**Session 01:**

**Part 01 (project structure and postman):**

* API Vs MVC:

|  |  |
| --- | --- |
| MVC | API |
| - هوا عباره عن MVC: Model View Controller  - يعني لما اعمل request بيreturn الview | - هوا عباره عن MC:Model Controller  - يعني لما اعمل request بيreturn الjson فيه data |

1. Project steps:
2. Creating API Project :-

* Open VS code
* Create New Project
* Project type: web API
* ASP.NET core web API (C#)

ملحوظة:

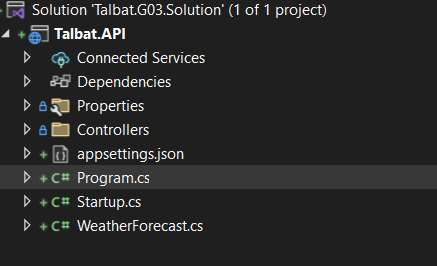
* البروجيكت ده هي target الEnd user مش الAdmin ، لو الAdmin بعمل Dashboard بالMVC.
* في الAPI : الEndPoint هيا Action.

1. Name the project – Solution

- Enable HTTP Request

- Enable OpenApi Support (Enable Swagger)

1. Project Structure:



1. Connected services

* ودي بتكون خاصة بالDocker

1. Dependencies

* هننزل الPackages عليها
* الProject اللي احنا هنعملها Add project Reference

1. Properties:

* launchSettings.json:

Profiles:

1. Kestrel(Project):

- 2 ports

- Code:

|  |
| --- |
| "Talbat.API": {  "commandName": "Project",  "dotnetRunMessages": "true",  "launchBrowser": true,  "launchUrl": "swagger",  "applicationUrl": "https://localhost:5001;http://localhost:5000",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  } |

1. IIS Express:

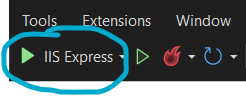
- 2 ports

- Code:

|  |
| --- |
| "IIS Express": {  "commandName": "IISExpress",  "launchBrowser": true,  "launchUrl": "swagger",  "environmentVariables": {  "ASPNETCORE\_ENVIRONMENT": "Development"  }  "iisSettings": {  "windowsAuthentication": false,  "anonymousAuthentication": true,  "iisExpress": {  "applicationUrl": "http://localhost:3023",  "sslPort": 44303  }  } |

ملحوظة:

* اختيار الprofile اللي هrun منه:



1. Controller
2. Appsetting.json

* ده اللي بنحط فيه الconnection string

1. Program

* أي ASP Project الself-Hosted على الconsole app
* الprogram هوا الentry Point
* لما اعمل run للبروجيكت في 2 بروجيكت بيrun:-



1. الKestrel(console app)
2. البروجيكت بتاع الapi او الmvc هيبقى hosted على الconsole app

|  |
| --- |
| namespace Talbat.API  {  public class Program  {  public static void Main(string[] args)  {  CreateHostBuilder(args).Build().Run();  }  implementationb  public static IHostBuilder CreateHostBuilder(string[] args) =>  Host.CreateDefaultBuilder(args)  .ConfigureWebHostDefaults(webBuilder =>   1. بيكريت الhost بتاعي:   {  webBuilder.UseStartup<Startup>();   1. بيستخدم الconfiguration بتاعة الstartup   });  }  } |

1. Startup

* بيكون جواه 2functions:

|  |
| --- |
| public void ConfigureServices(IServiceCollection services)  {  services.AddControllers();   * في الmvc بتكونaddControllerWithView   services.AddSwaggerGen(c =>   * الservices بتاعة الswagger  * ملحوظه: لو مكنتش عامل Enable OpenApi السطر دم مكنش هيبقى موجود   {  c.SwaggerDoc("v1", new OpenApiInfo { Title = "Talbat.API", Version = "v1" });  });  } |

1. الfunction اللي بنحدد فيها الservices اللي هتشتغل بالdependency Injection



1. تاني function اللي بنحدد فيها الـ:

* Middleware/pipline بتاعة الkestrel اللي أي request هيمشي فيها جوا ال kesteral

|  |
| --- |
| public void Configure(IApplicationBuilder app, IWebHostEnvironment env)  {  if (env.IsDevelopment())  {   * في مرحلة الdev بيعرض:  1. الexception page   app.UseDeveloperExceptionPage();  app.UseSwagger();   1. الswagger عشان يعمل documentation للApi   app.UseSwaggerUI(c => c.SwaggerEndpoint("/swagger/v1/swagger.json", "Talbat.API v1"));  }  app.UseHttpsRedirection();   1. لو انا قررت اكلم أي endpoint/action ب:http protocol   الmiddleware دي هتعملي هتخلي http لhttps  app.UseRouting();   1. بتشوف الurl اللي انا كتبته بيـmatch مع ايroute     app.UseEndpoints(endpoints =>  endpoints.MapControllers();   1. في الـapi بيمسك كل controller ويعرف الroute بتاعه   (في الmvc كان لازن نعرف كل الـroutes )  {  });  } |

1. RestFull APIs: