

משימה לתרגול - פרויקט מסכם node.js להלן פרטי המשימה

מטרת הפרויקט

בפרויקט זה נתנסה בבניית rest api בצד שרת.
תשתית זו תשמש אותנו בהמשך גם בפרויקט המסכם את מודול - react.
ידע ויכולת בניית rest api נחשב למשימה מבוקשת ביותר בעולם הפיתוח בהייטק ולכן יש ליחס חשיבות גדולה מאוד לפרויקט זה ומומלץ להשקיע בקריאה ובהבנה של הקוד, המודולים, הקבצים, והתיקיות המרכיבים פרויקט זה!

כלים

פרויקט זה יהיה מבוסס על node js והוא ישלב את המודולים הבאים:

- Express
- nodemon
- mongoose
- bcrypt
- lodash
- joi
- jsonwebtoken
- config
- morgan
- cors
- chalk

* ניתן להשתמש במודולים ובספריות נוספות במידת הצורך

בסיס נתונים

יש להשתמש בפרויקט זה בבסיס הנתונים של MongoDB בלבד כדי להתנסות באופן שמירת נתונים לפרויקט שצפוי להתמודד עם Big Data .
חשוב מאוד לקבל נסיון פרקטי באחד מתוכנות לבסיסי הנתונים הפופולריים השייכים למשפחת ה NoSQL

מידע ועזרים

ניתן למצוא הסברים נרחבים ודוגמאות מפורטות של קוד לגבי אופן הבנייה ושימוש במודולים ובספריות השונות לצורך בניית rest api בצד שרת בעזרת node js בספר "Node.js Web Development" – החל מפרק "rest api A" ואילך. ספר זה פתוח לכם בקמפוס...

להלן דרישות פיתוח הפרויקט

מקרא:

נקודת גישה = endpoint

שיטת = method (POST PUT GET DELETE PATCH)

1. יש ליצור logger שירשום את כל הבקשות שנשלחות לשרת ויפרט את שיטת הבקשה, לאיזה כתובת נשלחה וזמן התגובה לבקשה. ניתן להשתמש בספריית morgan לצורך כך.
 2. יש להשתמש ספריית chalk על מנת לצבוע בקונסול הודעות שגיאה והצלחה בצד שרת. יש להשתמש בגרסה chalk@4.1.1
 3. יש להשתמש בספריית cors כדי לאפשר בקשות לשרת מכתובת ספציפית.
 4. צרו נקודת גישה בשיטת POST שתאפשר הרשמה של לקוח חדש למערכת.
 - 4.1 הלקוח יכול להיות לקוח רגיל או לקוח עסקי.
 - 4.2 בתהליך ההרשמה יספק הלקוח שם, אימייל (יחודי), סיסמה, ודגל המסמן האם הוא לקוח עסקי או רגיל.
 - 4.3 במהלך ביצוע בקשה לנקודת גישה זו, יש לבצע ולידציה על הנתונים שסופקו בעזרת ספריית json במידה והנתונים נמצאו לא תקינים יש להחזיר תשובה המסמנת סטטוס 400 עם מלל המפרט את השגיאה הרלוונטית. והדפסה בקונסול של השגיאה צבועה בצבע אדום בעזרת ספריית chalk
 - 4.4 במידה והנתונים תקינים יש לשמור את פרטי הלקוח החדש בבסיס הנתונים ולהחזיר תשובה המסמנת סטטוס 200 לצד json המייצג את ה id השם והאימיל של הלקוח החדש שנוצר בבסיס הנתונים.
- * יש להצפין את הסיסמה של הלקוח בבסיס הנתונים בעזרת המודול bcrypt.**
5. צרו נקודת גישה בשיטת POST שתאפשר התחברות של לקוח למערכת דרך שליחת אימייל וסיסמה.
 - 5.1 בנקודת גישה זו יש לבצע בשלב ראשון ולידציה מקדימה המוודאת תבנית תקינה לאימייל וכמו כן שהסיסמה באורך של שישה תווים לפחות.

