

פרויקט - סביבות פיתוח באינטרנט - עבודה 3.2 - פיתוח Backend

בחלק זה של העבודה תפתחו את השרת ובסיס הנתונים, על פי ה-API אותו כתבתם והאפיון שקיבלתם.

להלן דגשים שיסייעו לכם במימוש העבודה:

- בסיס הנתונים יהיה בשירות הענן Azure כפי שהוצג לכם בתרגולים.
 - תחת מטלה 3.2 באתר הקורס, מופיע מסמך הסבר לפתיחת חשבון במידה וטרם פתחתם.
 - כמו כן, להלן תוסף ל-VSCODE אשר יאפשר לכם לעבוד בקלות מול ה-AZURE DB:
 - <https://marketplace.visualstudio.com/items?itemName=ms-mssql.mssql>
 - <https://docs.microsoft.com/en-us/sql/visual-studio-code/sql-server-develop-us-e-vscode?view=sql-server-ver15>
- מידע בו תעשו שימוש בשרת:
 - מקורות המידע של השרת שלכם נחלקים ל:2:
 1. שימוש ב-API חיצוני : <https://spoonacular.com/food-api/docs>
 2. שימוש בבסיס הנתונים שלכם

שימוש ב-API חיצוני:

כל הנוגע למידע על מתכונים **כלליים**: חיפוש + הבאת מידע כללי על המתכון, יעשה באמצעות תשאול של ה-API.

לנוחיותכם, להלן המשאבים הרלוונטיים לכם:

- Search Recipes
- Get Recipe Information
- Get Random Recipes

1. לפני שאתם מתחילים לעבוד מול ה-API, הבינו איזה מידע נדרש מכם להביא, ולאחר מכן, קראו את המידע אותו מספק ה-API. שימו לב להשתמש בפרמטרים אותו ה-API מספק כך שתקבלו מידע שרלוונטי עבור הדרישות שלכם.
 2. צאו מנקודת הנחה כי השרת יכול לתשאל תמיד את ה-API, ואין צורך לשמור מידע מיותר ב-DB.
 3. שימו לב, ישנה מגבלה יומית מסוימת של תשאול ה-API, מגבלה זו הינה גדולה, ולא צריכה להיות בעיה לשלוח בקשות רבות ביום.
- עם זאת, הדרך הכי פשוטה להתגבר על זה, היא לאחר שבדקתם שהקוד שלכם עובד, שמרו חלק מהמידע שחזר לכם, והשתמשו בו כמידע סטטי באפליקציה לצורך הפיתוח (בלבד).

שימוש בבסיס הנתונים:

כל הנוגע למידע **אישי** על משתמשים/מתכונים: מתכונים משפחתיים, מתכונים אישיים, פעולות משתמשים וכו', ינוהל בבסיס הנתונים.

הנחיות כלליות לעבודה:

1. עליכם להקפיד על כללי הפיתוח והדגשים שנלמדו לפיתוח צד שרת בהרצאות ובתרגולים.
2. עליכם להשתמש בספריות שנלמדו בקורס בלבד - כל ספריה נוספת, נא לאשר מול הסגל.

3. הקלה לעבודה - אין צורך לעשות בדיקות קלט בהרשמה של משתמשים בצד שרת.
תהיו חייבים לעשות זאת רק בצד לקוח.
4. עליכם להקפיד על כתיבת קוד נכונה ונקייה
 - a. הפרדה למודולים
 - b. הפרדה לפונקציות - לוגיקה מורכבת צריכה להיות בפונקציות, אם צריך אז לפצל למספר פונקציות
 - c. שמות משתנים משמעותיים
 - d. וכד'
5. הגשת העבודה תעשה באותם זוגות שהתחילו את הפרויקט
6. עליכם לממש את השרת והroutes שלו, על פי הAPI אותו כתבתם.
 - a. עבור כל משאב:
 - i. השרת יקבל בקשה, כפי שמופיע בAPI
 - ii. השרת יחזיר תשובה, כפי שמופיע בAPI
 - b. במידה וברצונכם לשנות את המימוש, אין בעיה, כל זמן שאתם דואגים שמסמך הAPI תואם למימוש שלכם
7. שאלות על המטלה - ניתן לשאול בפורום הייעודי
8. המלצה חמה: חלקו את המימוש ביניכם, חלוקה טובה תהיה מימוש של routes שונים. ושתפו את הקוד בgit יחד. שמירה של הקוד בgit לאחר כל חלק שאתם מפתחים, תבטיח שלא תקרנה טעויות ותאבדו/תהרסו קוד עובד.
סרטון עזרה למי שלא מרגיש בנוח: https://www.youtube.com/watch?v=_OZVJpLHUaI

הגשה:

1. העבודה תוגש בGIT
<https://classroom.github.com/g/fOWgBwq->
2. שימוש לב, כי גם בעבודה זו כמו בכל הפרויקט, הקוד ייבדק במערכת ההענקות
3. תאריך הגשה: **7.6 בחצות**

בהצלחה!!