Teszt Megoldókulcs Feltörése Lineáris Programozás Solver Segítségével

A program egy JSON REST API-t biztosít mikroszervíz stílusú használatra.

Két részből áll, egy solver komponens ami önmagában is használható NodeJS-ből, valamint az API-t biztosító modul, ami a solveres részt kiegészíti.

# Lekérdezések:

Az API a szerver IP címén fut, konfigurálható porton, elsősorban belső használatra, de a user token authentikációja kivételével a bemeneti adatok érvényességét ellenőrzi.

### JSON formátum

A szerver ezt használja a lekérések formázására, ez hierarchikusan kombinált adatokat tartalmaz, az adatok közötti szóközök és sortörések nem számítanak, de más szempontokból szigorú.

**Szám típus**

Lehet tizedes tört vagy egész szám, a szerver dönti el, hogy miként kezeli. Decimális formában kell megadni, a tizedesjegyeket pont választja el az egészrésztől.

**Szöveg típus**

Idézőjelek közé kell helyezni, kizárólag az ASCII 0x22 idézőjel használható. Speciális karaktereket például sortöréseket, tabulátorokat, belső idézőjeleket és “\” karaktereket meg kell jelölni egy “\” karakterrel, amit a speciális karakter elé helyezünk.

**Boolean adattípus**

Igaz/hamis bináris adatot tárol el, ezeket a true és false kulcsszavak jelölik.

**Objektum típus**

Kulcs-érték párokat tartalmaz, a kulcsok kötelezően szöveges típusúak, az érték bármilyen típusú lehet, egy másik objektum vagy tömb is.

Például:  
{  
 “kulcs1”: “érték1”,  
 “kulcs2”: érték2  
}

**Tömb típus**

Más típusok felsorolását tartalmazza, ezek bármilyen típusúak lehetnek és nem kötelező, hogy a tömb elemei azonos típusúak legyenek.

Például:  
[ “érték1”, “érték2”, “érték3” ]

A program ezt használja a saját formátumának leírására, a legtöbb programozási nyelv tartalmaz JSON segítő könyvtárat.

### URL enkódolási formátum / százalékos enkódolás

Magában az URL-ben így tárolhatunk kulcs-érték párokat, jellemzően mindkettő szöveges típusú, a speciális karaktereket (vagy speciális karakterek bájtjait) le kell cserélnünk a %00 stílusú, százalékjellel ellátott hexadecimális számokra.

Az így enkódolt adatot query paraméterekbe tesszük, ?kulcs1=ertek1&kulcs2=ertek2 elrendezésben.

### GET és POST

A HTTP-ben használt REST API két leggyakoribb lekérdezés fajtája.

**GET**  
Csak az URL-ben tud adatot adni és akkor használjuk ha adatot kapni szeretnénk.

**POST**  
A lekérés testében küldhetünk adatot, ez lehet bináris de gyakran szöveges formátumú és ezt a szöveget értelmezi a program mint például JSON adatstruktúra. Akkor használjuk ha adatot adni szeretnénk vagy komplex lekérdezést küldünk.

## 

## Saját adatstruktúrák

### POST “/Attempts”

Különböző emberek teszt próbálkozásai eltárolására/beküldésére szolgál, egy beküldés egy felhasználóhoz tartozik és egy próbálkozás adatait tartalmazza egy lekérdezés.

### A POST adat egy JSON objektumból áll, ebben több fontos kulcs-érték pár van.

**“user”**  
Szöveges típusú, ez azonosítja a felhasználót, lehet egyszerű ID, token, esetleg név.

**“course”**  
Szöveges típusú, a kurzus nevét tartalmazza.

**“path”**A teszt elérési útja, ez jellemzően az óra vagy téma neve vagy sorszáma, valamint a teszt neve.

**“score”**A teszt kitöltése során elért pontszám, vagyis a beküldött válasz kombinációk hány pontot érnek.

**“maxScore”**A teszt kitöltése során elérhető maximális pontszám, a számításokhoz nem szükséges.

**“questions”**A kérdés objektumokat tartalmazó tömb, ezek az objektumok tartalmazzák a kérdést, lehetséges válaszokat és a kitöltő választásait.

A kérdés/question objektumok különböző típusúak lehetnek, de vannak megegyező részeik is.

### 

### Alap kérdés objektum

**“type”**A kérdés típusa, ez határozza meg, hogy hogyan értelmezzük a többi adatot. Szöveges típusú és lehet “truefalse”, “singlechoice” és “multichoice”.

**“questionText”**Szöveges típusú, a kérdés szövegét tartalmazza, speciális esetekben ez hiányozhat vagy megegyezhet több kérdésben is, a kérdés azonosításához a válaszokat is felhasználjuk.

**“answers”**Válaszokat tárol tömb formátumban, a kérdés fajtájától függenek az általa tárolt objektum megkötései, a tömb legalább egy és maximum 20 elemből áll.

### Válaszok alap felépítése

**“text”**Szöveges típusú, a választható válasz szövegét tartalmazza.

**“checked”**Boolean-szerű típusú, tehát true/false de 0/1 is elfogadott, valamint ha hiányzik akkor false. Azt határozza meg, hogy a teszt kitöltője kiválasztotta-e ezt, mint helyes válasz.

### Igaz-hamis kérdés

**{ “type”: “truefalse”, … }** határozza meg.

**“answers”**Szigorúan 2 opciót tartalmazhat, ezekből csak egy lehet kiválasztva vagy egy sem, tehát legalább az egyik válasz “checked” értéke false.

#### Igaz-hamis válaszok

**“text”**Akármi lehet, de ha ez “Igaz”, akkor az megegyezik “True” értékkel, ha “Hamis” az megegyezik a “False” string típusú értékkel.

### 

### 

### Egyválasztásos kérdés

**{ “type”: “singlechoice”, … }** határozza meg.

**“answers”**Maximum egy választást tartalmazhat, tehát nulla vagy egy a **{ “checked”: true, … }** stílusú válaszok száma.

### Többválasztásos kérdés kérdés

**{ “type”: “multichoice”, … }** határozza meg.

Az alap kérdés formátum kritériumokon kívül más megkötést nem tartalmaz, nulla vagy több olyan választ is tartalmazhat, ami kiválasztott.

A POST “/Attempts” lekérdezés az ilyen formában kapott adatokat a beküldés idejével kiegészítve egy adatbázisban eltárolja.

### GET “/Info”

Paraméter nélküli lekérdezés, visszaadja az adatbázisban eltárolt próbálkozások kurzusait úgy, hogy mindegyik kurzus csak egyszer fordul elő a listában. A visszaadott érték egy -szövegeket tartalmazó- tömb.

### GET “/Info?course=...”

“**course**” paraméter tartalmazza a kiválasztott kurzust, amin belül keresünk.

Visszaadja a kurzushoz tartozó próbálkozások elérési útjait úgy, hogy mindegyik elérési út csak egyszer fordul elő a listában. A kurzus és elérési út kombináció egyértelműen meghatároz egy adott tesztet (nem egy próbálkozást). A visszaadott érték egy -szövegeket tartalmazó- tömb.

### 

### GET “/Info?course=...&path=...”

“**course**” paraméter tartalmazza a kiválasztott kurzust, amin belül keresünk.

“**path**” paraméter tartalmazza a kiválasztott kurzuson belüli elérési utat, vagyis a teszt azonosítóját.

Visszaadja kiválasztott teszthez tartozó statisztikákat egy objektum formájában.

**“attemptsCount”**A teszthez tartozó összes beküldött próbálkozások száma. Szám típusú.

**“uniqueAttempts”**Szám típusú, a teszthez tartozó egyedi próbálkozások száma, egy próbálkozás egyedi, ha a válaszok nem egyeznek meg (a kérdések és válaszok sorrendje nem számít).

**“filterTimes”**A próbálkozások beküldési idő szerinti szűrésére fennálló lehetőségek, a precízió állítható, jelenleg egy hét ez a beállítás, tehát azokat a heteket adja vissza amikor volt lehalább egy próbálkozás. Szám típusú, ms precíziójú Unix timestamp formátumban.

### GET “/Info?course=...&path=...&after=...”

“**course**” paraméter tartalmazza a kiválasztott kurzust, amin belül keresünk.

“**path**” paraméter tartalmazza a kiválasztott kurzuson belüli elérési utat, vagyis a teszt azonosítóját.

“**after**” paraméter tartalmazza azt a hetet, amitől kezdve érdekelnek minket a próbálkozások.

Visszaadja a kurzus, elérési út és kezdeti hét szerint leszűrt próbálkozások statisztikáit, objektum formában.

**“attemptsCount”**A leszűrt próbálkozásokhoz száma. Szám típusú.

**“uniqueAttempts”**Szám típusú, a leszűrt próbálkozások közötti egyedi próbálkozások mennyisége.

**“questions”**A leszűrt próbálkozásokban megtalálható egyedi kérdések és hozzájuk tartozó lehetséges válaszok gyűjteménye. Objektumok tömbje.