5.2. בשלב שני יש לבצע בדיקה נוספת שהאימייל והסיסמה אכן קיימים בבסיס הנתונים. במידה ולא יש להחזיר תגובה המסמנת סטטוס 400 לצד מלל המתאר את השגיאה : המשתמש או הסיסמה לא נכונים.

5.3. במידה ושני שלבי הבדיקה עברו תקין יש להחזיר תגובה המסמנת סטטוס 200 זאת לצד token (שיוצר בעזרת המודול jsonwebtoken) שיכיל את פרטי המשתמש. ה- id שלו כפי שקיים בבסיס הנתונים וכמו כן את הערך המסמל האם הוא לקוח רגיל או עסקי. לצורך העניין יש להשתמש בספריית lodash

6. צרו נקודת גישה בשיטת **GET** שתאפשר קבלת הפרטים הבאים של המשתמש: שם, אימייל, האם לקוח עסקי, תאריך יצירה ומזהה רשומה.

6.1. לנקודה זו יסופק טוקן, אותו token שהתקבל בעת התחברות מוצלחת (מסעיף 2).

6.2. במידה ונמצא תקין, יש להחזיר תגובה המסמנת סטטוס 200, זאת לצד הפרטים האמורים של המשתמש.

6.3. במידה ולא סופק token או שנמצא לא תקין יש להחזיר תגובה המסמנת סטטוס 401 לצד מלל המתאר את השגיאה הרלוונטית.

7. יש ליצור נקודת גישה בשיטת GET שתאפשר קבלת כל כרטיסי הביקור ששמורים במאגר המידע. אין צורך ב token לשם קבלת הכרטיסים הללו. נקודה זאת תשרת את דף הכניסה הראשי בו יוצגו כלל הכרטיסים שבאתר

8. צרו נקודת גישה בשיטת GET שתקבל id של כרטיס ביקור ותחזיר את הפרטים המלאים של אותו כרטיס ביקור. גם בנקודה זאת אין צורך ב token מהמשתמש. נקודה זאת תשרת הקלקה על כרטיס ספציפי בדף הכניסה על מנת לראות פרטים נוספים עליו.

9. צור נקודת גישה בשיטת GET שתאפשר קבלת מערך של כל כרטיסי הביקור השמורים תחת המשתמש הנוכחי.

9.1. נקודת גישה זו תהייה פתוחה למחברים בלבד ותדרוש את ה token שסופק לאחר התחברות (מסעיף 2)

9.2. נקודה זאת תאפשר קבלת הכרטיסים רק למשתמשים שהם עסקיים.

9.3. אם הייתה בעיה בהתחברות לשרת של מאגר המידע יש לשלוח סטטוס 500 עם הודעת שגיאה מתאימה

10. צרו נקודת גישה בשיטת POST שתאפשר פתיחת כרטיס ביקור חדש

10.1. גישה זו תהיה פתוחה רק במקרה שצורפה לבקשה token אותו טוקן שמקבלים לאחר התחברות (כמופיע בסעיף 2).

10.2. יצירת כרטיס תאפשר רק למשתמשים שהוגדרו עסקיים.

