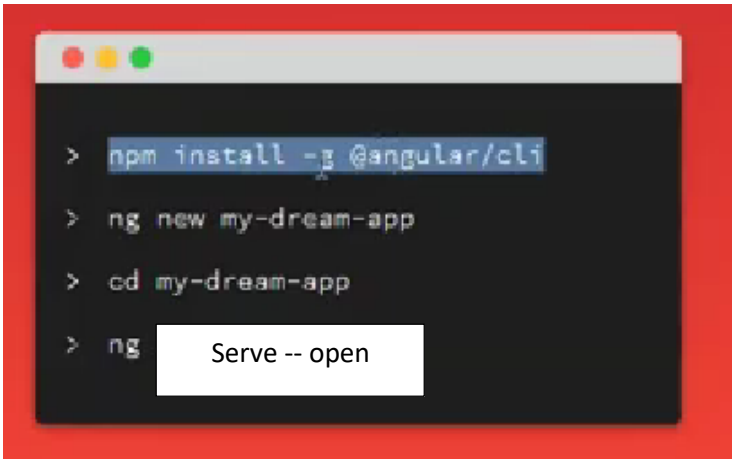

Microsoft
SQL Server


.NET
Core

אתחול פרויקט Ang



package.json

sample-app > package.json > name

```
{
  "name": "sample-app",
  "version": "0.0.0",
  ▷ Debug
  "scripts": {
    "ng": "ng",
    "start": "ng serve",
    "build": "ng build",
    "test": "ng test",
    "lint": "ng lint",
    "e2e": "ng e2e"
  },
  "private": true,
  "dependencies": {
    "@angular/animations": "~10.2.0",
    "@angular/common": "~10.2.0",
    "@angular/compiler": "~10.2.0",
    "@angular/core": "~10.2.0",
    "@angular/forms": "~10.2.0",
    "@angular/platform-browser": "~10.2.0",
    "@angular/platform-browser-dynamic": "~10.2.0",
    "@angular/router": "~10.2.0",
    "rxjs": "~6.6.0",
    "tslib": "^2.0.0",
    "zone.js": "~0.10.2"
  },
  "devDependencies": {
    "@angular-devkit/build-angular": "~0.1002.0",
    "@angular/cli": "~10.2.0",
    "@angular/compiler-cli": "~10.2.0",
    "@types/node": "^12.11.1",
    "@types/jasmine": "~3.5.0",
    "@types/jasminewd2": "~2.0.3",
    "codelyzer": "^6.0.0",
    "jasmine-core": "~3.6.0",
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

הגדרות תוספים ופקודות לתוכנה

הגדרות כלליות של ng

```
app.component.html
src > app > app.component.html > ...
1  <h1>Hello World from {{ title }}</h1>
2
3  <div class="directives" *ngIf="listOfMovies.length > 6">
4    {{ mainChar }}
5  </div>
6
7  <div
8    class="list-of-movies"
9    [ngStyle]="{ background: shouldBeRed ? 'red' : 'blue' }"
10 >
11   <div class="movie" *ngFor="let movie of listOfMovies">{{ movie }}</div>
12 </div>
13
14 <button (click)="shouldBeRed = !shouldBeRed">Change</button>
15
```

```
app.component.html  app.component.scss  app.component.ts  index
app > app.component.ts > AppComponent > movie
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-root',
5    templateUrl: './app.component.html',
6    styleUrls: ['./app.component.scss'],
7  })
8  export class AppComponent {
9    title = 'sampleApp';
10
11   listOfMovies = ['Avangers 1', 'Ironman 2', 'Spiderman'];
12
13   mainChar = 'Tony Stark';
14
15   shouldBeRed: boolean = true;
16
17   movie = {
18     movieName: 'Avangers 2',
19     rating: 9,
20   };
21 }
22
```

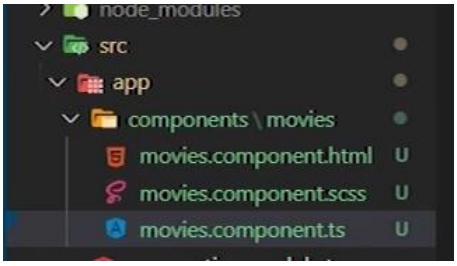
הוספת תכונות למחלקה

```
@Input() myLists !: string[];
myList2 = [
  [1,2,3],
  ['a','b','c']
];
```

```
You, 8 seconds ago • Uncommitted change
getStyle()
{
  if(this.isNewStyle)
  {
    return{fontSize:'30px',color:'blue'}
  }

  else{
    return{fontSize:'40px',color:'green'}
  }
}
```

```
<p [ngStyle]="getStyle()">
  my Spe
</p>

<button (click)="isNewStyle = !isNewStyle">change style</button>
</div>
```

הוספת קומפננטה - components



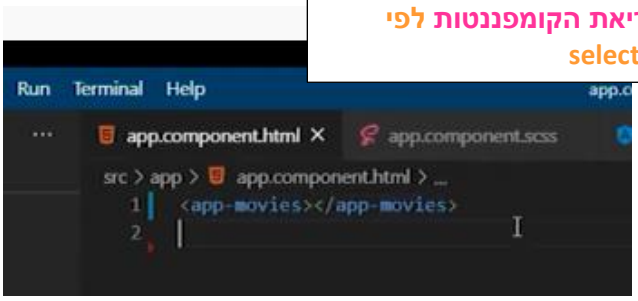
app > components > movies > movies.component.ts > ...

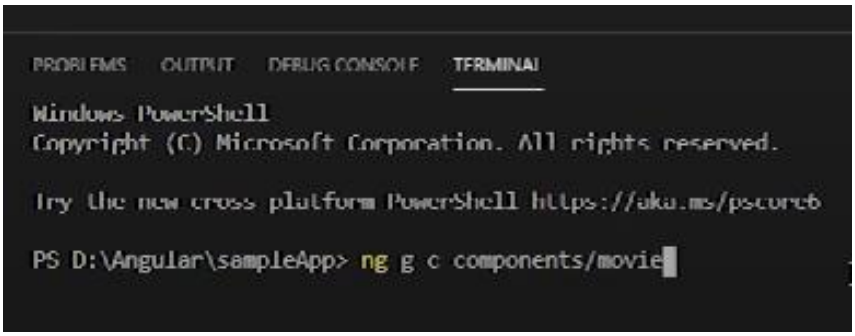
```ts
import { Component } from '@angular/core';

@Component({
  selector: 'app-movies',
  templateUrl: './movies.component.html',
  styleUrls: ['./movies.component.scss'],
})
export class MoviesComponent {
  title = 'Movie Component Works!!';
}
```

```ts
@NgModule({
  declarations: [AppComponent, MoviesComponent],
  imports: [BrowserModule, AppRoutingModule],
  providers: [],
  bootstrap: [AppComponent],
})
export class AppModule {}
```

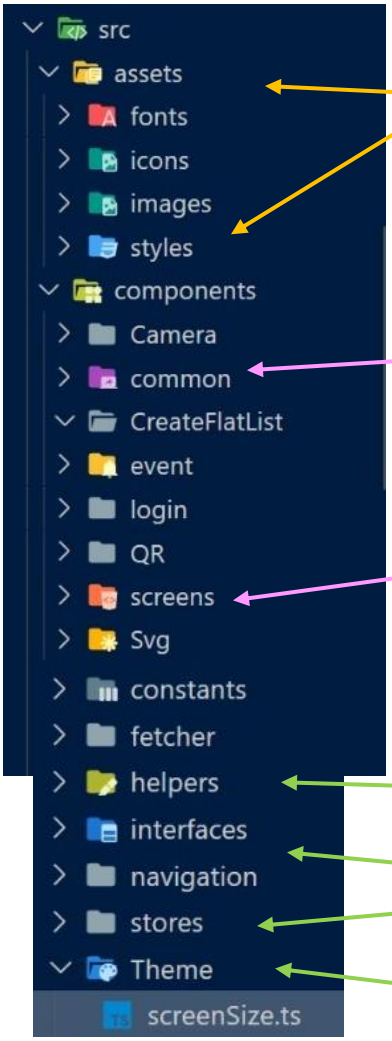
הוספת הקומפננטות

קריאת הקומפננטות לפי selector





העיצוב המודולרי



מאגר המידע בפרויקט

במקרה של צבעים למשל כדי לא לשנות את כל תגיות הצבע במקרה וירצו לשנות צבע מרכזי למשל כותרת , **עיצוב מרכזי**

רכיבי האתר

בכל עמוד

העמוד עצמו

פונקציות כלליות שנוכל לקרוא כל פעם שנצטרך ברכיב אחר

הגדרת צורות המחלקות

העיצוב המרכזי לרוב נגדיר בו את רכיבי אורך ורוחב המסך

```
tScreen.tsx   screenSize.ts   EventOverview.tsx
eme > screenSize.ts > ...
You, 21 seconds ago | 2 authors (omer and others)
import {Dimensions } from "react-native";
const { width, height } = Dimensions.get("window");

const screenSize = {
width:Dimensions.get('window').width,
height:Dimensions.get('window').height
};
export default screenSize;
```

כל חלק מסך שקוראים אליו במסך **הגדול יכנס לתיקיה עצמה** למשל Event

דגש על סינטקס **I של interface**

```
export interface IEventButtonProps {
    text: string;
    color: string;
    isSmall?: boolean;
}
```

לייף סייקל

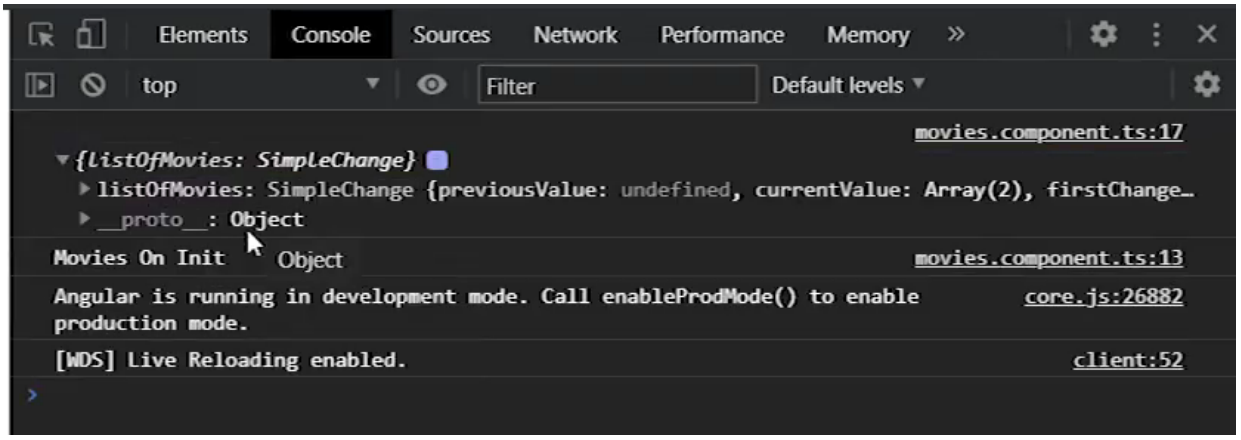
```
app > components > movies > movies.component.ts > MoviesComponent
import { Component, Input, OnInit, SimpleChanges } from '@angular/core';

@Component({
  selector: 'app-movies',
  templateUrl: './movies.component.html',
  styleUrls: ['./movies.component.scss'],
})
export class MoviesComponent implements OnInit {
  title = 'Movies Component Works!!';

  @Input() listOfMovies: string[];

  ngOnInit() {
    console.log('Movies On Init');
  }

  ngOnChanges(changes: SimpleChanges): void {
    console.log(changes);
  }
}
```

ngOninit -> לאחר שהקומפננטה נקראת ,מתרנדרת וכל המידע העובר אליה

ngOnChange -> נקראת ראשונה , מקבלת מידע על שינוי מגורם חיצוני

@input -> העברת , קבלת מידע לקופמננטה מהאב

```
ovies.component.ts   movie.component.ts ×   app.component.html ×   movies.component.html
app > app.component.html > app-movies
  <app-movies [listOfMovies]="['Ironman', 'Wonderwoman']"></app-movies>
```



Input עם set – החליף את **תפקידו לזהות שינוים בערך שקיבלנו מבחוץ**

```typescript
@Component({
  selector: 'app-movies',
  templateUrl: './movies.component.html',
  styleUrls: ['./movies.component.scss'],
})
export class MoviesComponent implements OnInit, OnChanges, AfterViewInit {
  title = 'Movies Component Works!!';

  @Input() listOfMovies: string[];

  ngOnInit() {
    console.log('Movies On Init');
  }

  ngOnChanges(changes: SimpleChanges): void {
    console.log(changes);
  }

  ngAfterViewInit() {
    console.log('After View Init');
  }
}
```

האזנת שינוים בפקדים

```typescript
  console.log(changes);
}

ngAfterViewInit() {
  console.log('After View Init');
}

ngOnDestroy(): void {
  //Called once, before t
  //Add 'implements OnDes
  console.log("OnDestory")
}
}
```

log(message?: any, ...optionalParams: any[]): void

1/2 Prints to stdout with newline.

השמדה

```
app > app.component.html > button
<app-movies *ngIf="shouldDisplayMovie" [listOfMovies]="['Ironman', 'Wonderwoman']"></app-movies>

<button (click)="toggleMovieComponent()" >Toggle Movie Component</button>
```

```typescript
toggleMovieComponent(){
  this.shouldDisplayMovie = !this.shouldDisplayMovie
}
}
```

```
entScreen.tsx   screenSize.ts ●   EventOverview.tsx
Theme > screenSize.ts > ...
You, 1 minute ago | 2 authors (omer and others)
import {Dimensions } from "react-native";
const { width, height } = Dimensions.get("window");

const screenSize = {
width:Dimensions.get('window').width,
height:Dimensions.get('window').height
};
export default screenSize;
```

```
const { height, width } = screenSize;
```

פריסת תכונות כדי שלא
נצטרך לרשום נתיב ארוך

Conditional (ternary) operator

אופרטור המשולש

האופרטור המותנה (טרינרי) הוא אופרטור JavaScript היחיד **שלוקח שלושה אופרנדים**: **תנאי ואחריו סימן שאלה (?),** לאחר מכן **ביטוי לביצוע אם התנאי אמת** ואחריו **נקודתיים (:),** ולבסוף **הביטוי ל לבצע אם התנאי שקרי**. אופרטור זה משמש לעתים קרובות **כחלופה להצהרת if...else.**

```
<div
[ngStyle]="{border: newIMovies.movieAge > 3 ? '3px dotted blue' : '3px dotted green'}"
>
<p>Movie Name : {{newIMovies.movieName}}</p>
<p>Movie Genre : {{newIMovies.movieGenre}}</p>
<p>Movie Date : {{newIMovies.movieDate.toLocaleDateString()}}</p>
<p>Movie Age : {{newIMovies.movieAge}}</p>
<p *ngFor="let actor of newIMovies.movieNameActor">Movie Name Actor : {{actor}}</p>
</div>
<button (click)="getDate()">Get my Date</button>
```

תצוגת תאריך בפורמט שלנו

Movie Name : Shrek 5

Movie Genre : Animation

Movie Date : 26.4.2022

Movie Age : 13

Movie Name Actor : Shrek

Movie Name Actor : Fiona

Movie Name Actor : Donkey

Movie Name Actor : Dragon

Movie Name Actor : Magical Fairy Dani

Get my Date | togggle

```
<div
[ngClass]="{'daniMov'  : newIMovies.movieAge > 3 ? '3px dotted blue' : '3px dotted green'}"
>
<p>Movie Name : {{newIMovies.movieName}}</p>
<p>Movie Genre : {{newIMovies.movieGenre}}</p>
<p>Movie Date : {{newIMovies.movieDate.toLocaleDateString()}}</p>
<p>Movie Age : {{newIMovies.movieAge}}</p>
<p *ngFor="let actor of newIMovies.movieNameActor">Movie Name Actor : {{actor}}</p>
</div>
```

שימוש ב ngClass

```
.roniMov {
  border: 1px double green;
  background-color: aqua;
}

.daniMov {
  border: 1px double blue;
  background-color: lightgreen;
  color: darkblue;
}
```


```
<div
[ngClass]="{daniMov:(newIMovies.movieAge > 3  && newIMovies.movieAge < 5 ),
            roniMov:(newIMovies.movieAge > 5  && newIMovies.movieAge < 15 )}"
>
<p>Movie Name : {{newIMovies.movieName}}</p>
<p>Movie Genre : {{newIMovies.movieGenre}}</p>
<p>Movie Date : {{newIMovies.movieDate.toLocaleDateString()}}</p>
<p>Movie Age : {{newIMovies.movieAge}}</p>
<p *ngFor="let actor of newIMovies.movieNameActor">Movie Name Actor : {{actor}}</p>
</div>
<button (click)="getDate()">Get my Date</button>
```

לשים לב שלא שמנו סוגריים

```
6   <div
7   [ngClass]="newIMovies.movieAge > 3 ? 'daniMov' : 'roniMov'"
8   >
9   <p>Movie Name : {{newIMovies.movieName}}</p>
0   <p>Movie Genre : {{newIMovies.movieGenre}}</p>
1   <p>Movie Date : {{newIMovies.movieDate.toLocaleDateString()}}</p>
2   <p>Movie Age : {{newIMovies.movieAge}}</p>
3   <p *ngFor="let actor of newIMovies.movieNameActor">Movie Name Actor : {{actor}}</p>
4   </div>
5   <button (click)="getDate()">Get my Date</button>
6
```

הוספת Pipe עיצוב לטקסט

```
<p>Movie Date : {{newIMovies.movieDate.toLocaleDateString()}}</p>
<p>Movie Date : {{newIMovies.movieDate | date}}</p>
```

```
<p>Movie Genre : {{newIMovies.movieGenre}}</p>
<p>Movie Date : {{newIMovies.movieDate.toLocaleDateString()}}</p>
<p>Movie Date : {{newIMovies.movieDate | date:'dd/MM/yyyy HH:mm:ss'}}</p>
```

26/04/2022 13:16:50

שליחה וקבלת אירוע

```
getEvent(e:Event)
{
  console.log(e.target!);
  this.newIMovies = {
    movieAge:3,
    movieDate:new Date(),
    movieGenre:'Fantasy',
    movieLoveAmos:true,
    movieName:'Sponge Bob',
    movieNameActor:[
      'Sponge Bob',
      'Patrick',
      'Squid',
      'Sandy',
      'Plankton',
      'Krabs'
    ]
  }
}
```

מאחר **ויתכן כי** האובייקט **null** נצהיר **שנקבל ערך בעזרת '!'**

מילה שמורה של אנגולר המצהירה על קבלת נתוני אירוע

```
<button (click)="getDat                    ton>
<button (click)="getEvent($event)" value="Change Movie">Get my Event</button>
```

Directives

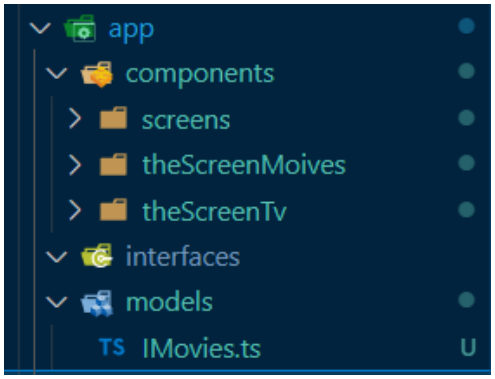
Directives allow us to extend the power of the template and HTML within it by providing new syntax

Structural Directives	Attribute Directives	Component Directives
ngFor	App-tooltip	header
ngSwitchCase	App-highlight	footer
ngIf		

change the DOM layout by adding and removing DOM elements.

change the appearance or behavior of an element, component, or another directive.

directives with a template.



Interface יימצאו בתיקיית modal

בדיקה אם קיים אובייקט סרט אחרת שגיאה

```
Go to component
<div
*ngIf="movieShow"
[ngClass]="(movieShow.movieAge > 3 && movieShow.movieAge < 13) ? 'daniMov' : 'roniMov'"
>
<p>Movie Name : {{movieShow.movieName}}</p>
<p>Movie Genre : {{movieShow.movieGenre}}</p>
<p>Movie Date : {{movieShow.movieDate.toLocaleDateString()}}</p>
<p>Movie Date : {{movieShow.movieDate | date:'dd/MM/yyyy HH:mm:ss'}}</p>
<p>Movie Age : {{movieShow.movieAge}}</p>
<p *ngFor="let actor of movieShow.movieNameActor">Movie Name Actor : {{actor}}</p>
</div>
```

```
Go to component
<p>movies works!</p>
<div *ngFor="let movie of moviesNow" [ngStyle]="{backgroundColor : movie.movieAge > 15 ? 'fuchsia'
                                                                 : movie.movieAge > 10 ? 'dodgerblue'
                                                                 :'aqua' }">
  <app-movie-now [movieShow] = movie></app-movie-now>
  <br>
</div>
<button (click)="getEvent($event)" value="Change Movie">Get my Event</button>
```

```
c.ts U | movies.component.html U | app.component.html M | movie-now.component.ts U | mov
rc > app > components > theScreenMoives > movie-now > movie-now.component.ts > ...
1   import { IMovies } from '../../../models/IMovies';
2   import { Component, Input, OnInit } from '@angular/core';
3
4   @Component({
5     selector: 'app-movie-now',
6     templateUrl: './movie-now.component.html',
7     styleUrls: ['./movie-now.component.css']
8   })
9   export class MovieNowComponent implements OnInit {
10
11    constructor() { }
12
13    ngOnInit(): void {
14    }
15
16    @Input() movieShow !: IMovies;
17
```

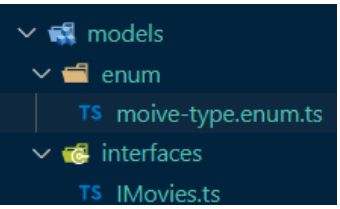
השם של **הערך אותו** **נשלח** צריך להיות זהה **לשם התכונה של Input**

```
p > components > movies > movies.component.html > div.movies > app-movie
<div class="movies">
  <app-movie *ngFor="let movie of listOfMovies; let i = index; let first = first; let
</div>
```

ריצה על אינדקסי הלולאה ועיצוב בהתאם

```
[ngStyle]="{'font-size':(30 - (i*5))+'px'}"
>Movie Name Actor : {{actor}}</p>
</div>
```

Enum + Interfaces



app > models > enum > moive-type.enum

```ts
export enum MovieType {
  Action = "Action",
  Drama = "Drama",
  Fantasy = "Fantasy",
  Comedy = "Comedy",
  Animation = "Animation"
}
```

app > models > interfaces > IMovies.ts > ...

```ts
import { MovieType } from "../enum/moive-type.enum";

export interface IMovies {
  movieName:string;
  movieDate:Date;
  movieGenre:MovieType;
  movieNameActor:string[];
  movieAge:number;
  movieLoveAmos:boolean;
}
```

```ts
moviesNow: IMovies[] = [
  {
    movieAge: 13,
    movieDate: new Date(),
    movieGenre: MovieType.Animation,
    movieLoveAmos: true,
    movieName: 'Shrek 5',
    movieNameActor: [
      'Shrek',
      'Fiona',
      'Donkey',
      'Dragon',
      'Magical Fairy Dani',
    ],
  },
```

שימוש ב ngSwitch

```ts
export class MovieNowComponent
```

נקבל את ערכי ה enum בקוד הקומפנטה **ולא בטמפלט** מאחר וזוהי הדרך הנכונה

```ts
get movieGenre () {
  return MovieType;
}
```

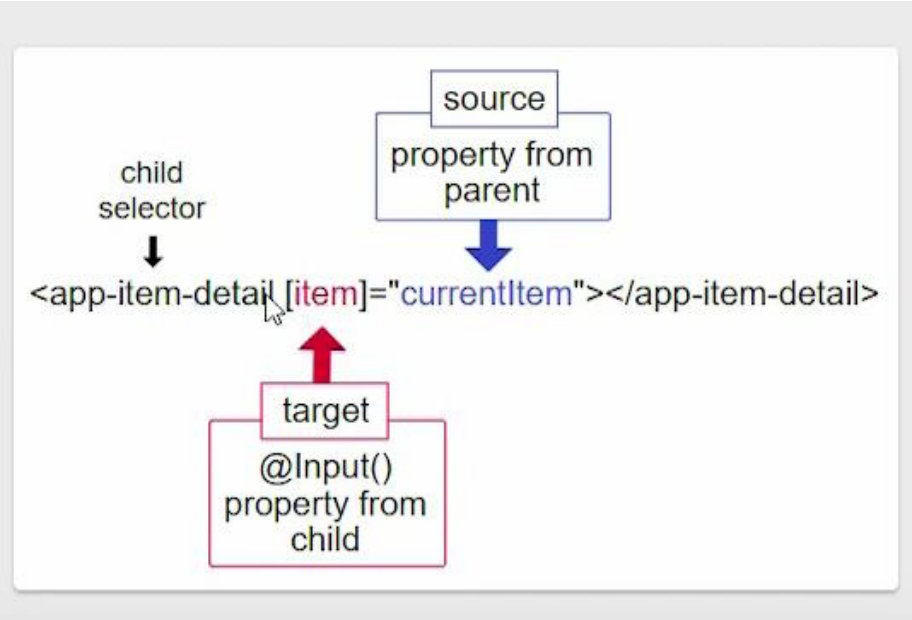
```html
<span [ngSwitch]="movieShow.movieGenre">
  <p *ngSwitchCase="movieGenre.Action" style="background-color: aqua; color: rgb(98, 250, 23
    Its Action
  </p>
  <p *ngSwitchCase="movieGenre.Animation" style="background-color: blue; color: rgb(184, 25,
    Its Animation
  </p>
  <p *ngSwitchCase="movieGenre.Comedy" style="background-color: rgb(0, 255, 64); color: rgb
    Its Comedy
  </p>
  <p *ngSwitchCase="movieGenre.Drama" style="background-color: rgb(255, 255, 0); color: rgb(
    Its Drama
  </p>

  <p *ngSwitchDefault style="background-color: aqua; color: bisque;">
    Its Fantasy
  </p>
```

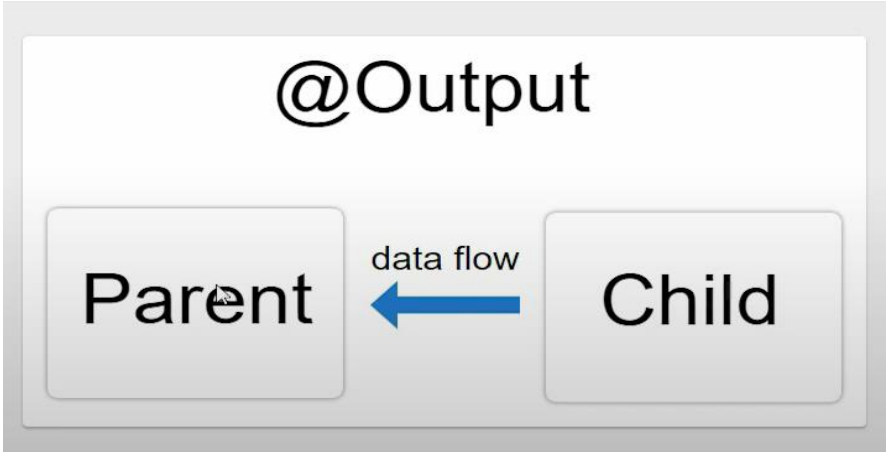
אנגולר 11 – שיפור ביצועים , אין **תמיכה ב3.9 של TS , שינוים בזמן אמת בלי צורך לרענן**

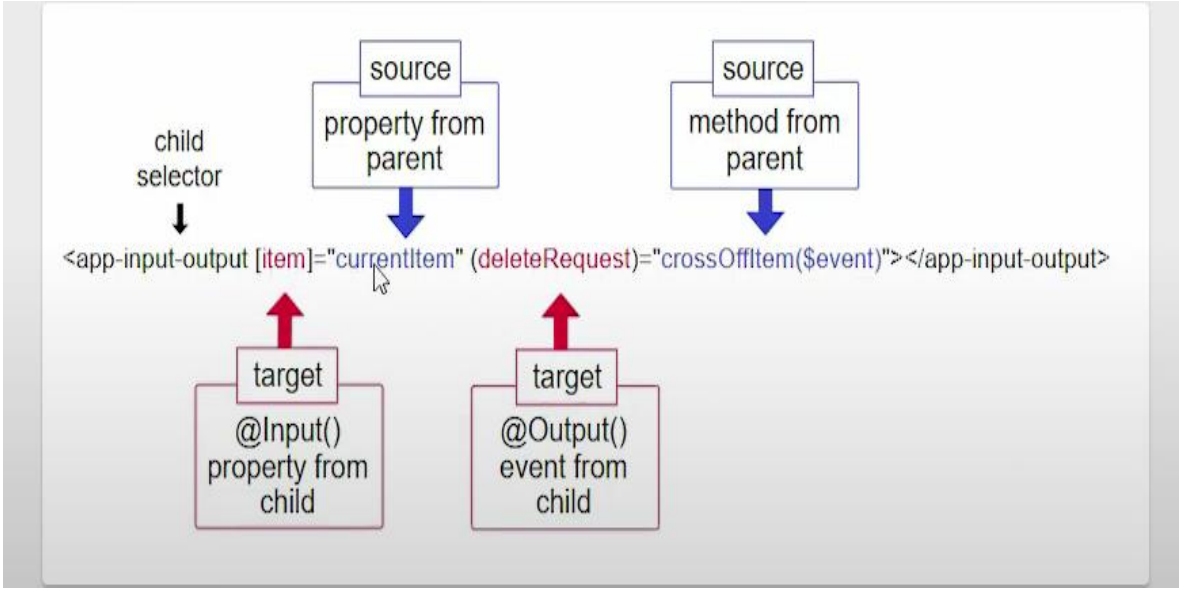
אנגולר 12 – **לא מיועד לפיתוח**

עדכונים

קלט ופלט


source
property from parent
child selector
<app-item-detail [item]="currentItem"></app-item-detail>
target
@Input() property from child


@Output
Parent
data flow
Child


source
property from parent
source
method from parent
child selector
<app-input-output [item]="currentItem" (deleteRequest)="crossOffItem($event)"></app-input-output>
target
@Input() property from child
target
@Output() event from child

קבלת ערך שהתקבל מאב
לעיצוב ללא גרשיים

```
<div
*ngIf="movieShow"
[ngStyle]="{border:'3px solid blue','border-width':myBorder}"
```

```
<ng-container *ngIf="condition">
    <li *ngFor="let item of items">
        {{item}}
    </li>
</ng-container>
```

שילוב תנאי עם לולאה כי אי
אפשר להציג יחד באותו
אלמנט

```
<ng-container *ngIf = "movieShow.movieName !='try'">
  <p
  *ngFor="let actor of movieShow.movieNameActor
            let i = index"
 [ngStyle]="{'font-size':(30 - (i*5))+'px'}"
 >Movie Name Actor : {{actor}}</p>
</ng-container>
</div>
```

```
<ng-container *ngFor="let actor of movieShow.movieNameActor
            let i = index">
  <p
  *ngIf="i%2 == 0"
  [ngStyle]="{'font-size':(30 - (i*5))+'px'}">
          Movie Name Actor : {{actor}}
 </p>
</ng-container>
```

```
@Output() deleteTheMovie : EventEmitter<number> = new EventEmitter<number>();


public deleteMyMovie()
{
  if(confirm('Sure baby ?'))
  {
    this.deleteTheMovie.emit(this.indexMovie);
    alert('ok see you soon');
  }
  else{
    alert('Yayy We saved !');
  }
}
```

כדי ליצור אירועים מותאמים אישית באופן סינכרוני או אסינכרוני, ולרשום מטפלים עבור אירועים אלה על ידי הרשמה למופע.

שיגור האירוע

חשוב לשים לב שבטמפלט נקרא לפונקציה ולא למייצר אירוע

```
  <button (click)="getDetailsWithIndex()">Get Details</button>
  <button (click)="deleteMyMovie()">Delete !</button>
</div>
```

תפיסת האירוע

```
tyle]="{backgroundColor : movie.movieAge > 15 ? 'fuchsia'
        : movie.movieAge > 10 ? 'dodgerblue'
        :'aqua' }"

[indexMovie]="i" (deleteTheMovie)="deleteMoviesFromArray($event)"></app-movie-now>


Get my Event</button>
```

קישור פונקציה באב לאירוע

```
  public deleteMoviesFromArray(indexMovie:number) : void
  {
    this.moviesNow.splice(indexMovie,1);
  }
}
```

כמה פרמטרים יחד

```
  <button (click)="getDetailsWithIndex()">Get Details</button>
  <button (click)="deleteMyMovie()">Delete !</button>
  <button (click)="setNameMovie('my Movie')">Set $</button>
</div>
```

ילד

```
@Output() setTheName = new EventEmitter<{index:number , name:string}>();

public setNameMovie(newName:string) : void
{
  this.setTheName.emit({index:this.indexMovie,name:newName});
}
```

```
this.addModel.emit({make: make, name: name});
//shorthand is below
this.addModel.emit({make, name});
```

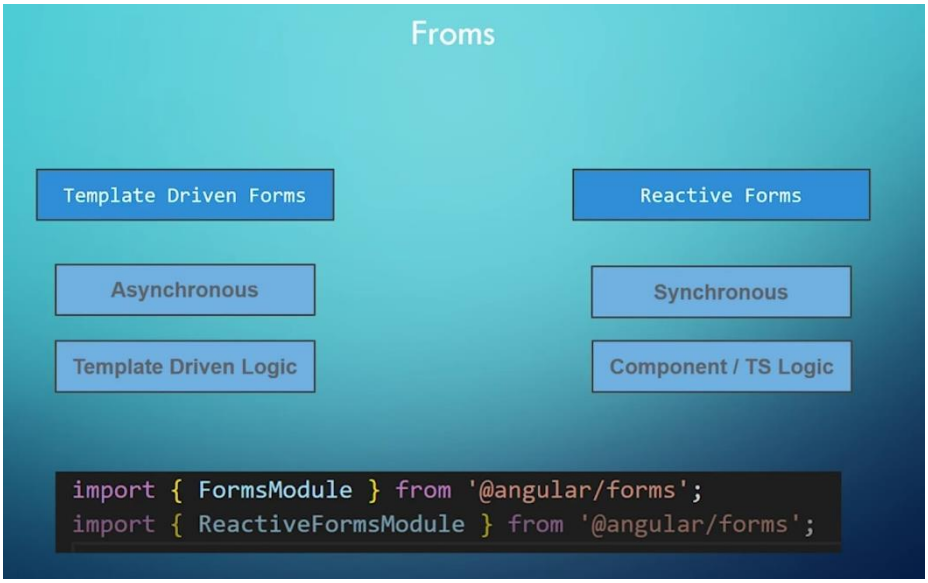
אבא

```
    (deleteTheMovie)="deleteMoviesFromArray($event)"
    (setTheName)="setMoviesFromArray($event)"></app-movie-now>
    <br>
  </div>
  <button (click)="getEvent($event)" value="Change Movie">Get my Event</button>
```

```
public setMoviesFromArray(detailsChange:any) : void{
  const {index,name} = detailsChange;
  this.moviesNow[index].movieName = name;
}
```

```
public setMoviesFromArray(detailsChange:{index: number,name:string}) : void{
  const {index,name} = detailsChange;
  this.moviesNow[index].movieName = name;
}
```

ולידציות



הגדרת **חוסר ברצון בבדיקת תקינות**

הגדרת **קבוצה בריבוע של פקדים**

```
<form novalidate>
  <fieldset>

    <div>
      <label>My First Name : </label>
      <input type="text">
    </div>

    <div>
      <label>My Last Name : </label>
      <input type="text">
    </div>

  </fieldset>

  <div>
    <label>Email : </label>
      <input type="text">
  </div>

  <div>
    <label>Password : </label>
      <input type="text">
  </div>

  <div>
    <label>Hobby : </label>
    <select>
      <option value="" disabled selected>Choose a Hobby :
      <option *ngFor="let hobby of hobbies"
                    value="hobby">{{hobby}}</option>
    </select>

  </div>


</form>
```

forms works!

My First Name :
My Last Name :

Email :
Password :
Hobby : Sing

הגדרת **select תחת שימוש חכם בלולאה**

```
hobbies : string[] = [
  "Dance",
  "Go out to eat",
  "Go to Escape Room",
  "Play on stage",
  "play on the computer",
  "Program",
  "See 'Our Song'",
  "Sing"
]
```

הגדרת **selected יעלה אוטומטי מסומן**

```
import { MovieNowComponent } from './components/theScre
import { FormsComponent } from './components/screens/fo
import { FirstComponent } from './components/theScreen

import { FormsModule } from '@angular/forms';

...
@NgModule({
  declarations: [
    AppComponent,
    TvComponent,
    MoviesComponent,
    DisneyComponent,
    SpongeBobComponent,
    MovieNowComponent,
    FormsComponent,
    FirstComponent,
  ],
  imports: [
    FormsModule,
    BrowserModule,
```

הגדרת import לכוח של FormsModule

יוצר מופע FormGroup ברמה העליונה ומקשר אותו לטופס כדי לעקוב אחר ערך טופס מצטבר וסטטוס אימות.

משתנה מקומי שסוגו פורם

ngModelGroup – יוצר ומקשר מופע FormGroup לרכיב

ngModel – יוצר מופע FormControl ממודל תחום ומקשר

```
<form novalidate #myForm = "ngForm">
  <fieldset ngModelGroup="myHeaderForm">
    <div>
      <label>My First Name : </label>
      <input ngModel name="firstName" type="text">
    </div>

    <div>
      <label>My Last Name : </label>
      <input  ngModel name="lastName" type="text">
    </div>
  </fieldset>

  <div>
    <label>Email : </label>
      <input ngModel name="ooEmail" type="text">
  </div>

  <div>
    <label>Password : </label>
      <input ngModel name="myPassword" type="text">
  </div>

  <div>
    <label>Hobby : </label>
    <select ngModel name="selectedHobby">
      <option value="" disabled selected>Choose a Hobby : </option>
      <option *ngFor="let hobby of hobbies"
                    value="hobby">{{hobby}}</option>
    </select>
  </div>

</form>

<pre>
  {{myForm.value | json}}
</pre>
```

forms works!

My First Name :
My Last Name :

Email :
Password :
Hobby : Choose a Hobby :

```
{
"myHeaderForm": {
  "firstName": "",
  "lastName": ""
},
"ooEmail": "",
"myPassword": "",
"selectedHobby": ""
```

pre – מוצג בגופן ברוחב קבוע, והטקסט שומר גם על רווחים וגם מעברי שורות. הטקסט יוצג בדיוק כפי שנכתב בקוד המקור של c.HTML

ViewChild – ההאזנה לשינוי הערכים של פקדים

```
@ViewChild('myForm') myFormChanges : any;
```

מדובר ב form ולכן יכול להיות any (תוכנו משתנה כל פעם)

ערכי ה form

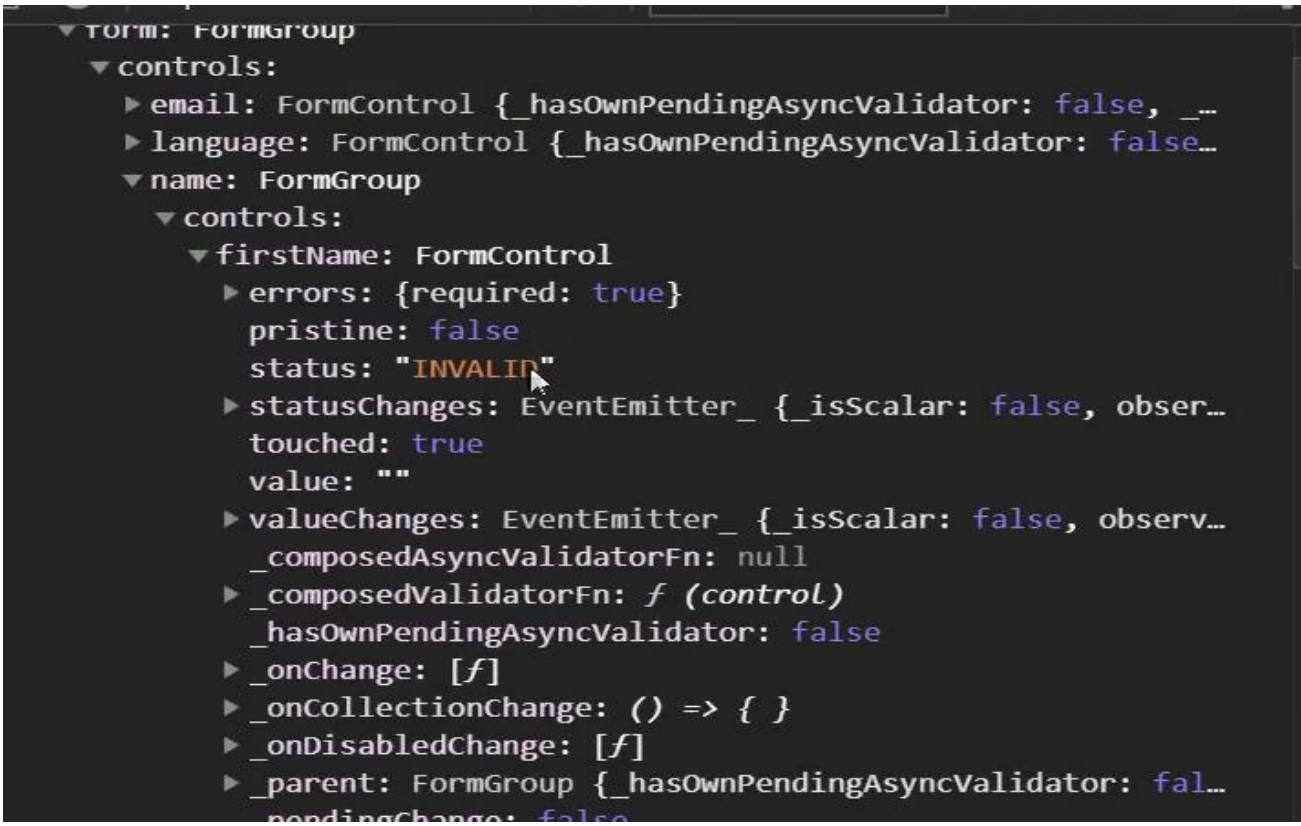
```
validator: (...)
value: Object
  email: "yuda@buzah.com"
  language: "Hebrew"
  name: {firstName: "judah", lastName: "buzah"}
  password: "123123"
  Object
  : (...)
  ControlContainer
```

```
<form novalidate #myForm = "ngForm" (ngSubmit)="mySentForm()">
  <fieldset ngModelGroup="myHeaderForm">
    <div>
      <label>My First Name : </label>
      <input ngModel name="firstName" type="text">
    </div>

    <div>
      <label>My Last Name : </label>
      <input  ngModel name="lastName" type="text">
    </div>
  </fieldset>

  <div>
    <label>Email : </label>
      <input ngModel name="ooEmail" type="text">
  </div>

  <div>
    <label>Password : </label>
      <input ngModel name="myPassword" type="text">
  </div>

  <div>
    <label>Hobby : </label>
    <select ngModel name="selectedHobby">
      <option value="" disabled selected>Choose a Hobby : </option>
      <option *ngFor="let hobby of hobbies"
                    value="hobby">{{hobby}}</option>
    </select>
  </div>

  <div>
    <button type="submit">Sent my First Form</button>
  </div>

</form>
```

מעקב השליחת טופס

```
@ViewChild('myForm') myFormChanges : any;

public mySentForm():void{
 console.log(this.myFormChanges);

}
```

איכן נמצאים הפקדים

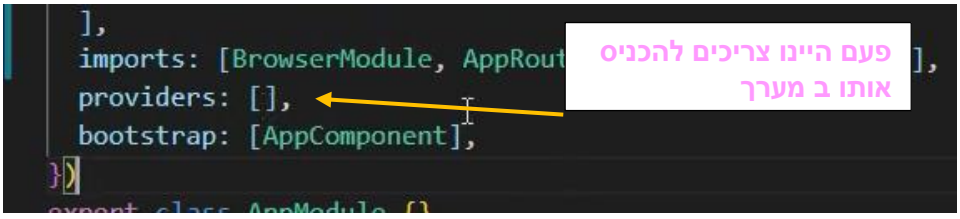


Services & Dependency Injection

Services - מאותחל פעם 1 , single ton , מאפשר לחלוק מידע והעברת נתונים בין קומפנטות , למשל יצירת משתמש לא נרצה שיהיה כמה פונקציות יצירת משתמש

```
PS D:\Angular\sampleApp> ng generate service services/movie
```

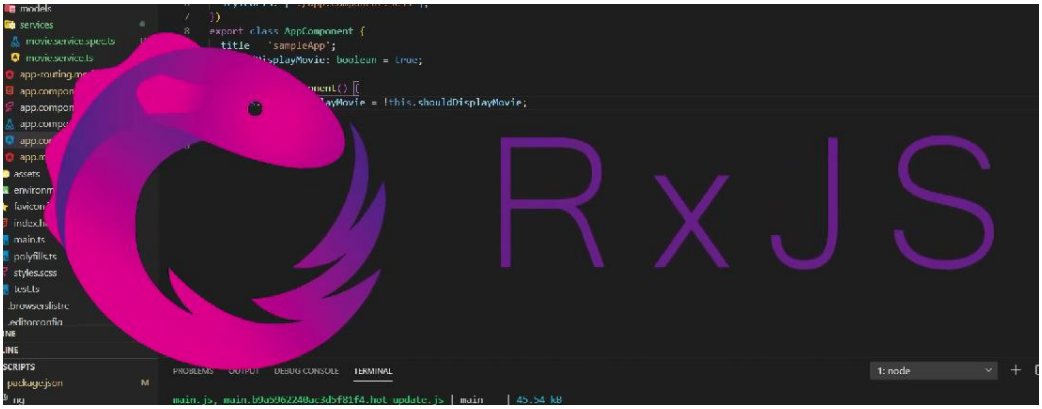
גישה מכל מקום באפליקציה



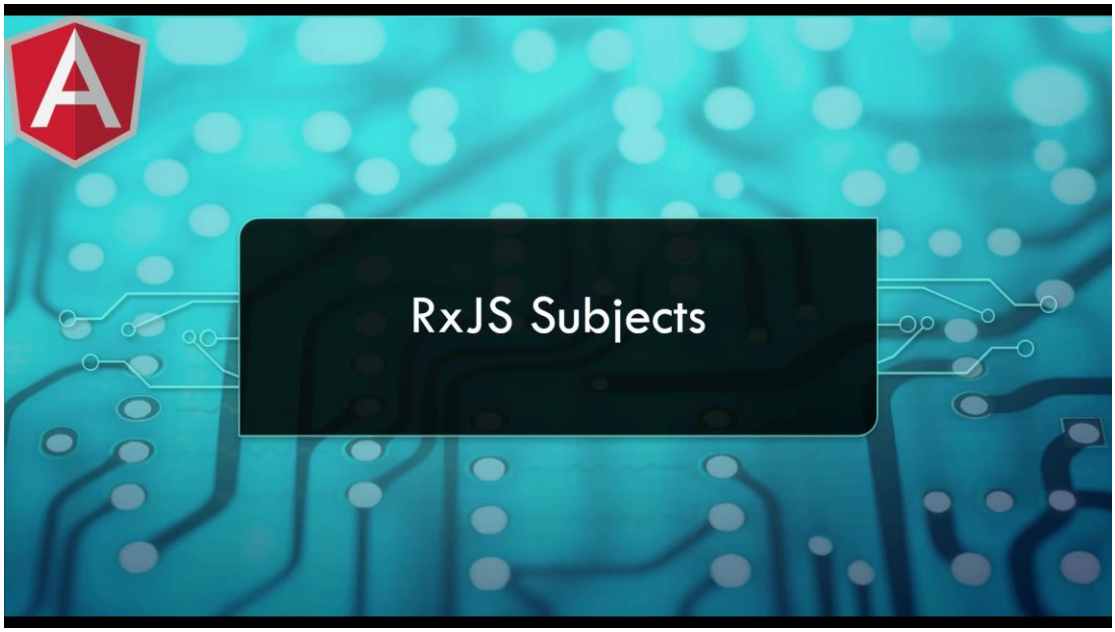
פעם היינו צריכים להכניס אותו ב מערך

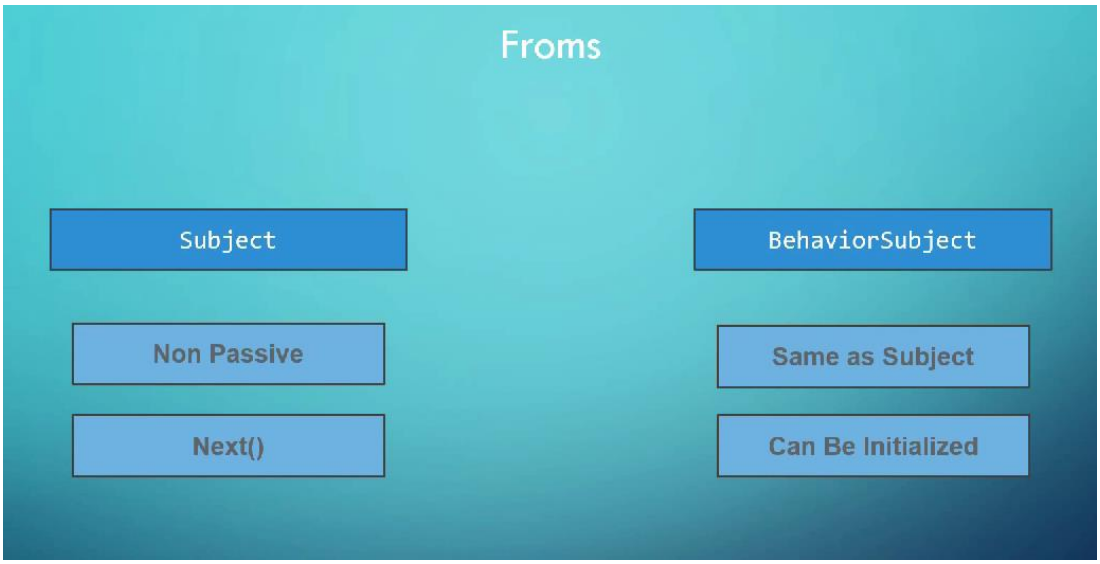
```ts
export class MovieService {
  public random: number = Math.floor(Math.random() * 100);

  private _movies: Movie[] = [];

  constructor() {}

  public getMovies(): Movie[] {
    return this._movies;
  }

  public setMovies(movies: Movie[]): void {
    this._movies = [...this._movies, ...movies];
  }
}
```

```ts
ngOnInit() {
  this.movieService.setMovies(this.listOfMovies);

  this.movies = this.movieService.getMovies();
}
```

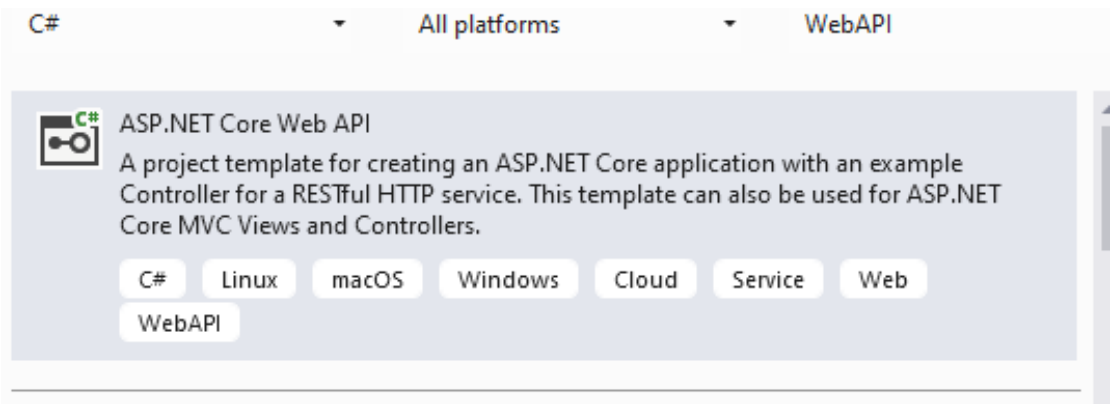


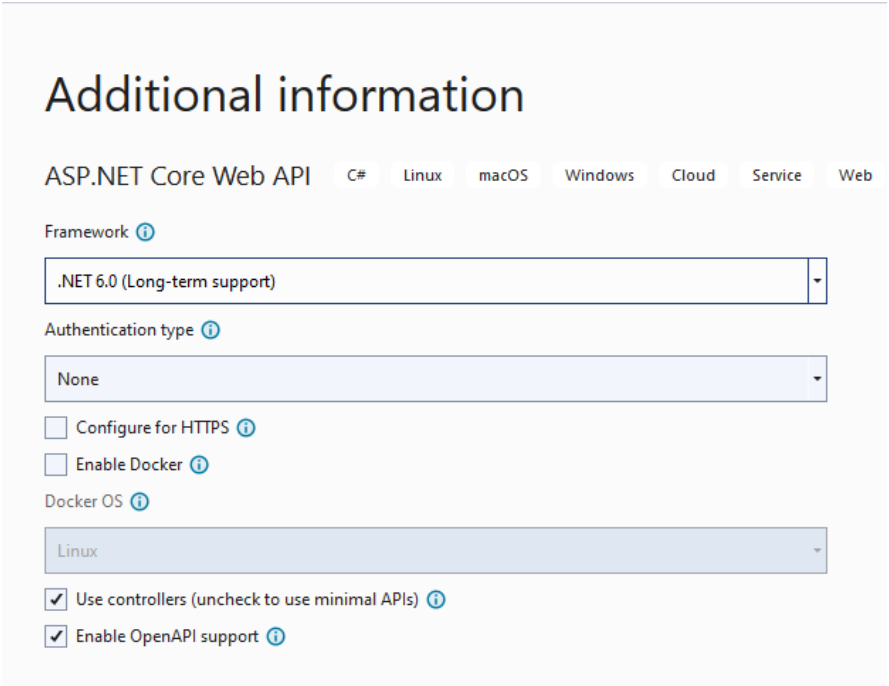
Observables – דוגמה בצאט
שמאזין לשליחת הודעות בלי רענון

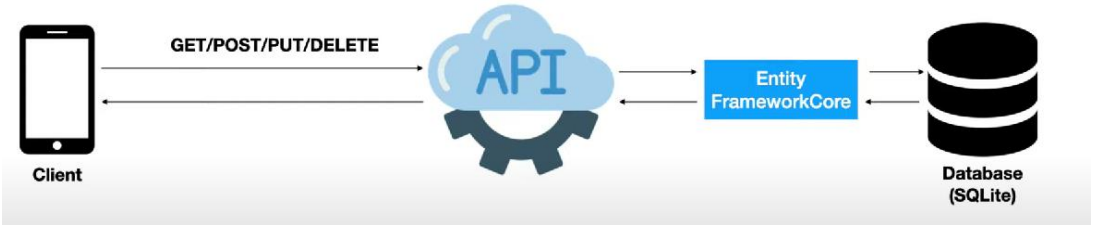

RxJS Subjects

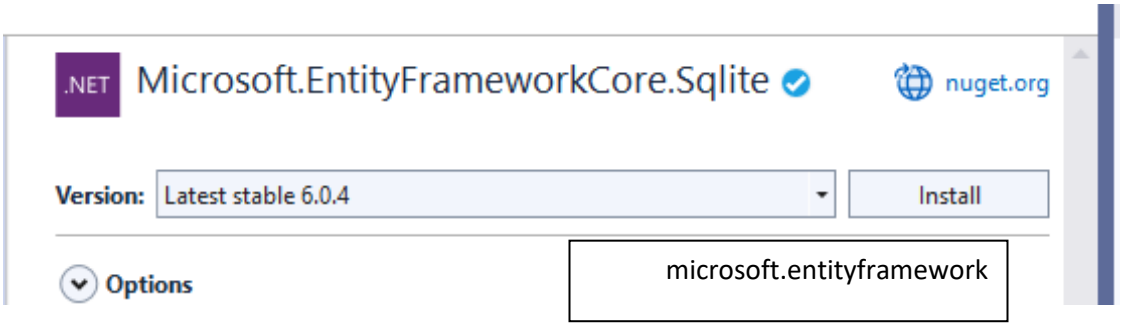

Froms
Subject
Non Passive
Next()
BehaviorSubject
Same as Subject
Can Be Initialized

יצירת Web API

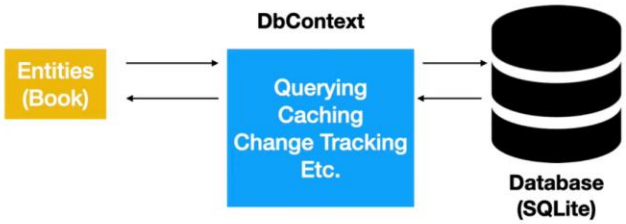








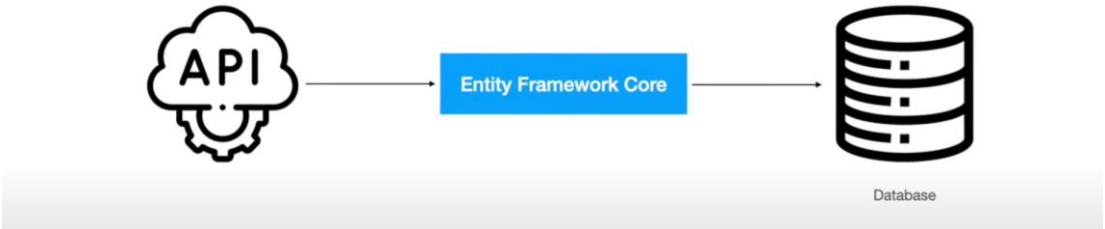