### 

### Egyedi kérdések struktúrája

**“id”**A kérdés azonosítója, a program generálja a kérdés adatai alapján. Szöveg típusú.

**“type”**Szöveg típusú, a kérdés típusát tartalmazza.

**“questionText”**A kérdés szövege.

**“answers”**Objektumok tömbje, az objektumok egy **“text”** attribútumot tartalmaznak ami a válasz szövege.

### GET “/Vectorized?course=...&path=...&after=...”

“**course**” paraméter tartalmazza a kiválasztott kurzust, amin belül keresünk.

“**path**” paraméter tartalmazza a kiválasztott kurzuson belüli elérési utat, vagyis a teszt azonosítóját.

“**after**” paraméter tartalmazza azt a hetet, amitől kezdve érdekelnek minket a próbálkozások.

Amennyiben van megfelelő mennyiségű leszűrt próbálkozás (konfigurálható, jelenleg 5), egy, solveres módon feldolgozható adatstruktúrát ad vissza amit a beküldött adatokból állít össze, de ezeket egymással kombinálja obfuszkációs szempontból.

Visszatérési érték formátuma:

"**questions**"

A kérdés objektumok egy tömbben, a lehetséges válaszokkal kiegészítve, a lehetséges válaszok közül csak azokat tartalmazza, amelyeket legalább egy próbálkozás tartalmaz (kivétel az igaz-hamis, ott mindkettő opciót tartalmazza).  
  
A kérdés objektumokon belül egy "answers" paraméter tartalmazza a szöveg formátumú válaszlehetőségeket.

"**vectorizedAnswers**"

A vektorizált válaszokat tartalmazza, tömbök tömbje, egyes obfuszkált próbálkozásokat tartalmaz, legbelül lebegőpontos számokkal, ami a feladat pontjának súlyozására használható, hasonlóan mintha 0 vagy 1 lenne.

"**scores**"

Obfuszkált pontszámok, lebegőpontos számok tömbje.

### GET “/Solver?course=...&path=...&after=...”

“**course**” paraméter tartalmazza a kiválasztott kurzust, amin belül keresünk.

“**path**” paraméter tartalmazza a kiválasztott kurzuson belüli elérési utat, vagyis a teszt azonosítóját.

“**after**” paraméter tartalmazza azt a hetet, amitől kezdve érdekelnek minket a próbálkozások.

Egy lineáris programozás solveres megoldással kiszámol egy lehetséges megoldókulcsot, amennyiben elég teszteset áll rendelkezésre, akkor ez a megoldókulcs mindenképp a helyes megoldás lesz.

"**truefalseQuestions**" és "**optionsQuestions**" az előző API hívással megegyezően kérdés objektumokat ad vissza lehetséges válaszokkal, de megkülönbözteti az igaz-hamis és a több lehetőséges válaszokat.

"**solverResult**”

A solveres megoldás értékei, egy objektum aminek paraméterei:

"**deltas**"

Amennyiben a solver nem tud tökéletes megoldást találni, a bemeneti adat helytelen. Ezek az értékek tartalmazzák, hogy melyik próbálkozás pont értékének mennyiben kell eltérnie ahhoz, hogy a kiszámolt megoldókulcs illeszkedjen. Számok tömbje, előjeles számok, az összegzéshez abszolútértéket kell venni.

"**binaryAnswers**"

A "truefalseQuestions" kérdésekhez és válaszokhoz tartozik, egy tömb ami pontszámot rendel az igaz és a hamis válaszokhoz, külön külön.

"**optionsAnswers**"

Az "optionsQuestions" kérdésekhez és válaszokhoz tartozik, pontszámot rendel az összes, mintaadatban megjelent válaszhoz.

# Minta hívások és válaszok

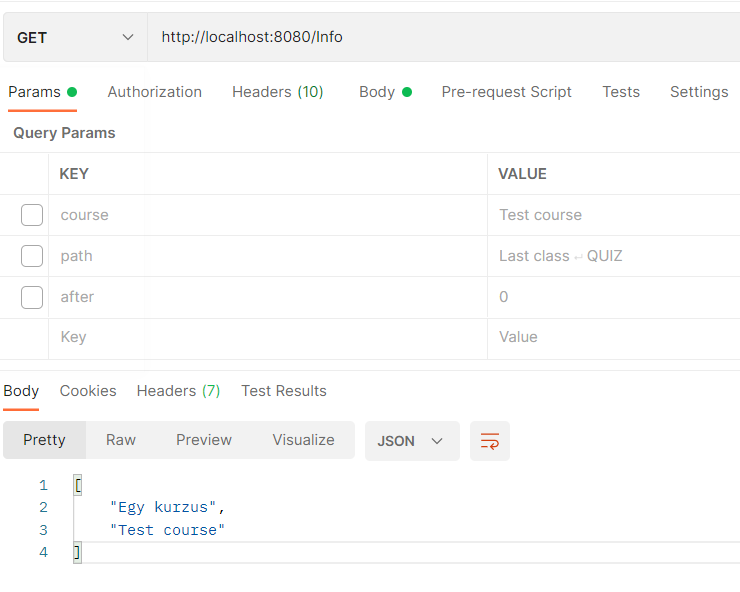
GET <http://localhost:8080/Info>

[

"Egy kurzus",

"Test course"

]

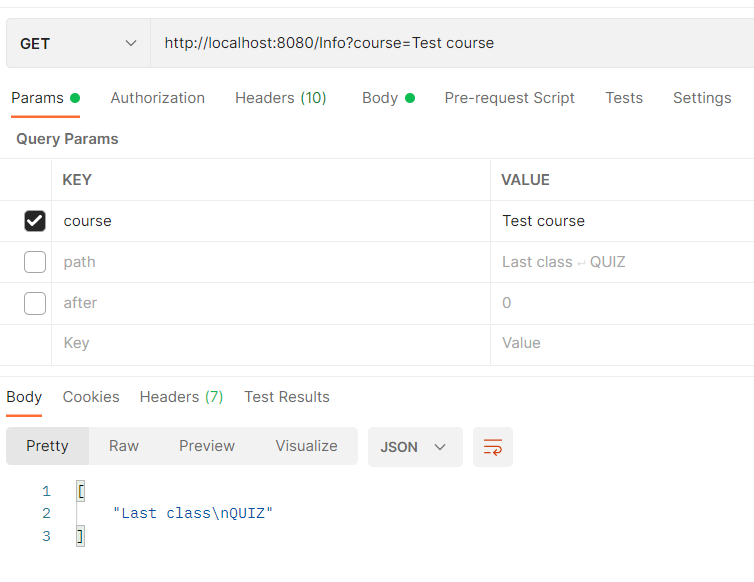


GET <http://localhost:8080/Info?course=Test%20course>

[

"Last class\nQUIZ"

]



GET [http://localhost:8080/Info?course=Test%20course&path=Last%20class%0AQUIZ](http://localhost:8080/Info?course=Test%20course&path=Last%20class%20QUIZ)

{

"attemptsCount": 10,

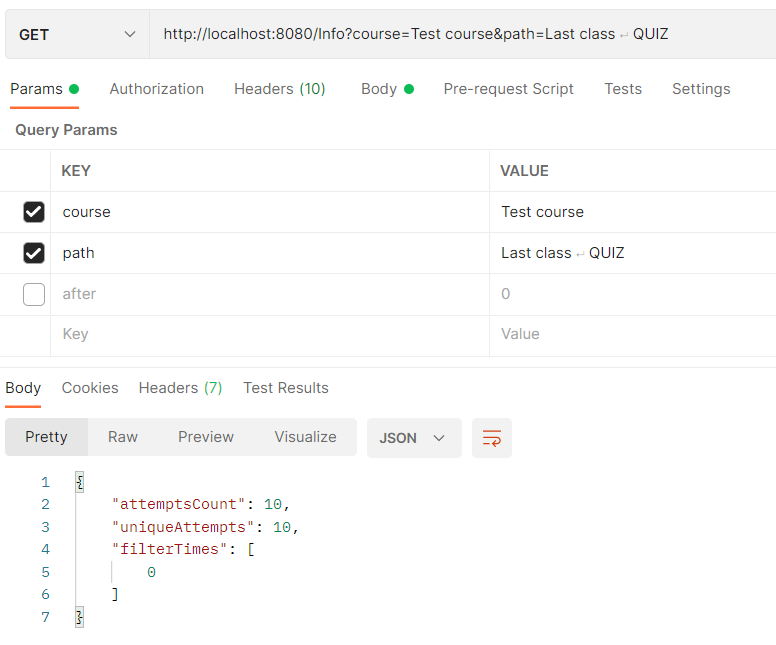
"uniqueAttempts": 10,

"filterTimes": [

0

]

}



GET [http://localhost:8080/Info?course=Test%20course&path=Last%20class%0AQUIZ&after=0](http://localhost:8080/Info?course=Test%20course&path=Last%20class%20QUIZ&after=0)

{

"attemptsCount": 10,

"uniqueAttempts": 10,

"questions": [

{

"id": "CylWp11J4T0QQTBi",

"type": "singlechoice",

"questionText": "Where in a supermarket can you expect to find Havarti?",

"answers": [

{

"text": "Dairy products section"

},

{

"text": "Drinks department"

},

{

"text": "Frozen food section"

},

{

"text": "Fruit and vegetable stalls"

}

]

},

{

"id": "vHOE4EfedYI7OIsK",

"type": "singlechoice",

"questionText": "In which 'Tomb Raider' title does cult leader Marco Bartoli stab himself with the Dagger of Xian and turns into a dragon?",

