

# **5.1**

## **תרגיל הגשה 1**

**שירותי מארג  
האינטרנט  
2020**



# תרגיל 1 להגשה – “צרות בהיוטק” שרות Express

(כל התמונות במצגת שייכות לי...)

## תרגיל 1 להגשה – פרטיים

משקל התרגיל הוא 10% מהציון הכללי בקורס.  
תאריך ההגשה הסופי הוא 20.12.20.4 ב-6 בבוקר. אין הגשות באיחור.  
הקוד יבדק מהתיקיה בתחום ה-Github Classroom הרלונטי. לינק לפוסטמן יש לשים בהגשה שבמודול.  
[הلينك لرفوديتوري:](https://classroom.github.com/a/iqvZrEhQ) <https://classroom.github.com/a/iqvZrEhQ>

### פירוט הציון:

- 40% – הרכבת שרת, API תיקני, פועלות CRUD עובדות, ניהול גיט / גיטהאב
- 15% – מודול לוגים הכלול שמירה וצפיה
- 15% – Postman – שמירת הביקשות מהשרת ופיבולוש בענן
- 20% – קוד מסודר, קריא, נקי, מינימליסטי, יעיל, הערות והדפסות רק בשציריך, שימושים שימושתיים
- 10% – טיפול בשגיאות ומרקרי קצה

**5 נקודות בונוס** – למי שיגיש בשבוע הראשון (עד ה-27.11.20) (עד ה-20.11.20)

# הנושאים בתרג'il 1 להגשה

Node.js & Express

Restful API

CRUD Operations

Postman & Documentation

Npm module

Node built-in module

Custom modules

Working with Json and objects

ES6

Error handling

## תרגיל 1 להגשה – הבוננה

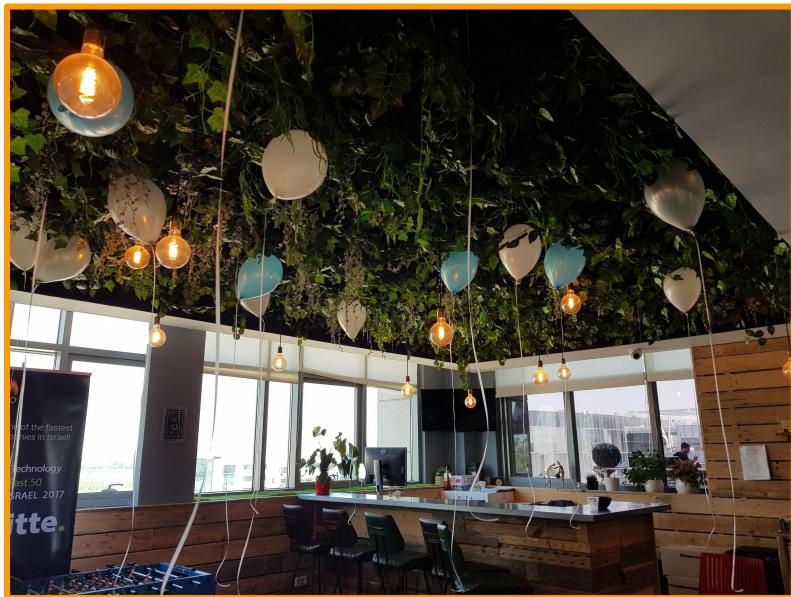
- ❖ השתמשו ב-`node-mongo`, אך לא תצטרכו להפסיק ולהפעיל את השירות בכל שינוי
- ❖ בשאותם בונים את ה-`API` `Restful` (`פעולות CRUD`) תבננו קודם.  
בדקו שהכל עובד דרך הפוסטמן ורק אחורי זה חברו את הדטה האמיתית מתוכה-
- ❖ חשבו על מקרים קצה ונסו להתמודד איתם כך שהקוד לא ישבר אלא ייחזיר הודעות מתאימות
- ❖ חזרו על הדוגמאות שבמצגות, קראו דוקומנטציה אם צריך ואם אתם נועדרים בקוד שמסתוובב באינטרנט – הסתבלו על התאריך ועל התגובה.
- ❖ הסקאך לרשותכם כדי להיעזר אחד בשני מבחינה רעיונית (ללא העתקת קוד).

# שלב 1 – Restful API



יום ראשון. תחילת שבוע חדש. הגענו לעבודה.  
ראש הצוות מעדכן בדילוי שיש משימה חדשה  
וממש דחופה, עליינו לבצע אותה בהקדם.  
הוא רוצה שנקים שירות Node.js בעזרת Express.  
השירות הזה יידע לבצע פעולות CRUD על משתמשים  
במערכת.  
לכל משתמש במערכת יש מס' סידורי, שם פרטי, שם  
משפחה, אימייל, מגדר, ארץ וצבע (ערך hex).

# שלב 1 – Restful API המשך



מסד הנתונים לא זמין בעת, לבן נשתמש בקובץ JSON עם פרטיים של 5,000 משתמשים (data/users.json).

