

הסבר על המבנה הארכיטקטורי של האתר שלי ב- server side

1. Routes-נתיבים:

בשכבת הנתיבים, הגדרתי את הנתיבים הקיימים באפליקציה שלי ואילו פעולות יש לבצע כאשר מתקבלת בקשה לכל נתיב. לרוב, כל נתיב מקושר לפונקציה או פעולה מתאימה בשכבת ה-Controller. זה מאפשר ניתוב תקשורת בין הפונקציות השונות בשרת.

2. Schema/Model-סכמה/מודל:

בשכבת סכמה או מודל, הגדרתי את מבנה הנתונים של האפליקציה שלי. זו כוללת את תכונות הנתונים ואת רוב בדיקות הולידציה והתקינות שלהן.

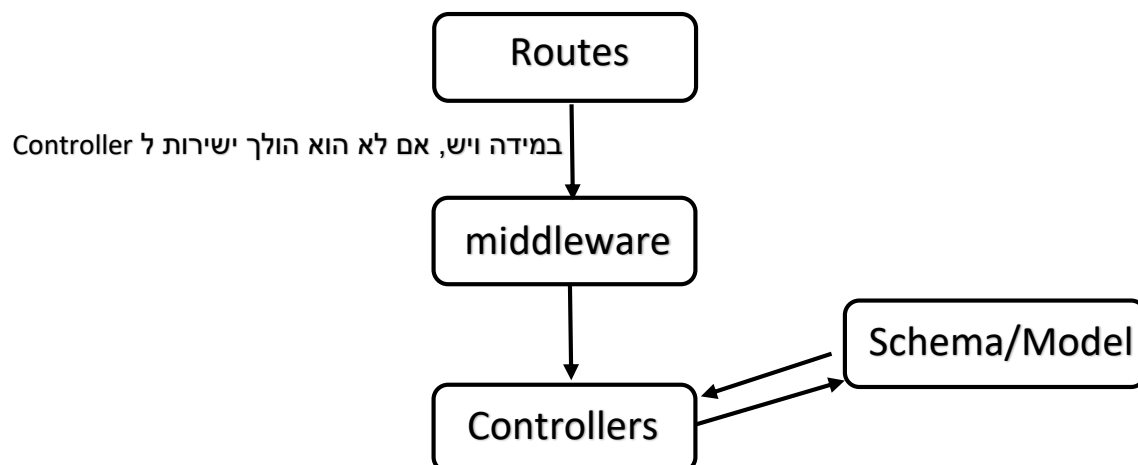
3. Controllers-בקרים:

בשכבת בקרים, הגדרתי את הפונקציות או הפעולות שמתבצעות כאשר מתקבלת בקשה מהלקוח. הבקרים הם האחראים לתגובה לבקשות שמגיעות מהלקוח, והם פעילים במרחב הדינמי של הפעילות הכללית של האתר. הם יכולים לפנות לשכבת ה-Model כדי לקבל או לעדכן נתונים ולספק את התוצאה הרצויה ללקוח.

4. Middleware-שכבת הביניים:

שכבת ה-Middleware עוסקת בעיבוד ובתגובה לבקשות בכלל, לפני שהן מגיעות לבקרים. זה יכול לכלול פעולות כמו אימות, תקשורת עם כלים חיצוניים (כמו אסינכרוניות), עיבוד של נתונים שהגיעו בבקשה ועוד.

השימוש במבנה זה מאפשר הפרדה בין הלוגיקה העסקית של האפליקציה (בבקרים), הלוגיקה המתמקדת בניהול התקשורת והעיבוד המרכזי של הבקשות (בשכבת ה-Middleware) וההגדרה של הנתונים ופעולות עליהם (בסכמה/מודל). זה מאפשר קוד נקי, ניתוח ותחזוקה קלים, והבנה טובה יותר של כל חלק בנפרד.



הטבלאות שלי:

patients		
Name field		type
id		String
FirstName		String
lastName		String
Address	{city	Object{String
	street	String
	BuildingNumber}	Number}
birthDate		Date
tel		String
mobilePhone		String
image		String(client side-File)

Covid19Details	
Name field	type
patientId	String
vaccineDates	array
positiveTestDate	Date
recoveryDate	Date