"answers": [

{

"text": "Tomb Raider I"

},

{

"text": "Tomb Raider II"

},

{

"text": "Tomb Raider III"

},

{

"text": "Tomb Raider: The Last Revelation"

}

]

},

{

"id": "Pvl5aGwAKhhE0-ht",

"type": "singlechoice",

"questionText": "Who wrote the 1936 self-help book \"How to Win Friends and Influence People?\"",

"answers": [

{

"text": "Dale Carnegie"

},

{

"text": "Noam Chomsky"

},

{

"text": "Richard Dawkins"

},

{

"text": "Yann Martel"

}

]

},

{

"id": "xy0FKdBI5\_mF19gz",

"type": "truefalse",

"questionText": "A 12-year old person is considered a teenager.",

"answers": [

{

"text": "False"

},

{

"text": "True"

}

]

},

{

"id": "tmSyhUZGYI1D45T4",

"type": "truefalse",

"questionText": "\"Azul\" is the Spanish word for red.",

"answers": [

{

"text": "False"

},

{

"text": "True"

}

]

},

{

"id": "ApIF29TMugF6lW5P",

"type": "truefalse",

"questionText": "The Southeast Asian speciality of balut is fertilised egg embyros.",

"answers": [

{

"text": "False"

},

{

"text": "True"

}

]

},

{

"id": "3M-M6jR1QpdgkHYY",

"type": "truefalse",

"questionText": "Vitamin B1 is ascorbic acid.",

"answers": [

{

"text": "False"

},

{

"text": "True"

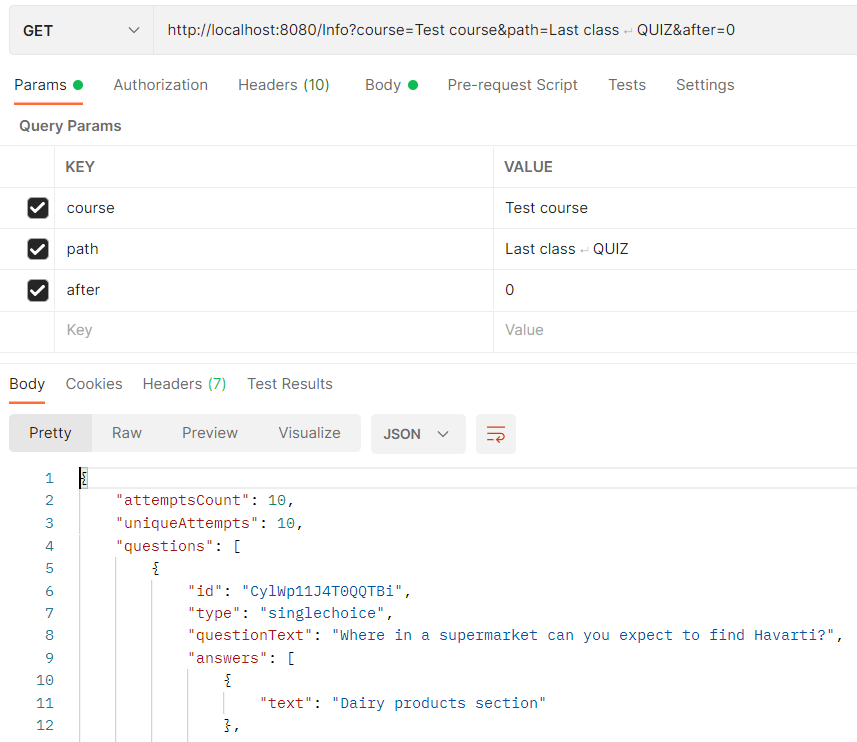
}

]

}

]

}



POST <http://localhost:8080/Attempts>

Visszatérési érték, ha nincs hiba akkor csak egy 200-as státusz kód.

Bemeneti adat:

{

"user": "AAA",

"course": "Test course",

"path": "Last class\nQUIZ",

"questions": [

{

"type": "singlechoice",

"questionText": "Where in a supermarket can you expect to find Havarti?",

"answers": [

{

"text": "Fruit and vegetable stalls",

"checked": **true**

},

{

"text": "Dairy products section",

"checked": **false**

},

{

"text": "Drinks department",

"checked": **false**

},

{

"text": "Frozen food section",

"checked": **false**

}

]

},

{

"type": "singlechoice",

"questionText": "In which 'Tomb Raider' title does cult leader Marco Bartoli stab himself with the Dagger of Xian and turns into a dragon?",

"answers": [

{

"text": "Tomb Raider I",

"checked": **true**

},

{

"text": "Tomb Raider II",

"checked": **false**

},

{

"text": "Tomb Raider III",

"checked": **false**

},

{

"text": "Tomb Raider: The Last Revelation",

"checked": **false**

}

]

},

{

"type": "singlechoice",

"questionText": "Who wrote the 1936 self-help book \"How to Win Friends and Influence People?\"",

"answers": [

{

"text": "Dale Carnegie",

"checked": **false**

},

{

"text": "Noam Chomsky",

"checked": **false**

},

{

"text": "Richard Dawkins",

"checked": **false**

},

{

"text": "Yann Martel",

"checked": **true**

}

]

},

{

"type": "truefalse",

"questionText": "A 12-year old person is considered a teenager.",

"answers": [

{

"text": "True",

"checked": **false**

},

{

"text": "False",

"checked": **true**

}

]

},

{

"type": "truefalse",

"questionText": "\"Azul\" is the Spanish word for red.",

"answers": [

{

"text": "True",

"checked": **false**

},

{

"text": "False",

"checked": **true**

}

]

},

{

"type": "truefalse",

"questionText": "The Southeast Asian speciality of balut is fertilised egg embyros.",

"answers": [

{

"text": "True",

"checked": **true**

},

{

"text": "False",

"checked": **false**

}

]

},

{

"type": "truefalse",

"questionText": "Vitamin B1 is ascorbic acid.",

"answers": [

{

"text": "True",

"checked": **false**

},

{

"text": "False",

"checked": **true**

}

]

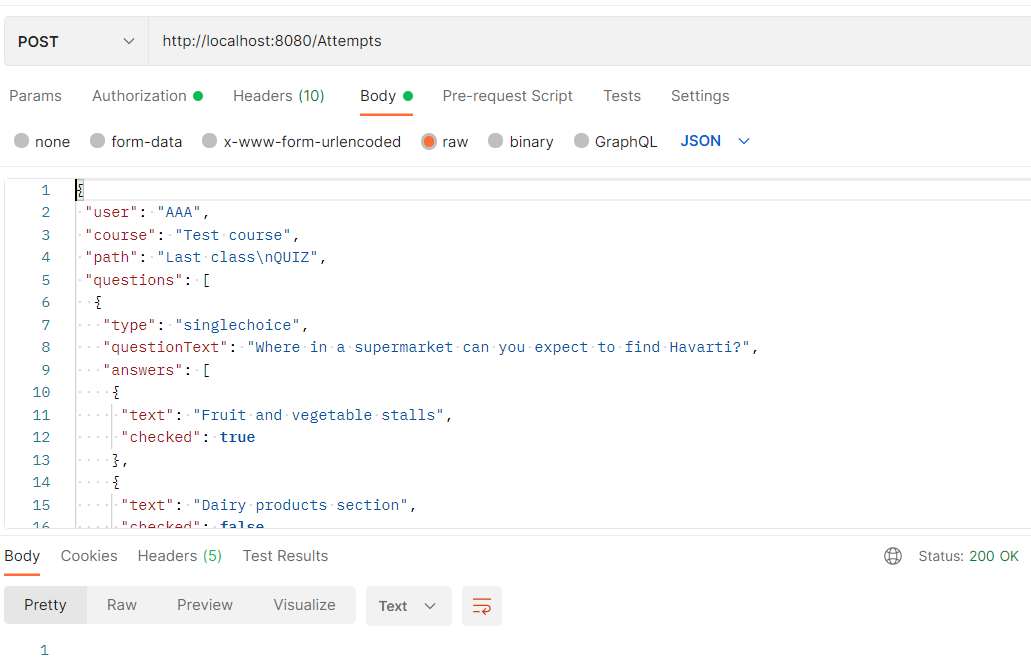
}

],

"score": 4,

"maxScore": 7

}



GET <http://localhost:8080/Vectorized?course=Test%20course&path=Last%20class%0AQUIZ&after=0>

{

"questions": [

{

"id": "xy0FKdBI5\_mF19gz",

"type": "truefalse",

"questionText": "A 12-year old person is considered a teenager.",

"answers": [

"False",

"True"

]

},

{

"id": "tmSyhUZGYI1D45T4",

"type": "truefalse",

"questionText": "\"Azul\" is the Spanish word for red.",

"answers": [

"False",

"True"

]

},

{

"id": "ApIF29TMugF6lW5P",

"type": "truefalse",

"questionText": "The Southeast Asian speciality of balut is fertilised egg embyros.",

"answers": [

"False",

"True"

]

},

{

"id": "3M-M6jR1QpdgkHYY",

"type": "truefalse",

"questionText": "Vitamin B1 is ascorbic acid.",

"answers": [

"False",

"True"

]

