קורס Web Services - תרגיל 1 להגשה: שלב ראשון, צד השרת

חובת הגשה. הגשה מיד אחרי פסח ללא DB. אפשרות לשיפור ציון לאחר מכן, להוספת DB.

בשלב השני סטודנט X יקבל לתכנת את צד הלקוח עבור WS של סטודנט Y.

:ציון

- 1. 81-100 [אם ירוץ תקין אצלי על המחשב + הדאטה יגיע מה- DB]
- 2. 60-80 [אם ירוץ תקין אצלי על המחשב. דאטה מקובץ ison סטטי על השרת]
 - 3. נכשל [אם יהיו הודעות שגיאה ולא יהיה פלט נכון]

:Deployment

:GitHub .1

- את הקוד לבדיקתי. כ GitHub → מעלים ל
 - שמים **במודל** את הקישור.
 - ס באתר שלי יש מצגת נפרדת ⊙

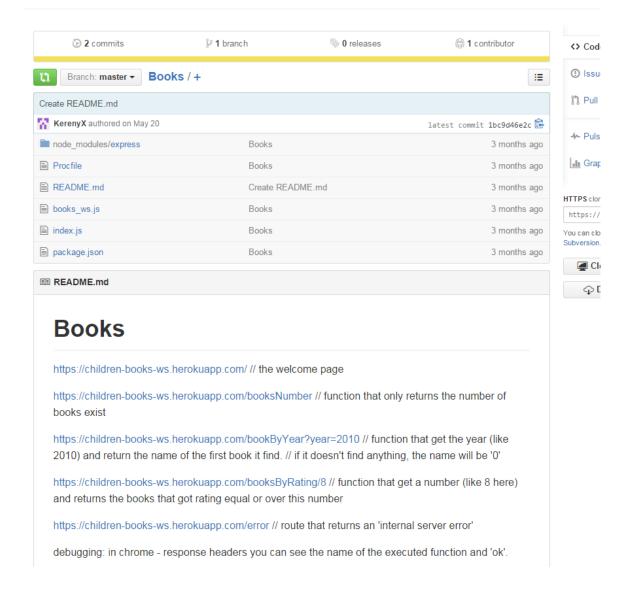
:Heroku .2

- את הקוד להרצת ה- WS סמעלים ל- Heroku את הקוד להרצת ⊙
 - o שמים במודל את הקישור להרוקו ול- API: ⊙
 - שם ה- WS ומטרתו
- שמות 3 הפונקציות, תיאור הפרמטרים ששולחים אליהן, הפרמטרים שמקבלים בחזרה
 - תיאור אפשרות לדיבאגינג למתכנת שיעבוד בצד הלקוח
 - באתר שלי יש מצגת לעבודה עם הרוקו 🏻 🔾

לא יתקבלו תרגילים באיחור. נקודות למחשבה: קוד יעיל, מינימליסטי, מסודר, בעל משמעות, עם הערות בקוד

הוראות התרגיל: WS לניהול סטודנטים בשנקר

- 1. יש ליצור WS, שיספק לאפליקציה בצד הלקוח 3 נתונים מסוימים מתוך אובייקט JSON ששמור ב- mLab.
- StudentsGrades או ExcellenceStudents שלכם. לדוגמה, WS שלכם של ה- WS שלכם. לדוגמה,
 - getAllExcellenceStudent לאובייקט ללא פרמטר, כמו get dellExcellenceStudent .1
 - id לפי getStudGrade לנתון לפי פרמטר שמגיע, כמו
- 3. פונקציית get aurc, כמו getExcellenceByYear שיחזיר את פרטי הסטודנטים עבור שנה get מורכב יותר, כמו מסוימת שיגיע כפרמטר (הצלבת נתונים והחזרת רשימת נתונים).
 - 2. לכל פונקציה ניתן לגשת על ידי פנייה ל- route מסוים.
 - 3. הפונקציות חייבות להיות מיוצאות ממודול
 - 4. האובייקט המוחזר חייב להיות אובייקט JSON תקני
 - .5 העמוד המפעיל את ה- WS יקרא לכל הפונקציות שבמודול.
- 6. המידע יישלח לדפדפן באובייקט response בצורת JSON. כדי לבדוק את עצמכם, צפו בתוצאה בדפדפן ... וראו אם אתם מקבלים JSON תקני, בו יוכל מתכנת אחר לעשות שימוש בשלב השני.
- 7. קובץ package חייב להכיל את השם שלכם, את שם ה- WS ואת כל ה- package חייב להכיל את השם שלכם, את שם ה- שימוש, במידה ויש כאלו.
 - 8. אני ממליצה להתחיל עם json סטטי בספריית data ורק כשיעבוד, תעלו את ה-json ל- mLab.



