Node JS

Node.js: סביבת הרצה של JS מחוץ לדפדפן.

אין ב Node.js אבייקט window, אין אובייקט Document.... להשלים

אפשר לגשת ל.... להשלים

REST API

זהו שרת המחצין מידע.

REST: Representational State Transfer – שרת שמנגיש מידע מ-DB ומאפשר לבצע:

* קריאת המידע
* הוספת מידע
* עדכון מידע

Package.json

קובץ הגדרות עבור פרויקט Node.js

יצירה: npm init -y

Ts-node:

סיפרייה המותקנת גלובלית במחשב, המאפשרת להריץ פרויקט TypeScript Node.js.

Hot-Reloading:

זהו מצב בו כל שינוי של הקוד, מריץ מחדש את התוכנית.

Nodemon:

קיצר של Node monitor, המאפשרת להריץ את פרוייקט ה Node.js כ- Hot-Reloading.

התקנה עבור פרויקט TypeScrupt על Node:

npm I @types/node -D

**Express**

ספרייה המאפשרת לבנות שרת REST API ב- Node.js

npm i express

npm I @types/express -D

**פרוייקט מבוסס שכבות – Layers**

שכבת ה Frontend:

שכבת ה Controllers – Controllers layer:

שכבה המאזינה לבקשות הרשת.

שכבת ה Business Logic - Business Logic Layer:

שכבת ה services המכילים את הלוגיקה.

שכבת DAL – Data Access Layer:

השכבה הניגשת ל DB.

שכבת Data – Data layer:

השכבה המכילה את המידע, DB.

**Models**

מחלקות מידע טהור.

לדוגמא, מחלקה המתארת מוצר, לקוח, ספק.

**Middleware**

זו פונקציה המתבצעת אוטומטית בכל גלישה, בין ה Request ל Response.

AppConfig

מחלקה המכילה את נתוני הקונפיגורציה של כלל הפרויקט.

בקובץ TypeScript יש אך ורק OOP Classes.

או שאנו חושפים את ה Class מחוץ לקובץ או שאנו חושפים אובייקט מה Class.

אם דרוש אך ורק אובייקט אחד מה Class, יש לייצר אותו בתוך הקובץ (אחרי ה Class) ולהחצין אותו.

אם דרושים הרבה אובייקטים מה Class, אנו לא יוצרים את האובייקט בתוך הקובץ אלה מחצינים החוצה.

**Services**

מחלקה המכילה פעילות לוגית.

במקרים רבים מכילה את שאילתות ה-SQL ששולחים ל-DB.

Controller

אותו חלק שמאזין לבקשות מה-Front.