},

{

"id": "CylWp11J4T0QQTBi",

"type": "singlechoice",

"questionText": "Where in a supermarket can you expect to find Havarti?",

"answers": [

"Dairy products section",

"Fruit and vegetable stalls",

"Frozen food section",

"Drinks department"

]

},

{

"id": "vHOE4EfedYI7OIsK",

"type": "singlechoice",

"questionText": "In which 'Tomb Raider' title does cult leader Marco Bartoli stab himself with the Dagger of Xian and turns into a dragon?",

"answers": [

"Tomb Raider I",

"Tomb Raider II",

"Tomb Raider: The Last Revelation",

"Tomb Raider III"

]

},

{

"id": "Pvl5aGwAKhhE0-ht",

"type": "singlechoice",

"questionText": "Who wrote the 1936 self-help book \"How to Win Friends and Influence People?\"",

"answers": [

"Yann Martel",

"Noam Chomsky",

"Richard Dawkins",

"Dale Carnegie"

]

}

],

"vectorizedAnswers": [

[

0.9821045703260873,

0.017895429673912742,

0.9821045703260873,

0.017895429673912742,

0.5038253162716602,

0.4961746837283398,

1,

0,

0.48592988659774744,

0.4961746837283398,

0.017895429673912742,

0,

0.14659932055960095,

0.017895429673912742,

0,

0.8355052497664863,

0.14659932055960095,

0.5038253162716602,

0,

0.34957536316873883

],

[

0.9178591882898224,

0.08214081171017758,

0.9178591882898224,

0.08214081171017758,

0.27754855827901814,

0.7224514417209819,

0.8045922534311595,

0.19540774656884055,

0,

0.5639821106753433,

0.08214081171017758,

0.3538770776144791,

0,

0.4290984638381021,

0.5709015361618979,

0,

0.3538770776144791,

0.4290984638381021,

0.2170244585474188,

0

],

[

0.9735876430629299,

0.0264123569370701,

0.9735876430629299,

0.0264123569370701,

0.40734122064445355,

0.5926587793555465,

0.6190711362926166,

0.38092886370738344,

0,

0.6832470938542515,

0.0264123569370701,

0.2903405492086784,

0.30231823014686804,

0.6976817698531319,

0,

0,

0.30231823014686804,

0.40734122064445355,

0.2903405492086784,

0

],

[

1,

0,

1,

0,

0.4091488611996737,

0.5908511388003262,

1,

0,

0.4091488611996737,

0.5908511388003262,

0,

0,

0,

0,

0,

1,

0,

0.4091488611996737,

0,

0.5908511388003262

],

[

1,

0,

1,

0,

0.5937875882720159,

0.40621241172798395,

0.7248079225275816,

0.27519207747241825,

0.31859551079959775,

0.6814044892004022,

0,

0,

0,

0.27519207747241825,

0.1762619212125216,

0.54854600131506,

0,

0.5937875882720159,

0.1762619212125216,

0.22995049051546232

],

[

0.40443854003487034,

0.5955614599651297,

0.40443854003487034,

0.5955614599651297,

0.5955614599651297,

0.40443854003487034,

1,

0,

0.40443854003487034,

0,

0.5955614599651297,

0,

1,

0,

0,

0,

1,

0,

0,

0

],

[

0.7135204656275961,

0.2864795343724039,

0.7135204656275961,

0.2864795343724039,

0.2864795343724039,

0.7135204656275961,

1,

0,

0.08922755081993676,

0.6242929148076592,

0.2864795343724039,

0,

0.3757070851923407,

0.1273731123165663,

0.32668427130101757,

0.17023553119007545,

0.3757070851923407,

0.1273731123165663,

0.32668427130101757,

0.17023553119007545

],

[

1,

0,

1,

0,

0,

1,

1,

0,

0,

0,

0,

1,

0,

0.11988695765042665,

0.8801130423495733,

0,

0.8801130423495733,

0,

0.11988695765042665,

0

],

[

1,

0,

1,

0,

0.22962379137529376,

0.7703762086247061,

0.7703762086247061,

0.22962379137529376,

0,

0.6881251198416248,

0,

0.31187488015837517,

0,

0.5414986715336689,

0.21003691342582528,

0.2484644150405058,

0,

0.22962379137529376,

0.5219117935842005,

0.2484644150405058

],

[

0.5665872761187227,

0.4334127238812773,

0.5665872761187227,

0.4334127238812773,

0.4334127238812773,

0.5665872761187227,

1,

0,

0,

0.5665872761187227,

0.4334127238812773,

0,

0.4334127238812773,

0,

0,

0.5665872761187227,

0.4334127238812773,

0,

0,

0.5665872761187227

],

[

1,

0,

1,

0,

0,

1,

1,

0,

0,

0.8263392723949821,

0,

0.17366072760501797,

0,

0.576036599297548,

0,

0.42396340070245203,

0,

0.40237587169253003,

0.17366072760501797,

0.42396340070245203

]

