



## Spring REST API & DB

בנה בסיס נתונים לטובת בית הספר

טבלת תלמידים

Field	--
Id	מפתח auto increment
LastName	Text
FirstName	Text
AvgGrade	Float ציון ממוצע שנתי
Gender	Enum: Male, Female
Class_id	מפתח ForeignKey



טבלת כיתות

Field	--
Id	מפתח auto increment
NumberOfStudents	Integer מספר התלמידים בכיתה
ClassAvg	Float ציון ממוצע כיתתי
ExternalClass	Enum External, Regular כיתה רגילה או אקסטרנית

1. בנה מחלקת Student (לא לשכוח בנאי ו- toString), וגם מחלקה ל enum
2. בנה מחלקת Classroom (לא לשכוח בנאי ו- toString), וגם מחלקה ל enum
3. בנה מחלקות Mapper ל- 2 המחלקות
4. בנה מחלקת Repo לתלמידים. המאפשר הוספת תלמיד, מחיקה, שינוי, הבאת רשימת כל התלמידים, הבאת תלמיד לפי תעודת זהות. (ה- Repo רק עושה פעולות ולא תהיה בתוכו לוגיקה)
5. בנה מחלקת Repo עבור כיתת התלמידים. המאפשר הוספת כיתה, מחיקה, שינוי, הבאת רשימת כל הכיתות, הבאת כיתה לפי תעודת זהות. (ה- Repo רק עושה פעולות ולא תהיה בתוכו לוגיקה)
6. בנה Service עבור התלמידים. ה Service יפעיל את הפונקציות ב- Repo. שים לב שכל הוספת תלמיד (וגם שינוי ומחיקה) עשוי להשפיע על מספר התלמידים בכיתה + וגם על ציון הממוצע הכיתתי. לכן עליך לעדכן את מספר התלמידים ו/או הממוצע בכל שינוי ב- service שים לב שבכיתה אקסטרנית, לא יהיו יותר מ- 20 תלמידים (שמור את הערך 20 בקונפיג yml)

7. בנה Service עבור כיתות התלמידים. שים לב שאי אפשר למחוק כיתה אשר רשומים בה תלמידים .  
אם יהיה נסיון שכזה- זרוק Exception

8. נא להקפיד שה- Repo שכתבת, וגם ה- services שכתבת יממשו interface מסודר,  
לדוגמא IStudentRepo, IClassRoomRepo, IStudentService, IClassRoomService

9. בנה StudentController: המאפשר הוספת תלמיד, מחיקה, שינוי, הבאת רשימת כל התלמידים ,  
הבאת תלמיד לפי תעודת זהות, \*אתגר: הבאת רשימת כל התלמידים השייכים לכיתה אקסטרנית  
ה Controller יפעיל את ה service ויחזיר תשובה מסוג ResponseEntity עם סטטוס מתאים  
(לדוגמא 201, 404, 500 וכו'). שים לב שב controller לא תהיה לוגיקה , מעבר להפעלת ה service  
והחזרת התשובה הנכונה (+סטטוס מתאים)

10. בנה ClassroomController: המאפשר הוספת כיתה, מחיקה, שינוי, הבאת רשימת כל הכיתות ,  
הבאת כיתה לפי תעודת זהות, \*אתגר: הבאת רשימת כל התלמידים השייכים לכיתה אקסטרנית  
ה Controller יפעיל את ה service ויחזיר תשובה מסוג ResponseEntity עם סטטוס מתאים  
(לדוגמא 201, 404, 500 וכו'). שים לב שב controller לא תהיה לוגיקה , מעבר להפעלת ה service  
והחזרת התשובה הנכונה (+סטטוס מתאים)

11. ייצר מחלקת DTO המכיל כיתה ורשימת תלמידים.

ייצר controller נוסף לבחירתך המביא כיתה (לפי תעודת זהות) ואת כל התלמידים באותה הכיתה  
השתמש ב service קיים, או צור אחד חדש (כנ"ל לגבי ה repo)



12. ייצר FeignClient הפונה לאתר וובי ומביא פרטי תלמיד אקראי (לדוגמא:

<https://randomuser.me/api>).

הפונקציה של ה FeignClient תהיה חלק מה- controller שכתבת בסעיף הקודם וה url שלה יהיה  
לדוגמא: '/create\_random\_student'

תאור הפונקציה: הפונקציה תפעיל את ה- REST API (לדוגמא- random users) ומשם תיקח את  
פרטי התלמיד (שם פרטי, משפחה, זכר/נקבה, הגרל ציון מ- 40-100 עבור הממוצע שלו, הגרל כיתה  
עבורו (בטווח הכיתות הקיים)) **ולאחר מכן** השתמש בפרטים אלה כדי לייצר תלמיד חדש בדאטא  
בייס (תוך קריאה לפונקציה הוספת תלמיד שכתבת קודם לכן).  
הפונקציה תחזיר את פרטי התלמיד החדש שנוצר

\*\*\*אתגר מטורף: חשב את ממוצע הכיתה באמצעות design pattern composite

\* \*\*אתגר סופר מטורף: חשב את ממוצע בית הספר באמצעות design pattern composite