```
namespace JobA.Models
{
    0 references
    public class Book
    {
        prop
    }
}
```



API Paradigms

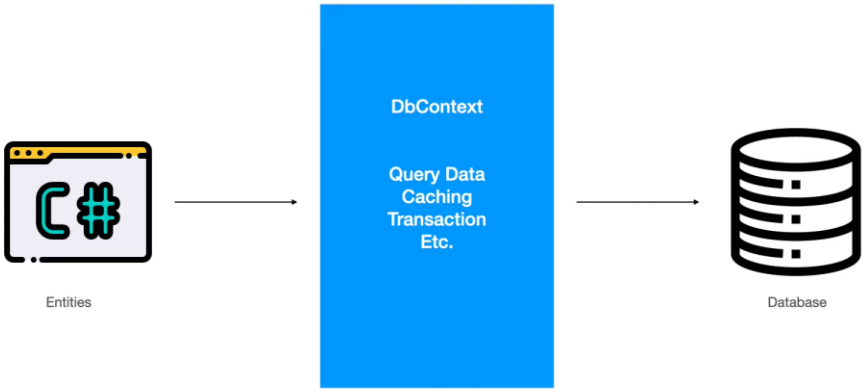
Rest API

Entity Framework Core

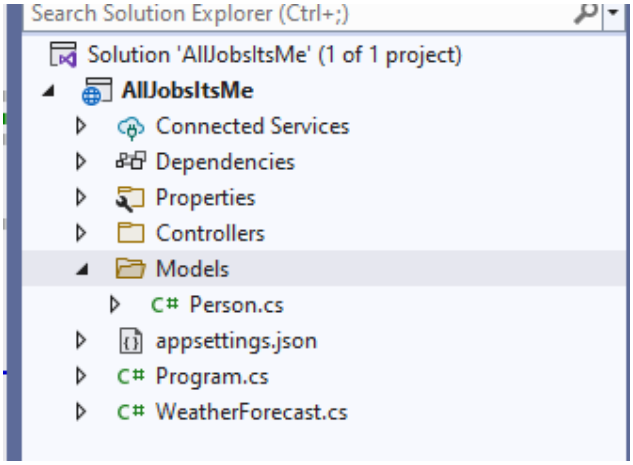


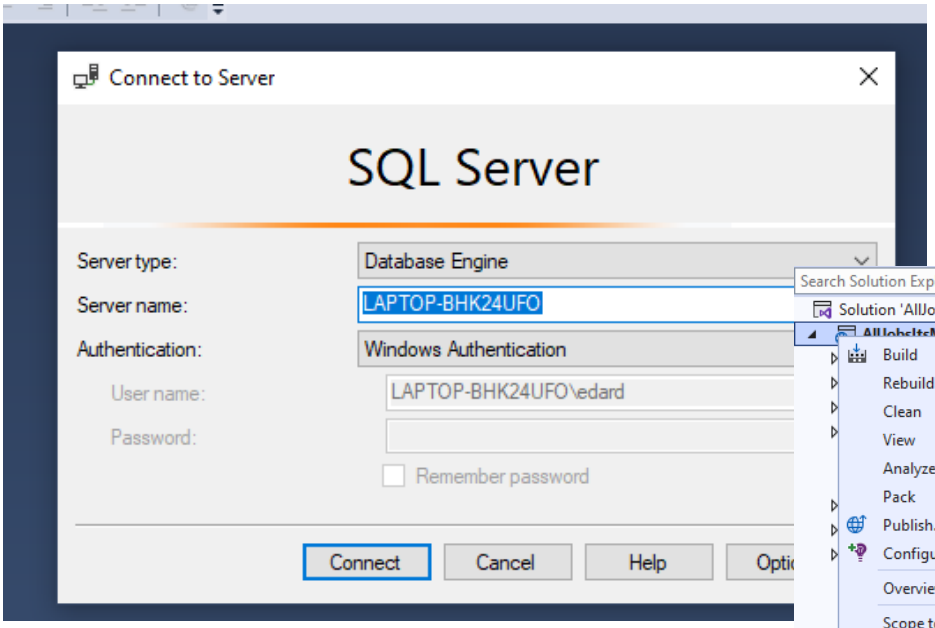
Entity Framework Core

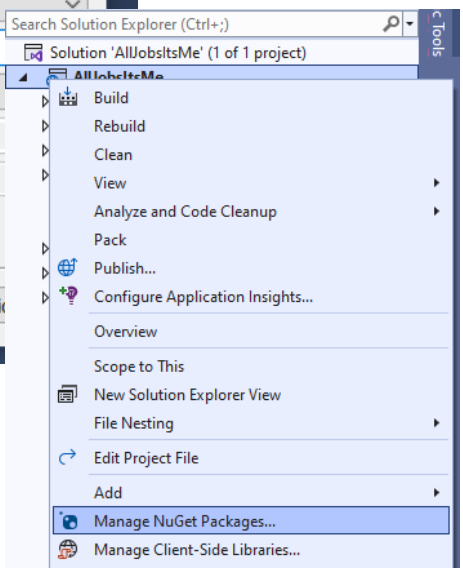
DbContext




```
public class Person
{
    0 references
    public int age { get; set; }
    0 references
    public string name { get; set; }
    0 references
    public kindPerson type { get; set; }

}

1 reference
public enum kindPerson
{
    Child,
    Young,
    Adult
}
}
```



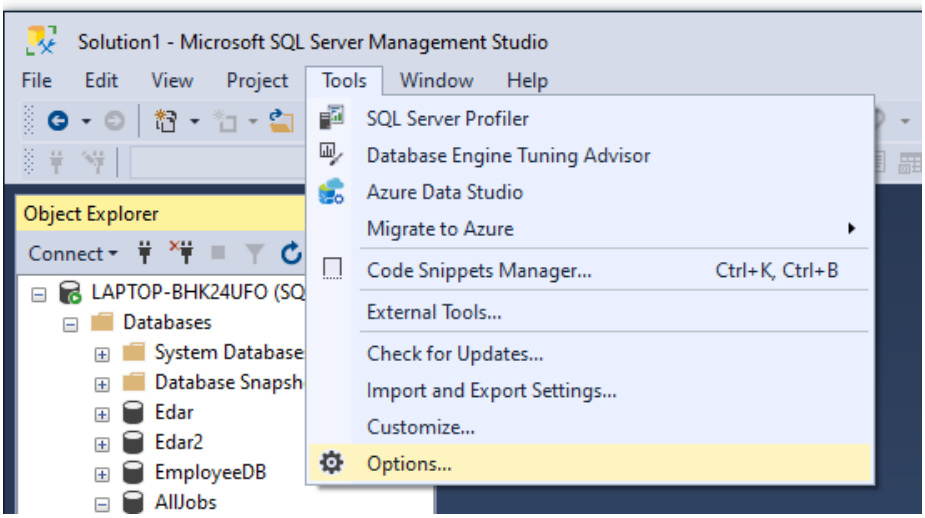


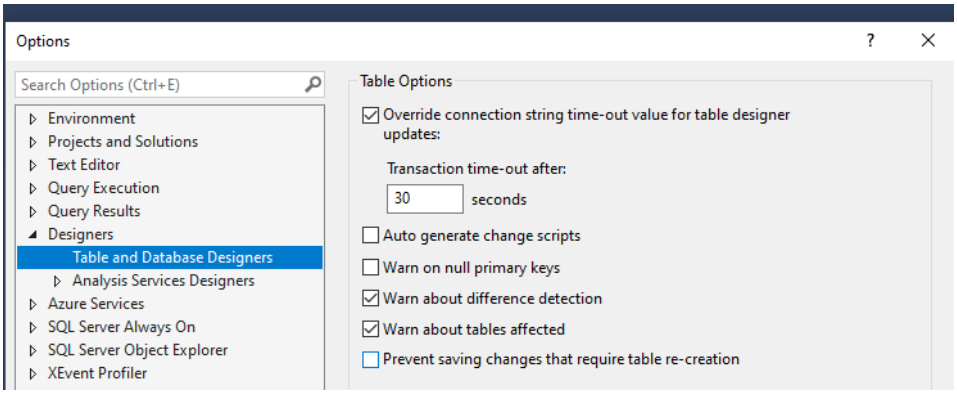


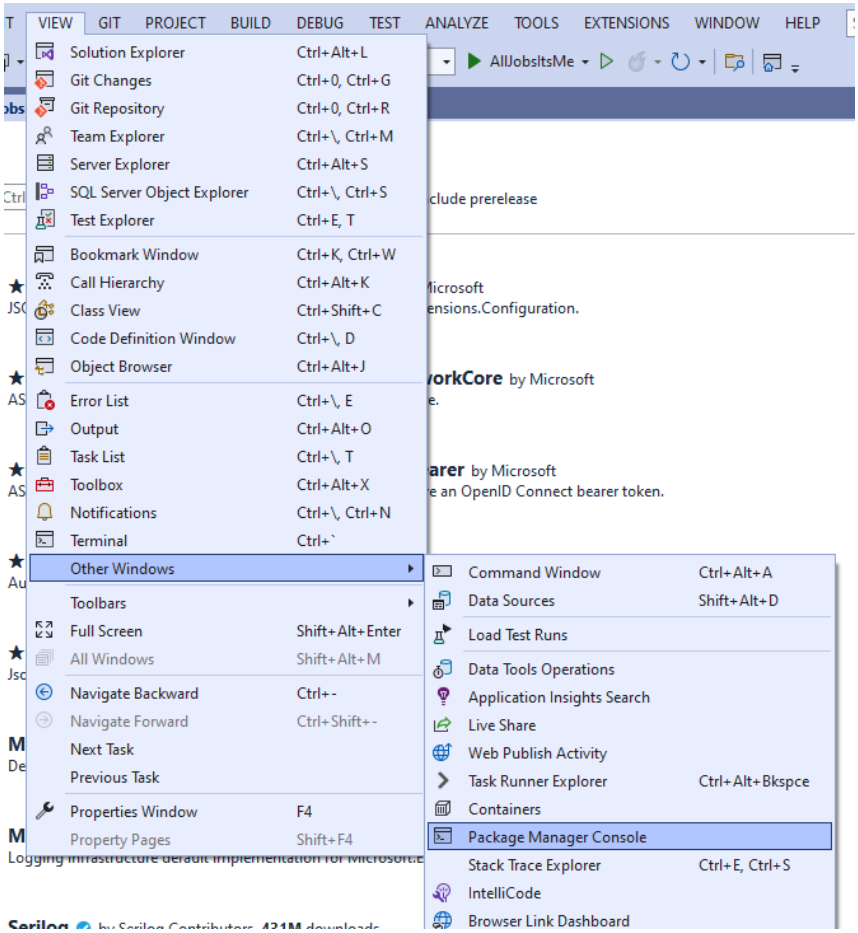
Microsoft.EntityFrameworkCore.SqlServer by Microsoft, **204M** downloads
Microsoft SQL Server database provider for Entity Framework Core.

Microsoft.EntityFrameworkCore.Tools by Microsoft, **143M** downloads
Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.

Microsoft.Extensions.Configuration by Microsoft, **1.11B** downloads 6.0.1
Implementation of key-value pair based configuration for
Microsoft.Extensions.Configuration. Includes the memory configuration provider.



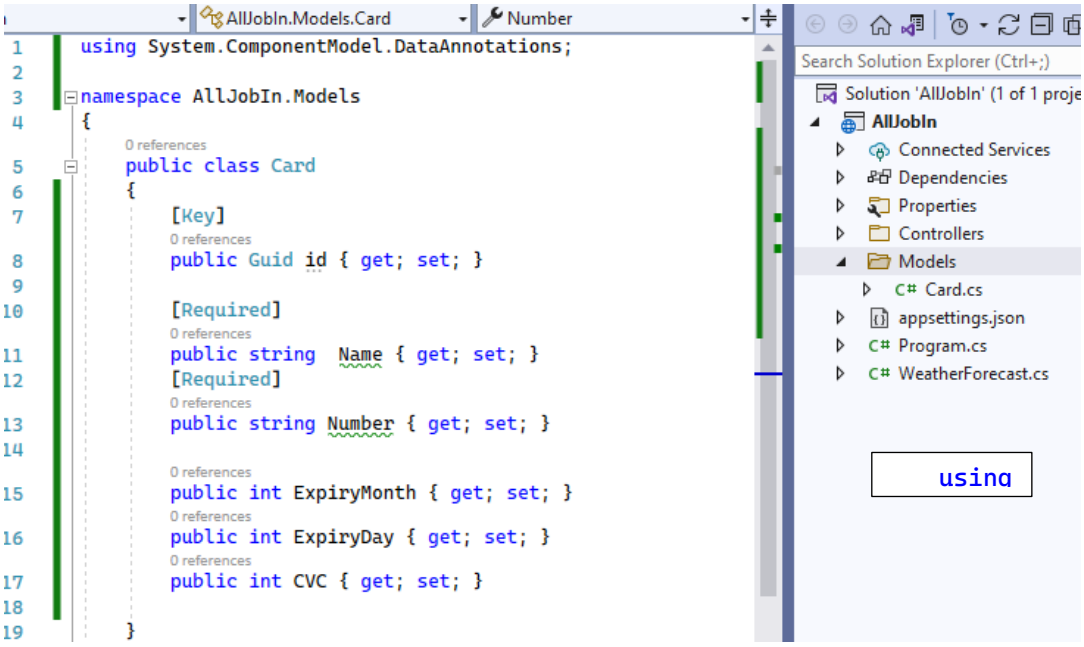



VIEW
GIT
PROJECT
BUILD
DEBUG
TEST
ANALYZE
TOOLS
EXTENSIONS
WINDOW
HELP
AllJobsItsMe
Solution Explorer Ctrl+Alt+L
Git Changes Ctrl+0, Ctrl+G
Git Repository Ctrl+0, Ctrl+R
Team Explorer Ctrl+\, Ctrl+M
Server Explorer Ctrl+Alt+S
SQL Server Object Explorer Ctrl+\, Ctrl+S
Test Explorer Ctrl+E, T
Bookmark Window Ctrl+K, Ctrl+W
Call Hierarchy Ctrl+Alt+K
Class View Ctrl+Shift+C
Code Definition Window Ctrl+\, D
Object Browser Ctrl+Alt+J
Error List Ctrl+\, E
Output Ctrl+Alt+O
Task List Ctrl+\, T
Toolbox Ctrl+Alt+X
Notifications Ctrl+\, Ctrl+N
Terminal Ctrl+`
Other Windows
Toolbars
Full Screen Shift+Alt+Enter
All Windows Shift+Alt+M
Navigate Backward Ctrl+-
Navigate Forward Ctrl+Shift+-
Next Task
Previous Task
Properties Window F4
Property Pages Shift+F4
Command Window Ctrl+Alt+A
Data Sources Shift+Alt+D
Load Test Runs
Data Tools Operations
Application Insights Search
Live Share
Web Publish Activity
Task Runner Explorer Ctrl+Alt+Bkspce
Containers
Package Manager Console
Stack Trace Explorer Ctrl+E, Ctrl+S
IntelliCode
Browser Link Dashboard
clude prerelease
Microsoft
ensions.Configuration.
workCore by Microsoft
arer by Microsoft
e an OpenID Connect bearer token.

Microsoft.EntityFrameworkCore by Microsoft, **401M** downloads
Entity Framework Core is a modern object-database mapper for .NET. It supports LINQ queries, change tracking, updates, and schema migrations with SQL Server, Azure SQL Database, SQLite, Azure Cosmos DB, MySQL, PostgreSQL, and other databases through a provider plugin API.

Microsoft.EntityFrameworkCore.SqlServer by Microsoft, **204M** downloads
Microsoft SQL Server database provider for Entity Framework Core.

Microsoft.EntityFrameworkCore.Tools by Microsoft, **143M** downloads
Entity Framework Core Tools for the NuGet Package Manager Console in Visual Studio.



