

Angular-NodeJs-MongoDBMySQL Task V2



הוראות:

1. לצורך ביצוע המשימה יש ברשותכם שלוש וחצי שעות.
חומר עזר: אינטרנט.
2. לאחר סיום המשימה, הבדיקה תתבצע בצורה פרונטאלית עבור כל תלמיד בנפרד.
3. יש להעלות את המשימה לאתר הקורס בלשונית מטלות.
4. **חל איסור מוחלט לדבר ולהיעזר בחברכם לכיתה או כל "חבר טלפוני".**
5. **חל איסור מוחלט ליצור קשר דרך האינטרנט בין חבריכם לכיתה או מול כל גורם אחר.**
6. המשימה מהווה מבחן!
7. התייחסו למשימה כחלק מראיון עבודה.
8. זכרו, אין רק מבחן התוצאה משקף את היכולת והציון שלכם אלא גם הדרך, צורת הכתיבה, best practice דרך וארגון בקוד.



משימה:

במשימה זו עליכם לבנות אפליקציה (SPA) לניהול לקוחות.
כל אחד מבעלי ההרשאה למערכת רשאי להזין למערכת לקוח ולהוסיף משימה לביצוע עבור לקוח.

מבנה האפליקציה:

צד לקוח: Angular

צד שרת: Node

בסיס נתונים: MongoDBMySQL לבחירת התלמיד.

1. בנו בסיס נתונים לפי בחירתכם, MySQL או MongoDB.
 - a. מזהה (ID)
 - b. שם (טקסט)
 - c. תחום עיסוק הלקוח (טקסט)
 - d. טלפון (טקסט)
 - e. כתובת דואר אלקטרוני (טקסט)
- ב. בסיס הנתונים יחזיק טבלה\אוסף המכיל את רשימת המשימות של הלקוחות, שדות:
 - a. ID
 - b. תיאור המטלה (טקסט)

- c. תאריך (היום בו נוצרה המשימה)
- d. מזהה של הלקוח אליו המטלה שייכת
- e. האם המשימה בוצעה (בוליאני)

2. בנו API ע"י שימוש ב *NodeJS Express* , *API* יחשוף את המידע הבא:
 - א. כלל הלקוחות – *GET* מחזיר את כל הלקוחות.
 - ב. רשימת המשימות – *GET* מחזיר את רשימת המשימות (חשוב מאוד!! להחזיר את הלקוח אליו שייכת המשימה).
 - ג. הוספת משימה – *POST* מוסיף משימה (תיאור, תאריך, מזהה).
 - ד. **בנוס בלבד – עדכון משימה (לביצוע רק במידה וכל המשימה הושלמה)**
3. בנו אפליקצית צד-לקוח – *Angular* שתשמש לאינטרקציה עם בני המשפחה.
4. הרכיבו את האפליקציה – צד לקוח מקומפוננטות
 - א. קומפוננטה 1- הדף הראשי (default) יציגת רשימה של כל המשימות.
 - ב. קומפוננטה 2 – תהיה נגישה מהדף הראשי – ותכיל קומפוננטה שתהיה אחראית על הוספת משימה.
 - ג. **בנוס – להוסיף אפשרות מחיקה מהטבלה בדף הראשי – כפתור שיבצע מחיקת משימה לכל שורה.**

5. משימות לדוגמה,
 - א. לשלוח ללקוח פקס
 - ב. להתקשר ללקוח בעוד יומיים
 - ג. להכין ללקוח הצעת מחיר



טיפים:

1. התחילו מ-DB.
2. התקדמו API, חברו אותו לDB וחשפו את הבקשות, בדקו זאת בעזרת *PostMan\RestAdvanced*.
3. לבסוף כתבו את ה UI.
4. חשוב לכתוב קודם את 2 הדפים + מימוש של ראוטר במודול הראשי.
5. לאחר מכן כתבו *Service* וממשו בו את הבקשות הבאות:
 - getCustomers* – מחזיר את כלל הלקוחות – שם, כינוי, מזהה.
 - getTasksList* – מחזיר את כלל המשימות עם המידע הרלוונטי – שימו לב שיש צורך לעבד את האובייקטים שחוזרים מהשרת, להוסיף להם במקום מזהה את השם של הלקוח, בטבלה מוחזק ID ובפועל צריך לחזור שם.
 - addTodoList* – צרו אובייקט זהה בצד לקוח שיוכל לקבל את כלל הפרמטרים של משימה, פרט לID כמובן שהוא מתחולל בDB.
6. כתבו עבור כל דף *Controller* שייצג את הלוגיקה + קריאה ל *Service*.
7. שימו לב שיש הבדל בביצוע הקריאות, בדף הראשי – הקריאה מתבצעת בעליית ה *Controller* לעומת זאת בדף ההוספה הקריאה להוסיף מטלה מתבצעת בלחיצה על *Add*.

Home page

Home	Customers	Tasks
------	-----------	-------

Task #	Customer	Date	Done?
1	Customer 1	28/08/2017	<input type="checkbox"/>
2	Customer 2	14/09/2017	<input type="checkbox"/>
3	Customer 4	20/09/2017	<input type="checkbox"/>

Tasks page

Home	Customers	Tasks
------	-----------	-------

Write here a task and assign to a Customer from the drop down below

Select Customer ▼

בהצלחה!!

28170-4578-01.vs2

© כל הזכויות שמורות לג'ון ברייס הדרכה בע"מ מקבוצת מטריקס