- 10.3 יש לעשות וולידציה על הכרטיס לפני שמעבירים אותו לשמירה בבסיס הנתונים ובמידה ויש שגיאות לעצור את הפונקציה ולהחזיר לגולש חיווי כי הפעולה נכשלה ואת הסיבה.
- 10.4 על הכרטיס שיפתח לכלול את השדות הבאים: שם העסק, תיאור העסק, כתובת העסק, טלפון העסק, תמונת העסק (קישור לתמונה), תעודת זהות של יוצר הכרטיס, מספר מזהה רנדומלי וייחודי.
- 10.5 בעת שמירת הכרטיס יתווספו השדות הבאים: מתי נוצר הכרטיס, ושדה מועדפים likes ששווה ערך למערך של תעודות זהות של משתמשים שסימנו את הכרטיס כמועדף
- 10.6 יש לשמור את הכרטיס שהתקבל בבסיס הנתונים.
- 10.7 במידה והשמירה הצליחה יש להחזיר תגובה המסמנת את פרטי כרטיס הביקור החדש שנשמר בבסיס הנתונים.
- 10.8 במידה והשמירה לא הצליחה יש להחזיר תגובה של הודעת השגיאה סטאטוס 500 והודעה האוטומטית של חברת mongoose.
- 10.9 יש לשמור בכרטיס הביקור מזהה משתמש (user_id) שמכוון לרשומה רלוונטית באוסף משתמשים, זאת כדי שנוכל לזהות איזה כרטיס שייך לאיזה משתמש.
11. נקודת גישה בשיטת PUT שתקבל id של כרטיס ביקור
- 12.1 נקודת גישה זו תהייה פתוחה למחברים בלבד
- 12.2 תאפשר עריכת כרטיס רק אם המשתמש שהוא עסקי.
- 12.3 תערוך ולידציה על האובייקט של הכרטיס שהפונקציה קיבלה מהגולש
- 12.4 במידה ותהיה שגיאה הפונקציה תעצור ותחזיר לגולש סטאטוס קוד 400 עם השגיאה וגם תדפיס בקונסול את השגיאה
- 12.5 יש ליצור קבוע בשם פילטר שיהיה שווה ערך לאובייקט שיכיל תעודת זהות של הכרטיס משורת הכתובות ותעודת זהות של המשתמש ממה שחוזר אלי מפונקציית הביניים של האותוריזציה
- 12.6 יש לחפש במאגר המידע כרטיס שעונה לפרמטרים של הפילטר שיצרנו
- 12.7 יש לבצע התניה שאם לא נמצא כרטיס כזה במאגר המידע יש להחזיר סטאטוס 400 עם הודעת שגיאה מתאימה לגולש
- 12.8 לצורך הבדיקה הוויזואלית (לאחר הבדיקה יש להוריד) יש ליצור משתנה שיחפש את הכרטיס הערוך במאגר המידע ולאחר מכן להציג אותו לגולש
12. נקודת גישה בשיטת DELETE שתקבל id של כרטיס ביקור ותאפשר את מחיקתו.

- 12.1 נקודת גישה זו תהייה פתוחה למחברים בלבד ותדרוש את ה token שסופק לאחר תהליך התחברות (סעיף 2)
- 12.2 נקודת גישה זאת תאפשר מחיקה של כרטיס רק למשתמש עסקי או !admin
13. נקודת גישה בשיטת PATCH שתקבל id של כרטיס ביקור
- 13.1 תבדוק אם המשתמש מחובר
- 13.2 תערוך ולידציה כדי לוודא שהגולש אכן שלח מערך של תעודות זהות
- 13.3 תבדוק במאגר המידע אם יש כרטיס עם תעודת הזהות משורת הכתובות, ואם כן אז היא תעדכן את המפתח likes במערך החדש שהתקבל מהגולש
- 13.4 לאחר מכן היא תשלח בחזרה לגולש את הכרטיס הערוך
- 13.5 אם הוא לא מוצא כרטיס יש להחזיר הודעת שגיאה לגולש ובקונסול שלא נמצא כרטיס עם תעודת הזהות שהגולש סיפק
- 13.6 אם הייתה בעיה בחיבור לשרת של מאגר המידע יש לשלוח על כך הודעה לגולש עם סטאטוס 500

בונוסים

- ליצור נקודת גישה בשיטת PATCH שתקבל תעודת זהות של כרטיס בשורת הכתובות. ותאפשר רק למשתמש שמוגדר כ – admin להפעיל פונקציה שתקבל מספר מה – admin, תבדוק אם המספר כבר תפוס אצל אחד מהכרטיסים. אם הוא לא תפוס אז מספר הכרטיס יעודכן למספר החדש, אחרת תשלח סטאטוס קוד שהעדכון נכשל עם הודעת שגיאה למשתמש כי המספר תפוס
- לעשות ולידציה עם regex על הסיסמה מסעיף 4 כך שרק סיסמה שתכיל תשעה תווים שביניהם לפחות אות אחת גדולה באנגלית ואות אחת קטנה באנגלית, ארבע מספרים וסימן מיוחד.