```
using Microsoft.EntityFrameworkCore;

namespace AllJobIn.Data
{
    0 references
    public class CardDbContext : DbContext
```

Extract base class...
Move to namespace...
Generate Equals(object)...
Generate Equals and GetHashCode...
Generate overrides...
Generate constructor 'CardDbContext()'
Generate constructor 'CardDbContext(options)'
Generate all
Add 'DebuggerDisplay' attribute

Lines 6 to 8

```
{
    public CardDbContext(DbContextOptions options) : base(options)
    {
    }
}
```

Preview changes

Controllers
Data
C# CardDbContext.cs

AllJobIn.Data.CardDbContext

```
using AllJobIn.Models;
using Microsoft.EntityFrameworkCore;

namespace AllJobIn.Data
{
    1 reference
    public class CardDbContext : DbContext
    {
        0 references
        public CardDbContext(DbContextOptions options) : base(options)
        {
        }

        0 references
        public DbSet<Card> Cards { get; set; }
    }
}
```

הוספת מקשר ביו המחלקה לטבלה (ייצוג)

```
0 references
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);
    modelBuilder.Entity<Friends>().ToTable("Friends");
    modelBuilder.Entity<Movies>().ToTable("Movies");
}
```

הוספת מקשר ביו המחלקה לטבלה (ייצוג)

C# Lovers.cs
appsettings.json

```
},
  "AllowedHosts": "*",
  "ConnectionStrings": {
      "Ed": "Data Source =LAPTOP-BHK24UFO;Initial Catalog=AllJobsICame;Integrated Security=True"
  }
```

```
using AllJobsICame.Data.CreateDb;
using Microsoft.EntityFrameworkCore;

var builder = WebApplication.CreateBuilder(args);
```

C# Program.cs
C# WeatherForecast.cs

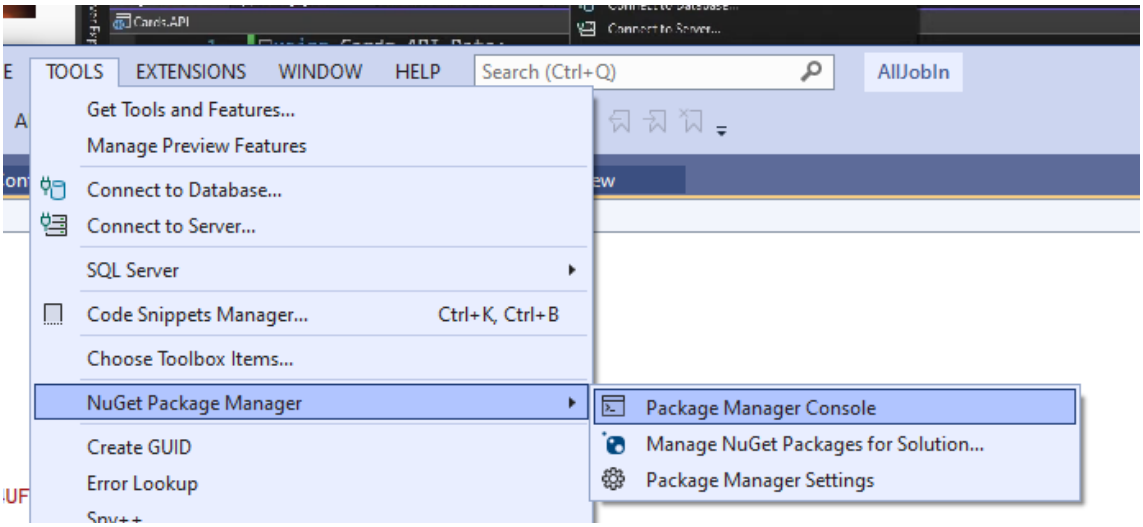
ore/swashbuckle

```
builder.Services.AddDbContext<CreateDb>(options =>
options.UseSqlServer(builder.Configuration.GetConnectionString("Ed")));
```

"Connection Strings": {

"Ed": "Data Source =LAPTOP-

builder.Services.AddDbContext<CreateDb>



Migrations

Migrations allow updating the database with the latest changes on the model.

```
Type 'get-help NuGet' to see all available NuGet commands.

PM> Add-Migration createdatabase -o Data/Migrations
```

```
Add-Migration
createdatabase
-o
```

```
        {
#pragma warning disable 612, 618
            modelBuilder
                .HasAnnotation("ProductVersion", "6.0.4")
                .HasAnnotation("Relational:MaxIdentifierLength", 128);

            SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder, 1L, 1);

            modelBuilder.Entity("MyAllJobsA.Models.Car", b =>
                {
                    b.Property<int>("id")
                        .ValueGeneratedOnAdd()
                        .HasColumnType("int");

                    SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("id"), 1L, 1);

                    b.Property<string>("kind")
                        .IsRequired()
                        .HasColumnType("nvarchar(max)");

                    b.Property<string>("name")
                        .IsRequired()
                        .HasColumnType("nvarchar(max)");

                    b.HasKey("id");

                    b.ToTable("Cars");
                });
#pragma warning restore 612, 618
```

```
add-migration

update-database
```

```
modelBuilder.E
ntity("MyAllJo
```

שיהיה כמה
תכונות מפתח

```
0 references
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    base.OnModelCreating(modelBuilder);
    modelBuilder.Entity<Friends>().HasKey(e=>new { e.Id, e.Name });
    modelBuilder.Entity<Movies>().ToTable("tblMovies");

}
}
```

לטבלה שצירפנו כמה שדות אין לרשום ToTable

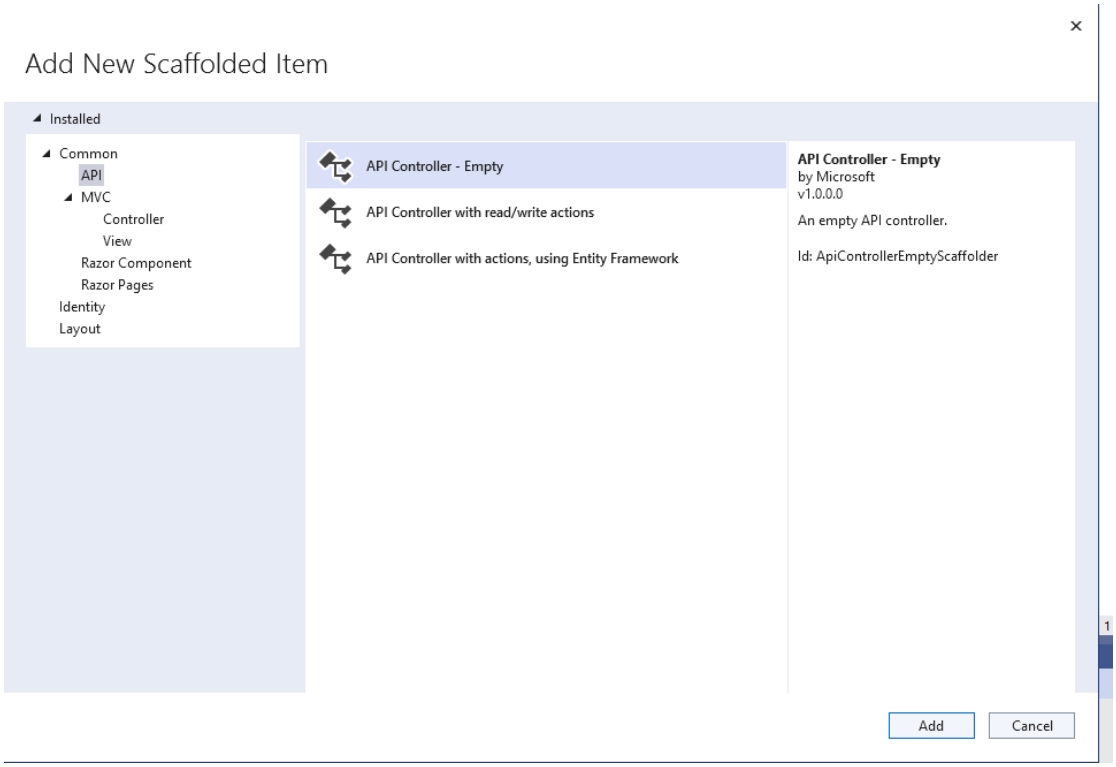
```
protected override
void
OnModelCreating(ModelB
```

```
using
Microsoft.EntityFrameworkC
ore;
using MyAllJobsA.Data;
using
MyAllJobsA.PostRequest;
using MyAllJobsA.Models;
                    using
    MyAllJobsA.Models.DTO;
```

sqlServer- לחיבור הDb , tools – לאינטגרציה , סיבת התקנת ספריות

```
1 reference
public class CarsController : ControllerBase
{
    private readonly CreateMyDbContext _dbContext;

    0 references
    public CarsController(CreateMyDbContext dbContext) => _dbContext = dbContext;

    [HttpGet]
    0 references
    public async Task <IActionResult> GetCars()
    {
        var cars = await _dbContext.Cars.ToListAsync();
        return Ok(cars);
    }
```

```
private readonly
CreateMyDbContext
_dbContext;

                    public
CarsController(CreateMyDbC
      ontext dbContext) =>
   _dbContext = dbContext;
```



IEnumerable

פעמים רבות יש **צורך לעבור בלולאה באוסף של כיתות או רשימות** שהן טיפוסים אנונימיים.
ממשק IEnumerable הוא אחת התכונות הטובות ביותר של שפת C# העוברת בלולאה על האוסף.

```
Customer[] customers = new Customer[]
{
    new Customer {Name="Vithal Wadje",City="Mumbai",Mobile=99999999999,Amount=89.45 },
    new Customer { Name = "Sudhir Wadje", City = "Latur", Mobile = 8888888888888888888
    new Customer { Name = "Anil", City = "Delhi", Mobile = 77777777777777777777, Amour
};
```

Now in the above, customer type array has different types of values we have assigned to the properties that we have created, now let us use the IEnumerable interface loop through above collection as:

