25 ASP.NET MVC Interview questions covered in the video is as follows

4:11 Question 1 :- What is ASP .NET MVC Core ?

4:45 Question 2 :- Differentiate between ASP .NET Webforms vs MVC vs MVC core ?

9:52 Question 3 :- Explain MVC Architecture ?

11:04 Question 4 :- Why do we have wwwroot folder ?



11:29 Question 5 :- Explain the importance of appsettings.json ?

12:10 Question 6 :- How to read configurations from appsettings.json ?

14:07 Question 7 :- What is dependency injection ?

15:26 Question 8 :- Why do we need dependency injection ?

17:19 Question 9 :- How do we implement dependency injection ?

19:32 Question 10 :- What is the use of Middleware ?

21:02 Question 11 :- How to create a Middle ware ?

23:21 Question 12 :- What does startup.cs file do ?

23:21 Question 13 :- ConfigureServices vs Configure method ?

25:47 Question 14 :- Explain the different Ways of doing DI?

25:47 Question 15 :- Explain Scoped vs Transient vs Singleton ?

30:42 Question 16 :- What is Razor ?

31:41 Question 17 :- How to pass Model data to a View ?

32:19 Question 18 :- What is the use of Strongly typed views ?

33:43 Question 19 :- Explain the concept of ViewModel in MVC ?

35:03 Question 20 :- What is Kestrel Web Server ?

36:06 Question 21 :- Why Kestrel when we have IIS server ?

37:33 Question 22 :- What is the concept of Reverse proxy ?

39:17 Question 23 :- What are cookies ?

40:09 Question 24 :- What is the need session management ?

41:51 Question 25 :- What are the various ways of doing Session management in ASP.NET ?

ASP.NET MVC Interview Questions & Answers - Part 2

Question 26 :- What exactly is a session ?

Question 27 :- Explain "HTTP is a stateless protocol" ?

Question 28 :- What are various way of doing session management ?

Question 29 :- Are sessions enabled by default ?

Question 30 :- How to enable sessions in MVC core ?

Question 31 :- Are sessions variables shared(global) between users ?

Question 32 :- Do session variables use cookies ?

Question 33 :- What is a cookie ?

Question 34 :- Explain idle time out in sessions ?

Question 35 :- What does a Context means in HTTP ?

ASP.NET MVC Interview Questions & Answers - Part 3

Question 36 :- When should we use viewdata ?

Question 37 :- How to pass data from controller to view ?

Question 38 :- In same request can viewdata persist across actions ?

Question 39 :- ViewData vs ViewBag

Question 40 :- How does ViewBag work internally?

Question 41 :- Explain ViewModel?

Question 42 :- ViewBag vs ViewModel whats the best practice?

ASP.NET MVC Interview Questions & Answers - Part 4

Question 43 :- Explain tempdata ?

Question 44 :- Can tempdata persist across action redirects ?

Question 45 :- How is tempdata different from viewdata ?

Question 46 :- If tempdata is read is it available for next request ?

Question 47 :- How to persist tempdata ?

Question 48 :- What does Keep do in tempdata ?

Question 49 :- Explain Peek in tempdata ?

Question 50 :- How is tempdata different from session variables ?

Question 51 :- If i restart the server does tempdata, session stay ?

Question 52 :- Is tempdata private to a user ?

Question 53 :- ViewData vs ViewBag vs Tempdata vs Session variables

ASP.NET MVC Interview Questions & Answers - Part 5

Question 54 :- What is WebAPI ?

Question 55 :- What is the advantage of WebAPI ?

Question 56 :- Explain REST and Architectural constraints of REST?

Question 57 :- Can we use TCPIP protocol with Web API?

Question 58 :- How WebAPI different from MVC controller ?

Question 59 :- What is content negotiations in Web API ?

Question 60 :- WebAPI vs WCF ?

Question 61 :- WCF REST vs WebAPI REST ?

Question 62 :- How to return HTTP status codes ?

Question 63 :- For error which status code is returned ?

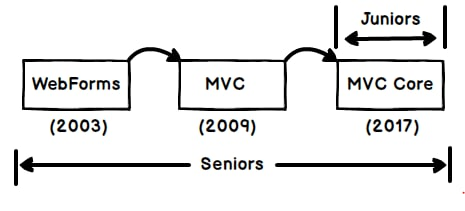
Question 64 :- How did you secure your web API ?

Question 65 :- How do current JS frameworks work with webAPI ?

Question 66 :- How did you implement JWT token security ?

**Lets plan before we start preparation**

When you talk about asp.net interviews the range of questions can be very narrow or it can be very wide. So do ensure you have proper preparation strategy at place. Remember ASP.NET has three versions of framework. So we had ASP.NET Webforms which came in 2003, followed by the ASP.NET MVC , then we had ASP.NET MVC Core.



Now if you are junior / fresher I would say stick to MVC Core and say you have just worked with MVC core , so that you do not get the questions of Webforms and MVC.

But if you are senior you do not have too many choices you will need to prepare through right from Webforms to MVC Core.

