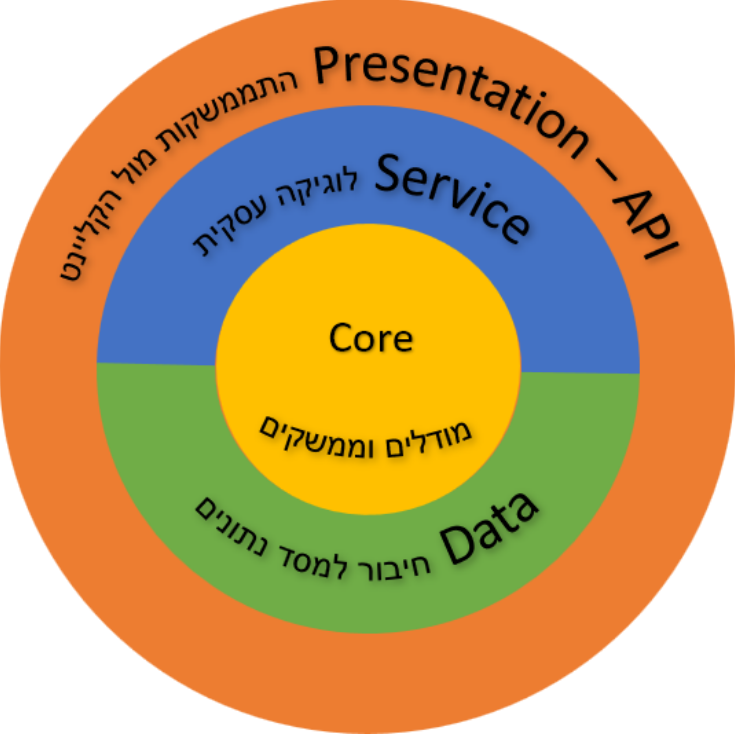
[Clean Architecture](https://learn.malkabruk.co.il/web/net/net-07-clean-architecture/):

פרויקט API מחולק ל 3 שכבות תלויות- כל שכבה תלויה בשכבות היותר פנימיות.



-שכבות הService וה- Data אינן תלויות זו בזו.

בשכבת הData זה כל החיבור לdata base – מבחינת הקוד- DataContext .

בשכבת הAPI יש את ההתממשקות מול הclient- - כולל controllers- פונקציות Get ,Post ,Put Delete. שכבה זו לא פונה ישירות לשכבת הData – אלא עוברת דרך שכבת הService.

בשכבת הService יוצרים לכל מחלקה שנמצאת בתיקיית Entities שבשכבת הAPI ClassService בClass הזה כותבים את פונקציות השרות של המחלקה- Get ,Post וכו'.

כיון שהשאיפה שלנו היא לנתק תלויות ככל האפשר- ClassService לא פונה ישירות ל DataContext ולכן ClassService פונה למחלקה שעוטפת את הפונקציות של ביצוע הפעולות על הClass .

כל Class מחולקת ל-2 חלקים: 1. מודל (Entity ) – החזקת הנתונים (props).

2. אוסף (repository) - מחלקה שאחראית על הפונקציונאליות שאפשר לעשות על הנתונים. (הוספה, שליפה וכו').

ה ClassRepository נמצאת ב Data -מכירה את DataContext ופונה אליה ישירות (שהרי הן באותה שכבה- לא יצרנו תלות).

כיון שהשאיפה שלנו היא לנתק תלויות ככל האפשר- בשכבת ה Core יוצרים IClassRepositoy שיגדיר את הפונקציות של ClassRepositoy

וב ClassService נצרוך את IClassRepository

למרות שלשכבת ה API יש תלות בשכבת ה- Service ולכן, לכאורה, אפשר בController להשתמש מיד בClassService עדיין השאיפה שלנו היא לנתק תלויות ככל האפשר- שהתלות תהייה רק בשלב ההזרקה.

לכו נשתמש בממשק.

ניצור בשכבת הCore IClassService וב Controller נשתמש בו ולא בClassService .

והזרקות התלויות- בProgram.

פרויקט דוטנט  <https://github.com/Shira2299/MyProject-.Net>

Node <https://github.com/Shira2299/TinyUrl_Backend>

שכבת Models שזה הישויות לפי מספר הטבלאות את יוצרת מחלקות וכמובן מגדירה שם את המשתנים הקיימים

שכבת Routers ששם קיימים לך כל ה-middleware הקיימים בפרויקט כמובן לפי כל מחלקה את פותחת לו את הראוט שלו (כמו בדוטנט)

שכבת Controller שזה בעצם הלוגיקה BL בדוטנט שהיא עושה קריאות לשכבת ה-Dal שבמקרה שלנו פה זה שכבת ה-Contex

שכבת Contex קריאות למסד הנתונים כל הפעולות הקיימות כמו ה-Dal בדוטנט

זה בגדול השכבות

יש לך גם דף נוסף db.js יותר מקובל לכתוב אותו בעמוד נפרד שהוא מתחבר למסד נתונים בדכ מונגו הולך איתו (זה ברעיון של connection string)

ויש לך את הדף הראשי index.js/app.js זה השמות המקובלים בדכ שהוא בעצם מנתב לך את הפרויקט עושה קריאות לשכבת ה-Routers

שהם עושים קריאות לשכבה שמעליהם ה-Controller והוא למסד הנתונים ה-contex

זה בגדול מה שאפשר לכתוב שזה יהיה מובן כמובן יש לך ממש מקביל כמו בדוטנט את הבעיה של חיבור לקליינט אז מוסיפים cors וכו'...

ועוד דבר במקום using יש לך import שימי לב שחיבור בין דפים צריך לייבא וחובה עם סיומת js