



פרוייקט NODEJS MONGO

יצירת מערכת. צעצועים
בקצרה על המערכת:

במערכת כל משתמש (רשום) יוכל להוסיף צעצוע עם שם, תיאור, קטגוריה, תמונה, מחיר, בנוסף יהיה ניתן לשלוח מידע שונה על צעצועים שנמצאים במסד נתונים

חלק 1

יש לייצר קולקשיון שנראה כך ב DB חדש עדיף בשם TOYS:

_ID	
name:string	שם המוצר
info:string	תיאור המוצר
category:string	קטגוריית המוצר (שם של הקטגוריה)
img_url:string	תמונת המוצר ב URL (ניתן להביא מאתר פיקסילס) - לא חובה לשיגור
price:number	מחיר
date_created:date	תאריך שנוסף בו המוצר (ברירת מחדל)
user_id : String	ישמש בחלק 2 של המשימה לקישור בין

	משתמש למוצר (לפי TOKEN)
--	-------------------------

תפקידכם במשימה זו לייצר את הצד שרת בלבד ולא צד לקוח ובעצם לספק API REST לצד לקוח שבו מתכנת צד לקוח יכול לבצע את הפעולות הרשומות מעלה,

להלן הכתובת הרצויות לביצוע פעולות ב API:

מייצג את הלוקהלוסט או הדומיין שהעלתם להירוקו -> Domain

domain/toys מספר עמוד לפי -> page? 10 צעצועים בדף	בקשה כללית לצעצועים במערכת - GET	ראוט א'
domain/toys/search קווארי של חיפוש -> s? החיפוש יתבצע במאפיין Name או info מספר עמוד לפי -> page? 10 צעצועים בדף	בקשה לחיפוש מוצר לפי השם או ה INFO שלו	ראוט ב'
domain/toys/category/: catname מספר עמוד לפי -> page? 10 צעצועים בדף	בקשת צעצוע לפי קטגוריה	ראוט ג'
	* מה שמסומן באדום - מחייב שהפעולה תתבצע עם TOKEN לפי חלק 2	

domain/toys/	*הוספת צעצוע- POST (לשים לב שיוסיף את ה-USER_ID של המשתמש שהוסיף אותו לפי הטוקן)	ראוט ד'
domain/toys/:editId	*עריכת צעצוע- PUT לפי הפרמס של ה-EDITID ולדאוג שהרשומה לפי הטוקן קשור ל-USER_ID	ראוט ה'
domain/toys/:delId	*מחיקת צעצוע- DELETE לפי הפרמס של ה-DELID	ראוט ו'
domain/toys/prices מספר עמוד לפי -> ?page 10 מוצרים בדף מחיר מנימום של - ?min מוצר מחיר מקסימלי של - ?max מוצר דוגמא להצגת מוצרים שהמחיר שלהם בין 10 ל-40 localhost:3000/toys/prices/?min=10&max=40	בנוס - יכולת לקבל מוצרים לפי טווח מחירים מנימום ומקסימום אפשר לקרוא על שאלות של גדול קטן וכו' במונגו כאן: https://docs.mongodb.com/manual/reference/operator/query-comparison/	ראוט ז'
domain/toys/single/:id	ישלוף רק רשומה אחת לפי איי די	ראוט ח'

חלק 2 המשך שכולל את כל חוקי האבטחה של AUTH (עם טוקן ו BCRYPT)

- צור קולקשן של USERS ואת שבנוי בצורה הבאה:
Name,Email,Password,date_created,role
- המאפיין ROLE - הוא סוג היוזר שיכול להיות ADMIN או USER בברירת מחדל
הוא יהיה USER

_id	
-----	--

name:string	
email:string	
password:string	
date_created:Date	
role:string	סוג המשתמש ADMIN או USER משתמש חדש שנרשם בבירור מחדל USER יהיה

*המאפיין EMAIL צריך להיות UNIQUE

- צור ROUTE ב USERS של POST ל "כתובת <http://localhost:3000/users>
שייצר משתמש חדש לפי הסכמה של הקולקשן (לשים לב שה PASSWORD מוצפן עם BCRYPT ושחזירים את המידע לצד לקוח לשים לב לכל חוקי האבטחה שלמדנו)

- בנוסף לכל יהיה כתובת שתאפשר למשתמש להתחבר בבקשת : POST
<http://localhost:3000/users/login>
שבו ישלח ב BODY רק המאפיינים EMAIL ו PASSWORD