https://index-ws.herokuapp.com/find?book_type=drama

Book Store Web Service API

this web application created to give numerous ways to retrieve text books from our inventory.

the books are returned in JSON objects containing an array of text books

all the requests should be in GET method

example of JSON object with one book:

Obtaining all Text Books

in order to obtain every text books one needs to add the following link:

```
https://ws-ex1-dudi.herokuapp.com/textbooks
```

the result will be an JSON object containing an "textbooks" key and an array with all the text books objects.

Obtaining one Text Book by ISBN

in order to obtain a textbook bt a certain ISBN one need to add the following link:

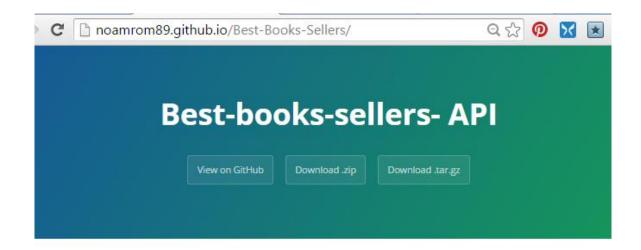
```
https://ws-ex1-dudi.herokuapp.com/isbn/:isbn
```

where ":isbn" should be the ISBN of the book

the result will be an JSON object containing an "textbooks" key and an array with the text book with that ISBN if the book was not found the result will be an empty array

Obtaining all text books by the same publisher

in order to obtain textbooks by a certain publisher one need to add the following link:



Best Sellers Books - API specification

This is a web service for third-party who want to use the Best-Sellers-Books app.

In this app we will use Node js for the basic functionality.

First thing we will need to do is to make the server (server.js) running...

URL	Description	Example
https://best-sellers-books.herokuapp.com/	Showing all the best sellers books by Json objects	{ Name: "Ronaldo", year: "1989", BSMonth: "March" },{},{}
https://best-sellers- books.herokuapp.com/name/BOOKID	Showing the requested book with the specific name of it (BOOKID)	{ Name: "Ronaldo", year: "1989", BSMonth: "March" }
https://best-sellers- books.herokuapp.com/BSForMonth/bookmonth	Showing array of books with the requested month of them/it (bookmonth)	{ Name: "Gerrard", year: "2010", BSMonth: "June" }, { Name: "Zidan", year: "2011", BSMonth: "June" }

Best-books-sellers- API is maintained by NoamRom89.

This page was generated by GitHub Pages using the Cayman theme by Jason Long.

WELCOME TO SPORT BOOKS WEB SERVICE

show all sport books:

if you want to get all the availble sportbooks click on the link bellow: https://sportbooks.herokuapp.com/showAllSportBooks

show book by specific ID:

if you want to get a specific book you need you use the link bellow and you can change the id number in the query string in this en https://sportbooks.herokuapp.com/id?id=111

show book in range of prices:

if you want to get books in a specific price range you need you use the link bellow and you can change the from and to numbers in https://sportbooks.herokuapp.com/betweenPrice?from=2&to=80

:אבישי חג'בי



Shoes store

structure: id, size, company, useFor

GET getAllShoes example: /getAllShoes

GET getShoeById

example:/getShoeById/?id=123

GET insertNewShoe

example: /insertNewShoe/?id=123&size=42&company=nike&useFor=running

GET deleteShoeById

example: /deleteShoeById/?id=123

GET findShow

example: /findShoe/?id=123&size=42&company=nike&useFor=running

GET updateShoe

example: /updateShoe/?id=123&size=42&company=nike&useFor=running

any time you see a multiple params you can ignore the params you want

Github optional folder structure:

shenkar web services course -- ping pong game