```
public IEnumerable<Customer> GetAllCustomer()
{
    return customers;
}
```

In the above example I have assigned the Customer class collection to the IEnumerable to loop through each element, now we have already explained that IEnumerable interface collection can be iterated with the help of foreach loop over the IEnumerable interface collection, now let iterate through each item using IEnumerable interface as

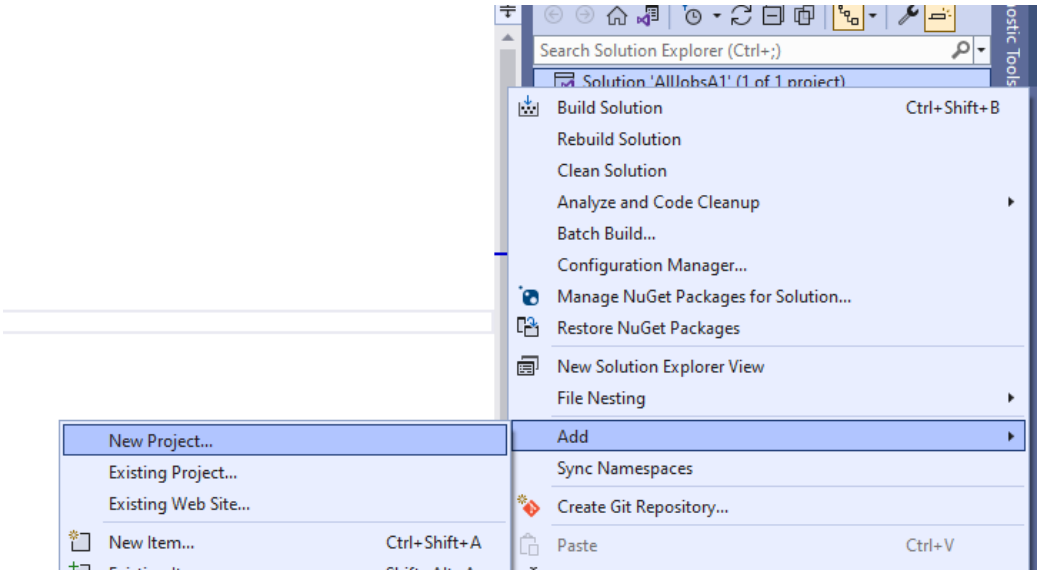
```
foreach (var cust in GetAllCustomer())
{
    Response.Write("Name: "+cust.Name + "
    <br> " +"City: " +cust.City + " <br> " +"Mobile "+cust.Mobile+"<br>
    "+"Amount :" +cust.Amount.ToString("c") + "<br>"+"-----"+"<br>");
}
```

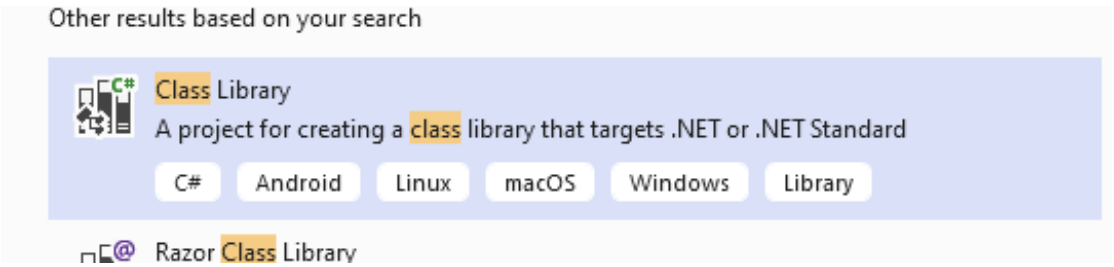
FromBody - לאלץ את ה-API של האינטרנט לקרוא **סוג פשוט מגוף הבקשה**,

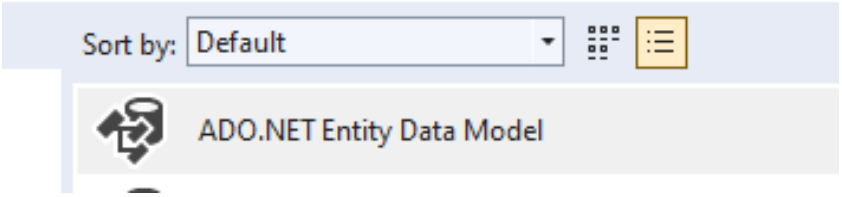
FromUri– כדי לאלץ את ה-API של האינטרנט **לקרוא סוג מורכב מה-URI**, **הוסף לפרמטר את התכונה** [FromUri].

Route– הוספת פרמטרים לנתיב

```
}

[HttpPut]
[Route("{id:int}")]
0 references
public void Put(int id,[FromBody] string myNewString)
{
    myString[id] = myNewString;
}

[HttpDelete]
0 references
public void Delete(int id)
{
    myString.RemoveAt(id);
}

[HttpGet]
[Route("{id:int}")]
0 references
public string Get(int id)
{
    return myString[id];
}
```


Search Solution Explorer (Ctrl+;)
Build Solution Ctrl+Shift+B
Rebuild Solution
Clean Solution
Analyze and Code Cleanup
Batch Build...
Configuration Manager...
Manage NuGet Packages for Solution...
Restore NuGet Packages
New Solution Explorer View
File Nesting
Add
Sync Namespaces
Create Git Repository...
Paste Ctrl+V
New Project...
Existing Project...
Existing Web Site...
New Item... Ctrl+Shift+A


Other results based on your search
Class Library
A project for creating a class library that targets .NET or .NET Standard
C#
Android
Linux
macOS
Windows
Library
Razor Class Library


Sort by: Default
ADO.NET Entity Data Model

```
using Microsoft.EntityFrameworkCore;
using MyAllJobsA.Data;

namespace MyAllJobsA.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    1 reference
    public class CarsController : ControllerBase
    {
        private readonly CreateMyDbContext _dbContext;

        0 references
        public CarsController(CreateMyDbContext dbContext) => _dbContext = dbContext;

        [HttpGet]
        0 references
        public async Task <IActionResult> GetCars()
        {
            var cars = await _dbContext.Cars.ToListAsync();
            return Ok(cars);
        }

        [HttpGet]
        [Route("{id:int}")]
        0 references
        public async Task<IActionResult> GetCarsById(int id)
        {
            var car = await _dbContext.Cars.FirstOrDefaultAsync(x => x.id > id);
            if(car == null)
                return NotFound();
            return Ok(car);
        }
    }
}
```

A `nameof` expression

מייצר את השם של משתנה, סוג או איבר בתור קבוע המחרוזת:

```
C#

Console.WriteLine(nameof(System.Collections.Generic));  // output: Generic
Console.WriteLine(nameof(List<int>));  // output: List
Console.WriteLine(nameof(List<int>.Count));  // output: Count
Console.WriteLine(nameof(List<int>.Add));  // output: Add

var numbers = new List<int> { 1, 2, 3 };
Console.WriteLine(nameof(numbers));  // output: numbers
Console.WriteLine(nameof(numbers.Count));  // output: Count
Console.WriteLine(nameof(numbers.Add));  // output: Add
```

CreatedAtAction(String, Object, Object)

Creates a CreatedAtActionResult object that produces a Status201Created response.

```
C#                                                                    Copy

[Microsoft.AspNetCore.Mvc.NonAction]
public virtual Microsoft.AspNetCore.Mvc.CreatedAtActionResult CreatedAtAction (string? actionName,
object? routeValues, object? value);
```

Parameters

`actionName` String

The name of the action to use for generating the URL.

`routeValues` Object

The route data to use for generating the URL.

`value` Object

The content value to format in the entity body.

```http
[HttpPost]
0 references
public async Task<IActionResult> AddCar(NewCar car)
{
    var myCar = new Car()
    {
        kind = car.kind,
        name = car.name
    };

    await _dbContext.Cars.AddAsync(myCar);
    await _dbContext.SaveChangesAsync();
    return CreatedAtAction(nameof(GetCarsById), new { id = myCar.id }, car);
}
```

The DTO Pattern (Data Transfer Object)

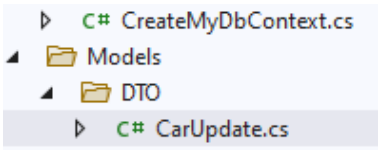
אובייקטים הנושאים נתונים בין תהליכים על מנת לצמצם את מספר הקריאות לשיטות .הדפוס הוצג לראשונה על ידי מרטין פאולר בספרו . EAA

פאולר הסביר שהמטרה העיקרית של הדפוס **היא לצמצם נסיעות הלוך ושוב לשרת** על ידי אצווה של מספר **פרמטרים בקריאה אחת** .זה **מקטין** את תקורה של הרשת בפעולות מרוחקות כאלה.

יתרון נוסף הוא אנקפסולציה של הלוגיקה של הסדרת (המנגנון שמתרגם את מבנה האובייקט והנתונים לפורמט ספציפי שניתן לאחסן ולהעביר) .הוא מספק נקודת שינוי אחת בניואנסים בהמשכה .זה גם מנתק את דגמי התחום משכבת המצגת, ומאפשר לשניהם להשתנות באופן עצמאי.

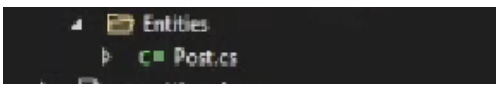
ם מבני נתונים שטוחים שאינם מכילים שום היגיון עסקי .

```http
namespace MyAllJobsA.Models.DTO
{
    1 reference
    public class CarUpdate
    {
        2 references
        public string name { get; set; }
    }
}
```



כל אוביקט בפני עצמו

Entity



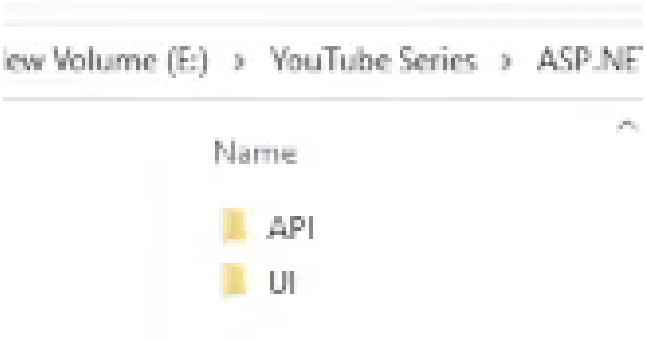
FindAsync – חיפוש על פי מפתח ראשי

```
[HttpDelete]
[Route("{id:int}")]
0 references
public async Task<IActionResult> DeleteOooCar(int id)
{
    var existCar = await _dbContext.Cars.FindAsync(id);
    if(existCar != null)
    {
        _dbContext.Remove(existCar);
        await _dbContext.SaveChangesAsync();
        return Ok(existCar);
    }
    return NotFound();
}
```

Alpha – לשליחת מחרוזת

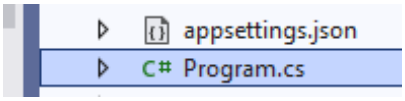
```
[HttpDelete]
[Route("{name:alpha}")]
0 references
public async Task<IActionResult> DeleteOoo2Car(string name)
{
    var existCar = await _dbContext.Cars.FirstOrDefaultAsync(x=>x.name == name);
    if (existCar != null)
    {
        _dbContext.Remove(existCar);
        await _dbContext.SaveChangesAsync();
        return Ok(existCar);
    }
    return NotFound();
}
}
```

```
[HttpPost]
0 references
public async Task<IActionResult> AddCar(NewCar car)
{
    var myCar = new Car()
    {
        kind = car.kind,
        name = car.name
    };

    await _dbContext.Cars.AddAsync(myCar);
    await _dbContext.SaveChangesAsync();
    return CreatedAtAction(nameof(GetCarsById), new { id = myCar.id }, car);
}

[HttpPut]
[Route("{id:int}")]
0 references
public async Task<IActionResult> UpdateCar([FromRoute] int id,CarUpdate carU)
{
    var car = new Car()
    {
        name = carU.name
    };
    var existCar = await _dbContext.Cars.FindAsync(id);
    if(existCar == null)
    {
        return NotFound();
    }
    else
    {
        existCar.name = carU.name;
        await _dbContext.SaveChangesAsync();
        return Ok(existCar);
    }
}
```

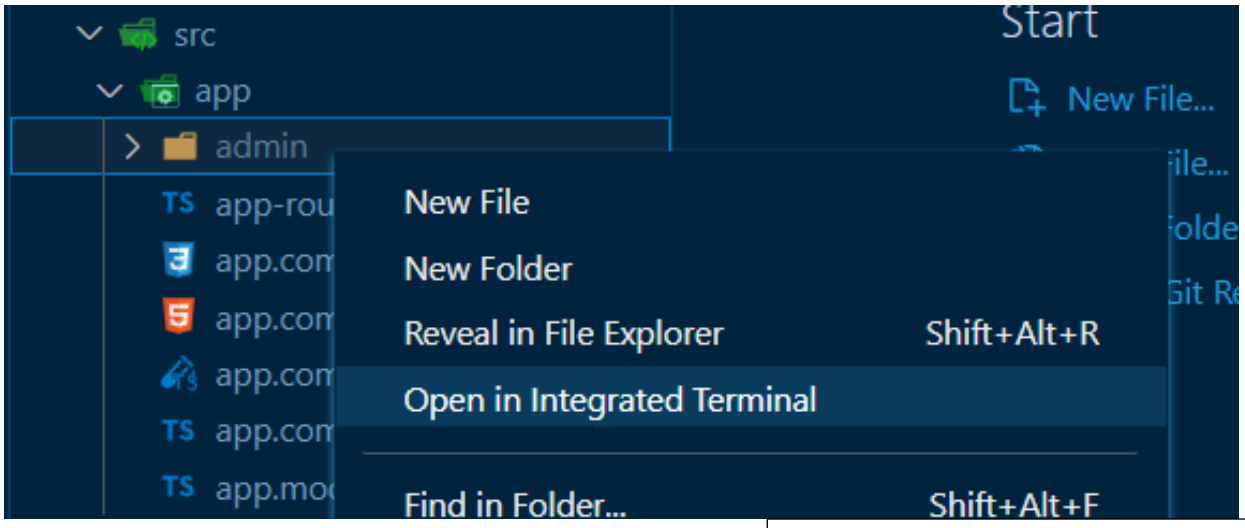


הרשאות לגשת ל API מכל מחשב / נתיב

```
builder.Services.AddCors(options => options.AddPolicy("EdPolicy", policy =>
{
    policy.AllowAnyHeader().AllowAnyOrigin().AllowAnyMethod();
}));

// Configure the HTTP request pipeline.
if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseHttpsRedirection();

app.UseCors("EdPolicy");
```



builder.Services.AddCors



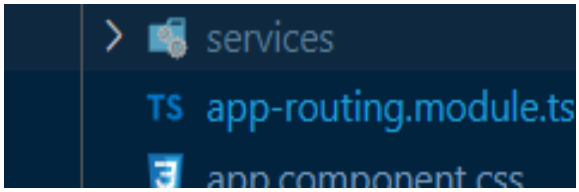
קבלת כל הרכבים

app-routing.module.ts

AllJobsTest1 > src > app > app-routing.module.ts > ...