במידה והאימייל והסיסמאות תקניות (שים לב לשמור את ה PASSWORD מוצפן עם מודול BCRYPT וגם לבדוק אותו מוצפן מול ה BCRYPT) המשתמש יקבל בחזרה , TOKEN

יש להוסיף לסכמה של הקולקשן TOYS מאפיין/קיי בשם USER_ID (שבעצם יציג לאיזה משתמש הצעצוע שייך)

• בROUTER של ה TOYS כל בקשות ה DELETE , POST ו PUT יחייבו שהמשתמש ישגר TOKEN תקני ב HEADER עם קיי בשם "x-api-key" ובמידה והטוקן לא תקין המשתמש יקבל הודעת שגיאה ב JSON שמתאימה לבעיה (יש

לבצע את הבדיקה עם פונקציית MIDDLEWARE כמו שלמדנו).

* יש לדאוג בפרוייקט להשתמש ב **JOI** , **BCRYPT** במקומות הנכונים , וכמובן
MIDDLEWARE של **AUTH** לכתובות שיש צורך בהם ב **TOKEN** עם ספריית
JWT

מומלץ שבמערכת יהיו קיימים כ 12 צעצועים עם 3 קטגוריות שונות.

יש לספק דוקומנטציה שמסבירה כיצד להשתמש ב API אפשר להסתכל בכתובת הבאה
לכתיבת דוקימנטציה נכונה (במיוחד כיצד לגשת לכתובות השונות)
<https://jsonplaceholder.typicode.com/>
יש להגיש את כתובת ה GITHUB של הפרוייקט (שימו לב שאין שם דברים סודיים כמו ה
ENV או חשיפה של סיסמאות)

בנוסף: נסו לעלות את הפרוייקט לשרת CYCLIC כפי שלמדנו (עונה 11 פרק אחרון)

ofer.shelli@gmail.com

דוגמאות טובות לדוקימנטציה (שאפשר לקחת השראה מהם):

<https://rawcdn.githack.com/kappey/toys4you/9b6e9716a0942930cb3f74e99ec7092d03cfce35/public/index.html>

<https://reqres.in/>

דוגמא בינוניות ומורכבות חלקן(פחות טוב):

[/https://picsum.photos](https://picsum.photos)

<https://api-project2021.herokuapp.com/>

אפשר לעשות שליד כל כתובת יהיה הסבר מפורט במיוחד בבקשות POST מה ניתן ומה לא ניתן לעשות ומה צריך לשלוח ב BODY

לבדיקת קוד GIT:
[/https://raw.githack.com](https://raw.githack.com)

לא להתייחס לא קשור לפרוייקט

פרוייקט תלמידים

כללי :
מערכת API שמאפשרת למורים להרשם למערכת , ולהוסיף /לערוך/למחוק/להציג סטודנטים שקשורים למרצה , כל סטודנט יהיה לו ציון , נושא וכיתה

המערכת תהיה בנויה מ2 קולקשיין

קולקשיין של הUSERS:

ID	איי די של הרשומה במונגו
Name	
Email	
School*	לא חובה שם הבית ספר
Password	
Date_created	
Role	תפקיד אם אדמין או לא

קולקשיין של הסטודנטים:

ID	
Name	
Class	קוד כיתה של הסטודנט
Score	ציון
Subject	מקצוע
user_id	
createdAt	

ב USERS יהיו לי את הראוטים הבאים

/users (POST)

יאפשר להוסיף משתמש חדש בבאדי יהיה צורך לשלוח

Name, email , password, shcool*

*- לא חובה

/users/login (POST)

יאפשר כניסה למשתמש שאם יש הצלחה יקבל טוקן בחזרה בבאדי ישלח

Email,password

ב STUDENTS:

/students

יחזיר הודעה שזה אזור התלמידים בלי להציג כולם

/students/myStudents (GET) (TOKEN)

יחזיר את כל התלמידים של אותו מרצה

/students (POST) (TOKEN)

יאפשר להוסיף תלמיד חדש יש לשלוח בבאדי:

Name , class, score , subject

/students/:editId (PUT) (TOKEN)

עריכה של הסטודנט לפי האיי די בבאדי ישלח:

Name , class, score , subject

פרמטר בשביל השאילתא שתדע איזה תלמיד לעדכן -> :editId

/students/:delId (DELETE) (TOKEN)

יאשפר מחיקה של הסטודנט לפי האיי די שלו

האיי די של הסטודנט שנרצה למחוק -> :delId