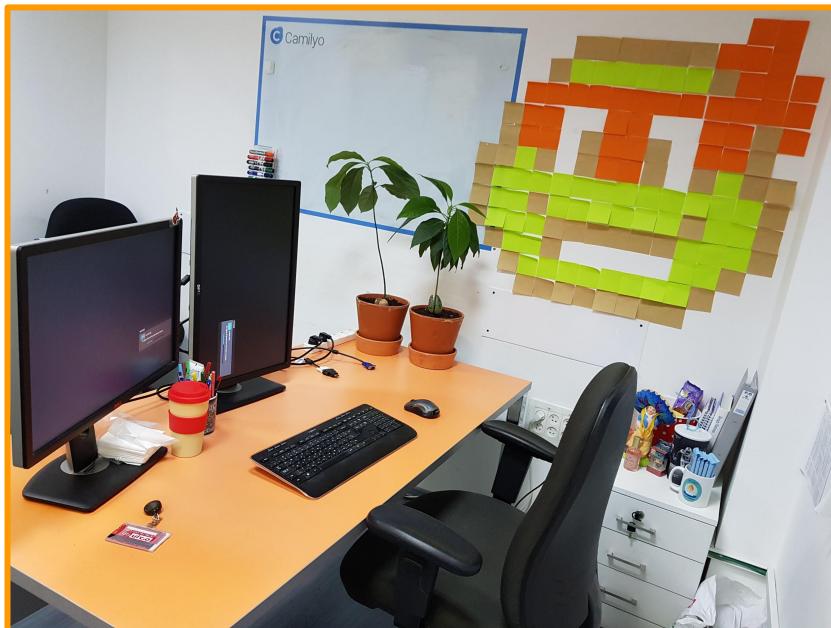
בהמראבת עולה בפעם הראשונה הפרטים נטענים לתוך אובייקט, והפעולות CRUD מבוצעות על האובייקט זהה (אבל לא נשמרות לתוך קובץ ה-JSON).

ראש הוצאות מדגיש שצריך להיות ראות דיפולטי שיחזיר למשתמש קוד סטטוס מתאים והודעה שהפעולה שביקש לא חוקית.

אנחנו ניגשים לעובדה ובמובן נעצרים בפוסטמן.

## שלב 2 - logs

חבר צוות הציע לנו לבתוב לוגים לكونסול, ומוסיף שזה יהיה יותר נחמד אם הלוגים יהיו בצבעים לפי הסוג שלהם, למשל שגיאה תודפס באדום. הוא אומר שבכדי לנו לבדוק את החבילה colors מתוך WK.



בגלל שהלוגים יישמשו איזוריים בכל המערכת, אנו ממקמים את הפונקציונליות במודול חיצוני בשם `logs` ונשתמש בו בכל מקום שנצרך במערכת. הלוגים נשמרים מתוך המערכת, אבל משתמש תיה אפשרות לראות אותם ע"י פניה לראוט `logs`.

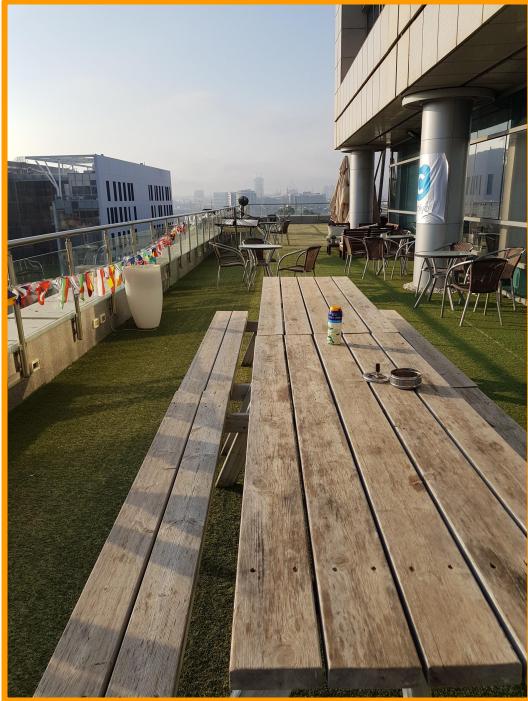
## שלב 3 – Saving backup logs



היא מזהירה אותנו לא לעשות את הטעות שהיא עשתה ביום הקודם, צריך להוסיף לקובץ ולא לדחוס אותו בכל הדפסה חדשה. ובמובן לא לשכוח לשימוש ב-`moment` כדי לשמור את התאריך והזמן המדויקים של כל אירוע בלוג

סיימנו את המשימות שניתנו לנו עד עבשו.  
החליטנו שmaguire לנו מנוחה על הנדנדה במרפסת.  
בדרכו חזרה למשרד אנחנו פוגשים מתכנתת מצוות  
אחר, מספרים על המשימה שסיימנו היום  
והיא מציעה לשמר את הלוגים לקובץ כדי שיישאר  
תיעוד. יש שהוא בדבריה.  
נשמר את הלוגים לקובץ מקומי בשם `logs.txt`  
נשתמש במודול המובנה `fs` של Node.

## שלב 4 – שאלות



אוקיי, המשימה שהוגדרה בוצעה בהצלחה.  
בינתיים אנחנו יוצאים לאבול ארוחת צהרים עם החברים במרפסת.

ראש הוצאות סיימ ישיבה וחוזר למשרד.  
הוא מוסיף שזה הגיוני שנייתן למשתמש במערכת אפשרות לבצע  
חיפוש לפי ארצ. אנחנו מוסיפים את הfonקציונליות הדו לאחד  
הרואטים שבבר בנינו בשלב הראשון.

## שלב 5 – Postman



סיימו את המשימה. וידאנו בגיטהאב שהקוד המעודכן נמצא שם. פותחים Pull Request וחברת צוות תעשה לנו code review. למקורה שהיא תרצה לבדוק את הפונקציונליות אנחנו מפרסמים את הקריאה API שעשינו מתוך פוסטמן (למרות שהקריאה לוקאלית) ונותנים לה Link, עם דוקומנטציה מושלמת כדי שלא תפריע לנו אחרי שעות העבודה.

בבוקר יומם שני אנחנו מגיעים למשרד.

מודיעים לנו שהפרויקט היה כל כך טוב והזמין לנו טישה לטקסס, ארה"ב ...

**בהצלחה !**

