (DELETE):
 - /api/location/{id}/
 - /api/equipment/{id}/
 - /api/employee/{id}/

Az endpoint-ok csak JSON (application/json) content type-ot fogadjanak el. Abban az esetben ha más content type érkezik 'Bad Request' HTTP hibakódot kell visszaadni.

A request body-ban Create és Update műveleteknél minden mezőt küldeni kell, melyek neve a sémában olvasható nevek csupa kisbetűvel.

Összesen: 36 pont

Feladat 4 - Négyzetek keresése

Adott a síkon pontok halmaza. Egy pontot egy X és egy Y koordinátával határozunk meg. Feladatod megadni, hogy hányféleképpen tudunk egy darab szabályos négyzetet rajzolni úgy, hogy a négyzet mind a 4 csúcsa a megadott ponthalmaz elemeiből kerüljenek ki.

A feladatot egy már előkészített keretrendszerben kell megoldanod. A request body neve amivel dolgoznod kell listOfPoints. Nem használhatsz új függőségeket sem. A rest endpoint a "/squares/getNumberOfSquares" url en kell legyen, és a berajzolható négyzetek darabszámát kell visszaadnia.

Példa input-output párok:

[{"x": 2, "y": 2}, {"x": 2, "y": 3}, {"x": 3, "y": 3}, {"x": 3, "y": 2}] -> 1 darab

[{"x": 2, "y": 2}, {"x": 2, "y": 3}, {"x": 3, "y": 3}, {"x": 3, "y": 20}] -> 0 darab

[{"x": 2, "y": 2}, {"x": 2, "y": 3}, {"x": 3, "y": 3}, {"x": 3, "y": 2}, {"x": 4, "y": 2}, {"x": 4, "y": 3}, {"x": 4, "y": 4}, {"x": 2, "y": 4}, {"x": 3, "y": 4}] -> 6 darab

Összesen: 40 pont

Pontozás:

Feladat 1:	
Feladat 2:	
Feladat 3:	
Feladat 4:	
Összesen:	