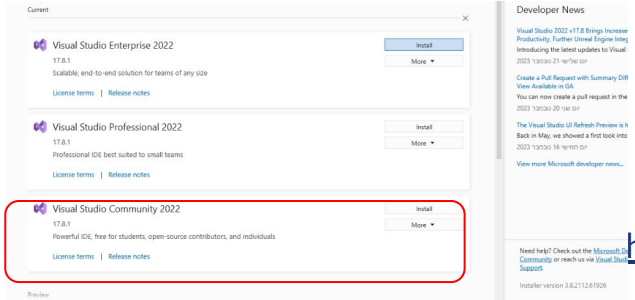
הנחיות לתבנית פרויקט

בדיקת גרסאות והוראות התקנה



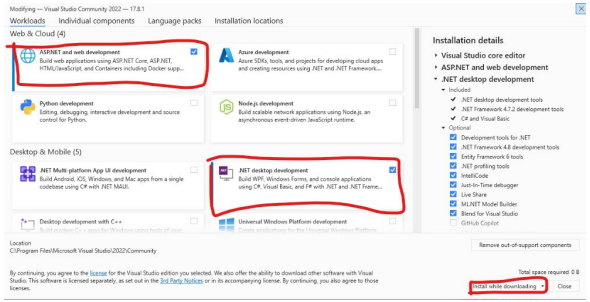
Visual Studio

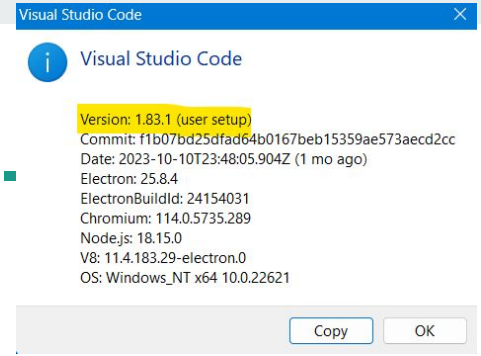
הגרסה המומלצת לעבוד עליה היא Visual Studio 2022

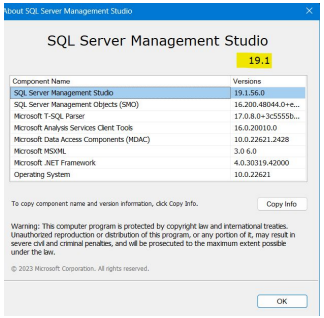
כדי לעדכן גרסה יש לפתוח את Visual Studio installer (לחפש בהתחל)

או להוריד מהקישור הנ"ל: https://visualstudio.microsoft.com/downloads

יש לבחור את הנ"ל:







Visual Studio Code

הגרסה העדכנית היא 1.84
את הגרסה המותקנת ניתן לבדוק בתגית help בתוכנה -> about
שמנה לב שהגרסה אצלכם במחשב לא רחוקה משם :)
קישור להורדה: https://code.visualstudio.com/

SSMS

הגרסה הנוכחית היא 19, ניתן לראות את מס' הגרסה ב help -> about
להורדה :
https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16

GIT

הפקודה הבאה בטרמינל בודקת אם גיט מותקנת על המחשב:

```
git --version
```

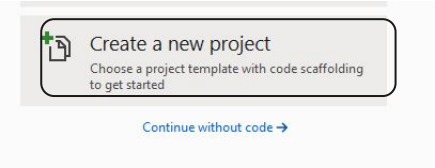
אם אכן מותקן גיט - יופיע מס' הגרסה

```
C:\Users\Sara Lu>git --version
git version 2.41.0.windows.1
```

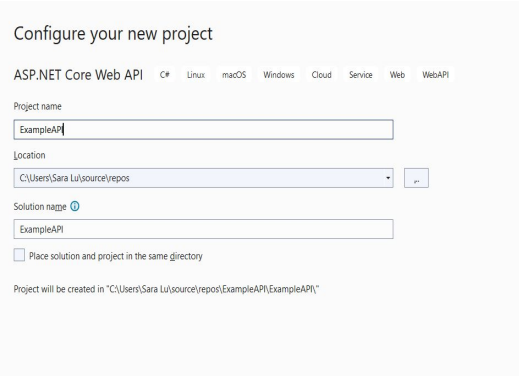
במקרה שלא יש להוריד מהקישור הבא ולהפעיל את קובץ ההתקנה https://git-scm.com/downloads

יצירת פרויקט WEB API

איך מתחילים?



נותנים שם בעל משמעות:



בוחרים סוג:

בוחרים גרסה: .net 8/7

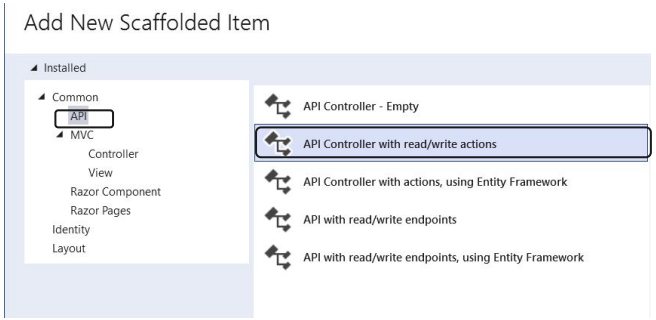
בפרויקט שנוצר יש שני קבצים לצורך הדגמה, ניתן למחוק אותם.