```
import { AdminCarComponent } from './admin/admin-car/admin-car.component';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';

const routes: Routes = [
  {path:'theManager/Car',
  component:AdminCarComponent
  }
];
```

הוספת נתיב לקומפננטה




```
ng g s actionsCar
```

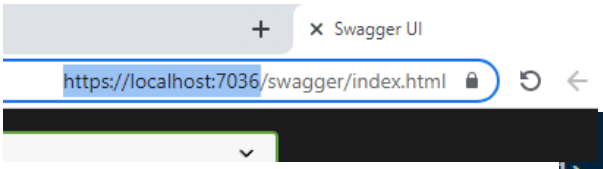
הוספת שרת לטיפול בבקשות

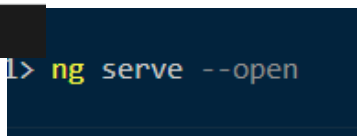
```
import { AdminCarComponent } from './admin/admin-car/ad
import {HttpClientModule} from '@angular/common/http'

You, 1 second ago | 2 authors (Edar Daniel and others)
@NgModule({
  declarations: [
    AppComponent,
    AdminCarComponent
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    HttpClientModule
  ],
  providers: [],       Edar Daniel, 10 minutes ago • ini
  bootstrap: [AppComponent]
})
```

הוספת קונפיגורציה לבקשות HTTP

```
import
{HttpClientModu
```

הגדרת נתיב השרת



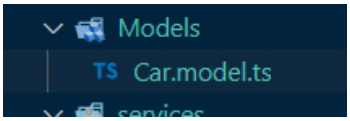



```ts
export const environment = {
  production: false,
  apiBaseUrl:'https://localhost:7036'
};
```

```ts
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { environment } from 'src/environments/environment';

@Injectable({
  providedIn: 'root'
})
export class ActionsCarService {

  constructor(private Http:HttpClient) { }
  apiBaseUrl = environment.apiBaseUrl;

  getAllCars(){
    this.Http.get(this.apiBaseUrl+'/api/Cars');
  }
}
```

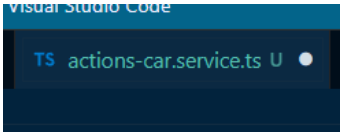
הגדרת גישה לשרת

יצירת מודל שנראה כמו בשרת




```ts
export interface ICar{
  id:number,
  name:string,
  kind:string
}
```

```cs
3 references
public class Car
{
    2 references
    public int id { get; set; }
    4 references
    public string name { get; set; }
    1 reference
    public string kind { get; set; }
}
```



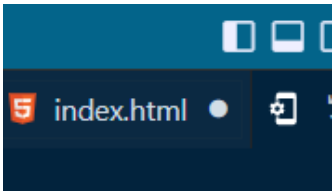
יצירת פונקציה שמבקשת רכבים (אסינכרוני)

```ts
getAllCars():Observable<ICar[]>{
  return this.Http.get<ICar[]>(this.apiBaseUrl+'/api/Cars');
}
```

בקומפננטה שניצור תחילה
נבקש את השירות

```
})
export class AdminCarComponent implements OnInit {

  constructor(private actionCarService:ActionsCarService) { }

  ngOnInit(): void {
    this.actionCarService.getAllCars()
    .subscribe(
      myRes=>{
        console.log(myRes);
      }
    )
  }

}
```

נרשם לאירוע כי הוא אסינכרוני



תחילה נייבא bootstrap

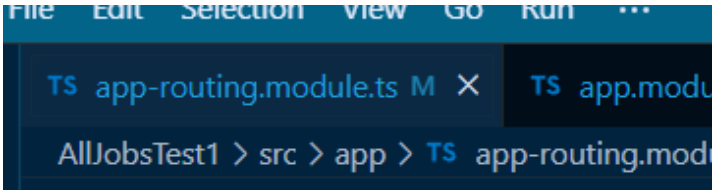
```
<meta name="viewport" content="width=device-width, initial-scale=1">
<link rel="icon" type="image/x-icon" href="favicon.ico">
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFldvKuhf
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OM
</head>
<body>
```

```
<p>admin-car works!</p>

<table  class="table">
  <thead>
    <th scope="col">#</th>
    <th scope="col">Name</th>
    <th scope="col">Kind</th>
  </thead>
  <tbody>
    <tr *ngFor="let car of cars">
      <td>{{car.id}}</td>
      <td>{{car.name}}</td>
      <td>{{car.kind}}</td>
    </tr>
  </tbody>
</table>
```

ניצור טבלה מתשובת הפונקציה

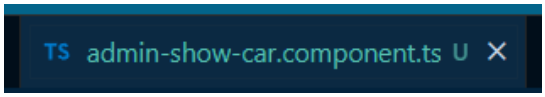
```
constructor(private actionCarService:ActionsCar

cars:ICar[] =[];

ngOnInit(): void {
  this.actionCarService.getAllCars()
  .subscribe(
    myRes=>{
    this.cars = myRes;
    }
  )
}
```

קבלת רכב




```
  {path:'theManager/Car/:myIdCar',
   component:AdminShowCarComponent},
];
```

הגדרת נתיב דינמי



paramMap - מפה המספקת גישה לפרמטרים הנדרשים והאופציונליים הספציפיים למסלול. המפה תומכת באחזור ערך בודד עם get() או מספר ערכים עם getAll().

```
export class AdminShowCarComponent implements OnInit {
  constructor(
    private route: ActivatedRoute,
    private actionsCarService: ActionsCarService
  ) {}

  ngOnInit(): void {
    this.route.paramMap.subscribe((myParam) => {
      const id = myParam.get('myIdCar');

      if (id) {
        this.actionsCarService.getCarInOnID(id)
            .subscribe((myRes) => {
          console.log(myRes);
        });
      }
    });
  }
}
```

```
<button>
  <a style="color: blue;text-decoration: none;"
  [routerLink]="['/theManager/Car',car.id]">
    Go Car
  </a>
</button>
```

routerLink - הופך את הרכיב הזה לקישור שמתחיל ניווט למסלול. ניווט פותח רכיב אחד או יותר מנותב במיקום אחד או יותר של <router-outlet>

עריכת רכב

TS app.module.ts M

```
],
imports: [
  BrowserModule,
  AppRoutingModule,
  HttpClientModule,
  FormsModule
],
providers: [],
```

הוספת פקד טופס

```
</div>
<div class="mb-3">
  <label class="form-label">Url Handle</label>
  <input class="form-control" name="urlHandle" [(ngModel)]="post.urlHandle">
</div>
<div class="form-check mb-3">
  <input class="form-check-input" type="checkbox" name="visible" [(ngModel)]="post.visible">
  <label class="form-check-label">
    Visible
  </label>
</div>
<div class="mb-3">
  <label class="form-label">Publish Date</label>
  <input class="form-control" name="publishDate" [(ngModel)]="post.publishDate">
</div>
<div class="mb-3">
  <label class="form-label">Updated Date</label>
  <input class="form-control" name="updatedDate" [(ngModel)]="post.updatedDate">
</div>
<div class="mb-3">
```

נוסיף מודל ספציפי לעדכון נזכיר חלוקה entity ו dto

Models
- Car.model.ts
- UpCar.model.ts

services

```
1  export interface IUpCar{
2    name:string,
3  }
```

נוסיף פעולת API

TS actions-car.service.ts U

```
upCarByID(myUpId:string , upCar:IUpCar):Observable<ICar>
{
  return this.Http.put<ICar>(this.apiBaseUrl+'api/Cars/'+ myUpId,upCar);
}
```

```
1  import { IUpCar } from './../Models/UpCar.model';
2  import { ICar } from './../Models/Car.model';
3  import { HttpClient } from '@angular/common/http';
4  import { Injectable } from '@angular/core';
5  import { environment } from 'src/environments/environment';
6  import { Observable } from 'rxjs';
```

נוסיף טופס לשליחה וקבלת נתונים

```
<p>admin-show-car works!</p>
<div class="container" *ngIf="Car">

  <form #myForm="ngForm" (ngSubmit)="mySent()">

    <div class="mb-3">
      <label class="form-label">The Id Of Car</label>
      <input type="text" name="IdCar" class="form-control" [(ngModel)]="Car.id">
      <label class="form-label">The Name Of Car</label>
      <input type="text" name="NameCar" class="form-control" [(ngModel)]="Car.nam
      <label class="form-label">The Type Of Car</label>
      <span> {{Car.kind}}</span>
      <br>
      <select name="typeCar" [(ngModel)]="Car.kind">
        <option [ngValue]=""  disabled >Choose a kind car</option>
        <option *ngFor="let type of TypeCars" [ngValue]="type">
          {{type}}
        </option>
      </select>
      <br>
    </div>
    <button type="submit">Sent</button>
  </form>

</div>
```

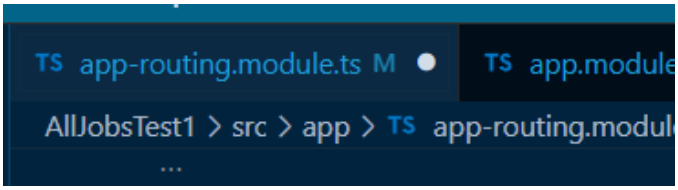
```
<p>admin-show-car works!</p>
```

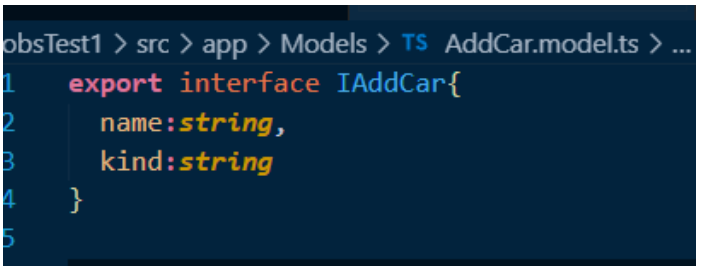

nt.html M
admin-car.component.html U
admin-show-car.component.ts U
admin-show-car > admin-show-car.component.ts > AdminShowCarComponent
import { ActionsCarService } from './../../services/actions-car.servi
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
@Component({
selector: 'app-admin-show-car',
templateUrl: './admin-show-car.component.html',
styleUrls: ['./admin-show-car.component.css'],
})
export class AdminShowCarComponent implements OnInit {
constructor(
private route: ActivatedRoute,
private actionsCarService: ActionsCarService
) {}
Car: ICar | undefined;
TypeCars = [
"BMV",
"Jaguar",
"Mercedes-Benz"
]
ngOnInit(): void {
this.route.paramMap.subscribe((myParam) => {
const id = myParam.get('myIdCar');
if (id) {
this.actionsCarService.getCarInOnID(id)
.subscribe((myRes) => {
this.Car=myRes;
});
}
});
}
קבלת ושמירת הרכב

שינוי ערכי הרכב לפי טופס

```
  mySent()
  {
      const upCar : IUpCar = {
        name : this.Car?.name!,
        kind : this.Car?.kind!
      }
      this.actionsCarService.upCarByID(this.Car?.id.toString()!,upCar)
      .subscribe(
        res=>{
          alert("Success")
        }
      )
  }
}
```

הוספת רכב






```
const routes: Routes = [
  {path:'theManager/Car',
  component:AdminCarComponent
  },

  {path:'theManager/Car/Add',
   component:AdminAddMyCarComponent},

   {path:'theManager/Car/:myIdCar',
   component:AdminShowCarComponent},

];
```

חשוב לשים לב **שאם נרצה להשתמש בנתיב עם מילה שמורה שזה מעל הנתיב הדינמי**

admin-add-my-car.component.ts U | app.module.ts M | admin-show-car.compo

AllJobsTest1 > src > app > admin > admin-add-my-car > admin-add-my-car.component.ts

```
})
export class AdminAddMyCarComponent implements OnInit {

  constructor(private actionsCarService:ActionsCarService) { }

  ngOnInit(): void {
  }

  Car:IAddCar={
    kind:'',
    name:''
  }

  TypeCars = [
    "BMV",
    "Jaguar",
    "Mercedes-Benz"
  ]

  mySent()
  {
    this.actionsCarService.newCar(this.Car)
      .subscribe(
        res=>{
          alert("Success")
        }
      )
  }

}
```

יצירת **אובייקט ריק בשביל שדות ריקים ואז שליחה**

admin-add-my-car.component.html U ● | admin-add-my-car.component.ts U | app.module.ts M

AllJobsTest1 > src > app > admin > admin-add-my-car > admin-add-my-car.component.html > ...

Go to component

```html
<p>admin-add-my-car works!</p>
<div class="container" *ngIf="Car">

  <form #myForm="ngForm" (ngSubmit)="mySent()">

    <div class="mb-3">
      <label class="form-label">The Name Of Car</label>
      <input type="text" name="NameCar" class="form-control" [(ngModel)]="Car.name">
      <label class="form-label">The Type Of Car</label>
      <span> {{Car.kind}}</span>
      <br>
      <select name="typeCar" [(ngModel)]="Car.kind">
        <option [ngValue]="" disabled >Choose a kind car</option>
        <option *ngFor="let type of TypeCars" [ngValue]="type">
          {{type}}
        </option>
      </select>
      <br>
    </div>
    <button type="submit">Sent</button>
  </form>

</div>

```

יצירת הטופס למילוי חדש

הקריאה

actions-car.service.ts U

```typescript
newCar(addCar:IAddCar) :Observable<ICar>
{
  return this.Http.post<ICar>(this.apiBaseUrl+'api/Cars',addCar);
}
```

```
[HttpPost]
0 references
public async Task<IActionResult> AddCar(NewCar car)
{
    var myCar = new Car()
    {
        kind = car.kind,
        name = car.name
    };

    await _dbContext.Cars.AddAsync(myCar);
    await _dbContext.SaveChangesAsync();
    return CreatedAtAction(nameof(GetCarsById), new { id = myCar.id }, car);
}
```

מחיקת רכב

```
OpsDeleteMe(delId:string):Observable<ICar>{
  return this.Http.delete<ICar>(this.apiBaseUrl+'api/Cars/'+delId);
}
```

```
  }

  addPost(addPostRequest: AddPostRequest): Observable<Post> {
    return this.http.post<Post>(this.apiBaseUrl + '/api/posts', addPostRequest);
  }

  deletePost(id: string | undefined): Observable<Post> {
    return this.http.delete<Post>(this.apiBaseUrl + '/api/posts/' + id);
  }
}
```

TS admin-show-car.component.ts U

```
delCar()
{
  this.actionsCarService.OpsDeleteMe(this.Car?.id.toString()!)
  .subscribe(
    res=>{
      alert("Success")
    }
  )
}
```

```
    <br>
    <select name="typeCar" [(ngModel)]="Car.kind">
      <option [ngValue]=""  disabled >Choose a kind car</option>
      <option *ngFor="let type of TypeCars" [ngValue]="type">
        {{type}}
      </option>
    </select>
    <br>
  </div>
  <button type="submit" class="btn btn-primary">Sent</button>
  <button  class="btn btn-danger" style="margin-left: 30px;" (click)="delCar()">Sent</button>
</form>

</div>
```

יצירת נתיב ראשי

```
const routes: Routes = [

  {
    path:'/',
    component:AdminMainComponent
  },
```