],

"scores": [

4.313784503820914,

3.7918605355698882,

3.856586971627155,

4.590851138800327,

3.9547584130430438,

2.6177541601394814,

3.5273975912093665,

4.119886957650427,

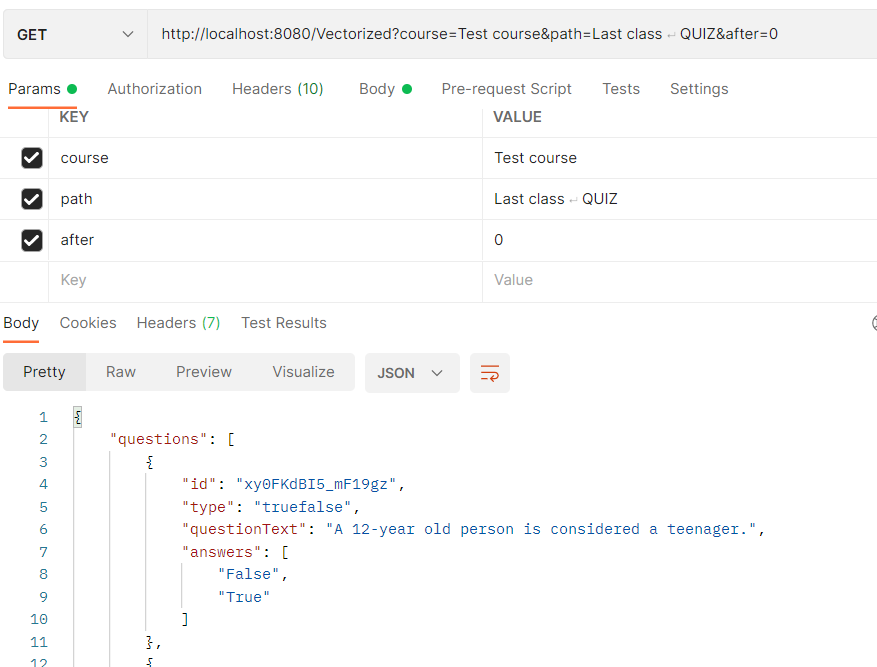
4.330715503823587,

3.266349104474891,

5

]

}



GET <http://localhost:8080/Solver?course=Test%20course&path=Last%20class%0AQUIZ&after=0>

{

"truefalseQuestions": [

{

"id": "xy0FKdBI5\_mF19gz",

"type": "truefalse",

"questionText": "A 12-year old person is considered a teenager.",

"answers": [

"False",

"True"

]

},

{

"id": "tmSyhUZGYI1D45T4",

"type": "truefalse",

"questionText": "\"Azul\" is the Spanish word for red.",

"answers": [

"False",

"True"

]

},

{

"id": "ApIF29TMugF6lW5P",

"type": "truefalse",

"questionText": "The Southeast Asian speciality of balut is fertilised egg embyros.",

"answers": [

"False",

"True"

]

},

{

"id": "3M-M6jR1QpdgkHYY",

"type": "truefalse",

"questionText": "Vitamin B1 is ascorbic acid.",

"answers": [

"False",

"True"

]

}

],

"optionsQuestions": [

{

"id": "CylWp11J4T0QQTBi",

"type": "singlechoice",

"questionText": "Where in a supermarket can you expect to find Havarti?",

"answers": [

"Dairy products section",

"Fruit and vegetable stalls",

"Frozen food section",

"Drinks department"

]

},

{

"id": "vHOE4EfedYI7OIsK",

"type": "singlechoice",

"questionText": "In which 'Tomb Raider' title does cult leader Marco Bartoli stab himself with the Dagger of Xian and turns into a dragon?",

"answers": [

"Tomb Raider I",

"Tomb Raider II",

"Tomb Raider: The Last Revelation",

"Tomb Raider III"

]

},

{

"id": "Pvl5aGwAKhhE0-ht",

"type": "singlechoice",

"questionText": "Who wrote the 1936 self-help book \"How to Win Friends and Influence People?\"",

"answers": [

"Yann Martel",

"Noam Chomsky",

"Richard Dawkins",

"Dale Carnegie"

]

}

],

"solverResult": {

"deltas": [

0,

0,

0,

0,

0,

0,

0,

0,

0,

0,

0

],

"binaryAnswers": [

1,

0,

1,

0,

0,

1,

1,

0

],

"optionsAnswers": [

1,

0,

0,

0,

0,

1,

0,

0,

0,

0,

0,

1

]

}

}