WeatherForecastController.cs

WeatherForecast.cs

שלב ראשון - יצירת Controller:

לחצן ימני על התיקיה Controllers -> Add



ללחוץ "add" ולתת שם ל-controller לפי שם הישות שהיא מטפלת בה, לדוג' ProductsController / UsersController

כעת נוצר קובץ עם ארבעת המתודות של http, (get, post, put and delete) ניתן להתבסס ולערוך אותם לפי הצורך.

הוספת השכבות

עבור כל שכבה- לחצן ימני על ה solution:
Add-> New Project-> Class Library->Next

וליצור פרויקט מסוג class Library עבור ה-Dal, BL , Entities

Dependency Injection

interface

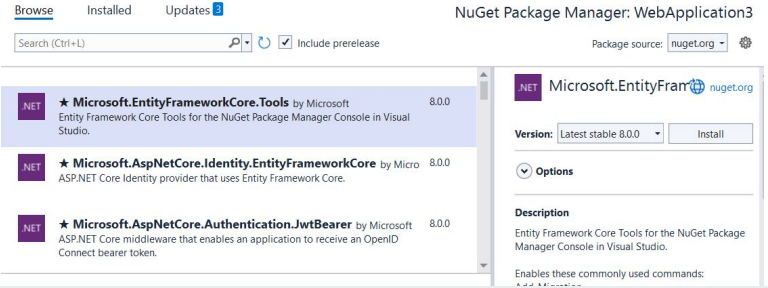
התחברות ל DB

לאחר שמסד הנתונים הוגדר כראוי, לכל הטבלאות יש Identity, ולא שכחנו שום קשר גומלין או שדה.
(תמיד אפשר להוסיף אח"כ, אבל זה פחות נחמד), צריך לחבר אותו לאפליקציה

1. ראשית יש להתקין את החבילות הבאות:

```
Install-Package Microsoft.EntityFrameworkCore.SqlServer
Install-Package Microsoft.EntityFrameworkCore.Tools
```

- ע"י הרצת הפקודות הנ"ל ב- **NuGet Package Manager** (Tools -> NuGet Package Manager -> package manager console)
- **התקנה דרך Nuget Package Manager:** לחצן ימני על הפרויקט -> manage Nuget Package ושם לעשות brows לחבילות הנ"ל ו-install



יש לודא כי מתקינים את החבילות הנ"ל על השכבה האחראית על החיבור ל DB (DAL / Repository)

Connection String

איך מוצאים?

ב- Visual Studio בוחרים בתפריט: Tools-> Connect To DataBase-> Microsoft SQL Server

מכניסים את שם שרת ה-SQL Server, שם ה -DB

אם מגיעה הודעה "The connection succeeded" אפשר לעבור לשלב הבא.

שימי לב כי הפקודה לא תרוץ כל עוד יש שגיאות בפרויקט, מומלץ לעשות rebuild (לחצן ימני על ה solution) ולוודא כי

הרצת הפקודה scaffold-dbcontext

ב package maneger console מריצים את הפקודה הבאה:

```
Scaffold-DbContext "Server=your-server-name;Database=your-DB-Name;Trusted_Connection=True"
TrustServerCertificate=true"Microsoft.EntityFrameworkCore.SqlServer -OutputDir Models
```

את הקבצים נמצא בתיקיית ה-Models. יש להעביר אותם ל-Entities, ואת ה- DBContext ל-DL

קובץ Configuration:

הגדרות הנפרדות מהקוד כמו חיבורים למסד הנתונים, כתובות מייל ומידע כללי נוסף כנ"ל יש לשמור בקובץ קונפיגורציה כך שנוכל לגשת אליהם ולשנות ולעדכן אותם בקלות.
ב.Net קובץ הקונפיגורציה נקרא appsetings.json, מומלץ להעתיק לשם את ה connection string בצורה הבאה:

```
  },
  "AllowedHosts": "*",
  "ConnectionStrings": {
    "School": "Data Source=SRV2\\PUPILS;Initial Catalog=SalesWebsite;Integrated Security=True;Trusted_Connection=True;",
    "Home": "Data Source=DESKTOP-QBHR7E5\\MSSQLSERVER01;Initial Catalog=ShoppingWebsite;Integrated Security=True;Trusted_Connection=True;"
  }
}
```

דוגמא לפונקציה get user ב controller, service ו repository

controller:

```
    [HttpGet]
    0 references
    public async Task<ActionResult<UsersHeaderDTO>> GetUserHeader(int id)
    {
        User user = await userService.getUserHeader(id);
        if (user == null)
        {
            return NoContent();
        }
        UsersHeaderDTO usersHeader = mapper.Map<User, UsersHeaderDTO>(user);

        return usersHeader;
    }
}
```

service:

```
2 references
public async Task<User?> getUserHeader(int id)
{
    User? user = await userRepository.getUserHeader(id);
    return user;
}
```

repository

```
2 references
public async Task<User?> getUserHeader(int userId)
{
    User? user = usersList.FirstOrDefault(u => u.UserId == userId);
    return user;
}
```

פעולות בסיסיות ב GIT:

שם הפקודה:	הסבר:	הפקודה:
commit	יצירת commit חדש המציין את השינויים שבוצעו.	"git commit -m "test commit
pull	משיכת השינוים האחרונים מה repository המרוחק.	git pull origin master
push	דחיפת השינויים האחרוניים (שבוצעו עליהם commit) לrepository המרוחק.	git push origin master