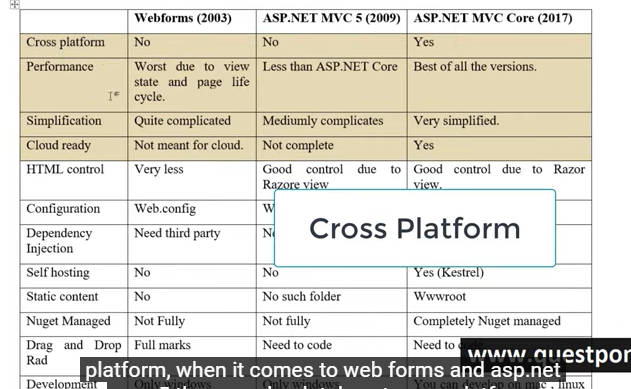
**What is ASP.NET MVC Core ?**

 ASP.NET MVC Core is an open source, **cross platform** framework to develop web applications. When you develop web applications using ASP.NET Core it can run on windows, Linux and mac.

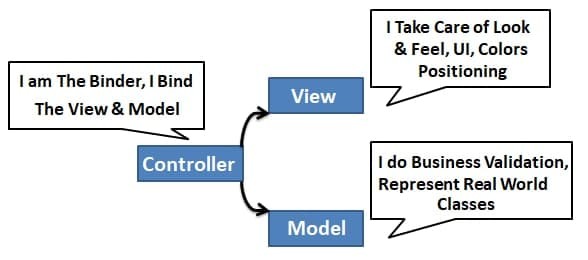
**Can you explain the difference between MVC core vs MVC 5 vs Webforms?**

 Now there are like 10 to 12 differences and if we try say all of them during interviews its not going to work. Always remember let the interviewer speak more and you as a candidate should speak to the point.

I have listed below all 12 points but would recommend to start with the first 4 points which are highlighted.



**Can you explain MVC Architecture?**



MVC is an architecture pattern where we divide project in to three layers.

The first layer is the view which has things like Color, UI positioning, CSS, input controls, buttons and so on.

Second layer is Model which is a simple class which has business validations.

Controller binds the model and the view.

So the first hit comes to the controller, it loads the model and binds the model data to the view.

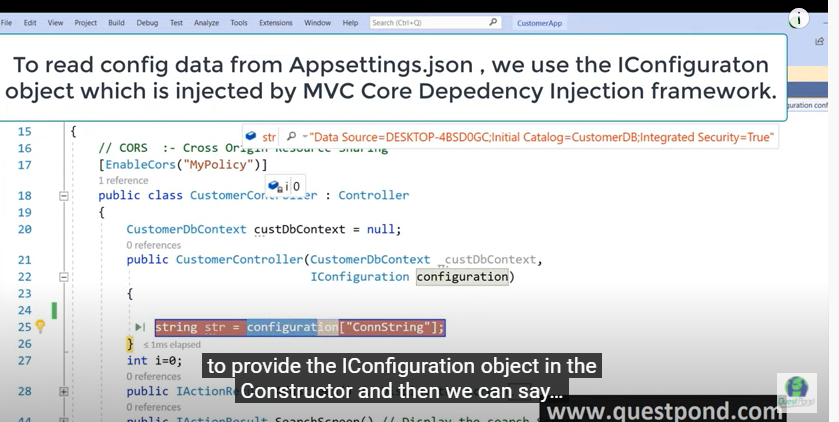
**Where do we store configuration in ASP.NET Core ?**



Configuration data is stored in a json file called as Appsettings.json .

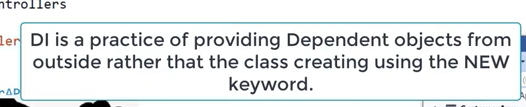
**How can we read configuration file in ASP.NET Core ?**

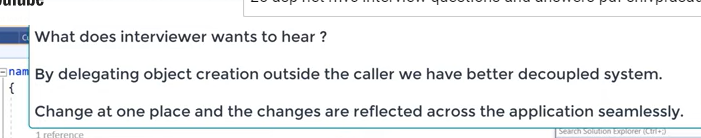
To read configuration from appsettings.json you need to use “IConfiguration” interface



**What is dependency injection?**

 Dependency injection is practice of providing dependent objects for a class from outside rather than the class creating it. So, in simple words the object creation process is isolated from the caller. This helps to create a proper decoupled system.





**Why do we need dependency injection?**

By using DI classes are decoupled from each other so you make changes at one place its reflected all over the places.

**How to implement dependency injection?**

To implement DI we need to add the service in “ConfigureServices” method which is in Startup.cs file.

**Explain the concept of middleware?**

Middleware helps to add pre-processing logic before the request is sent to the controller.

**How to implement middleware in MVC Core ?**

Middlewares are added in “Configure” method of Startup.cs file.

**Explain the importance of Startup.cs file in ASP.NET Core ?**

 Startup.cs file helps to configure Dependency injection and Middle wares.

**Explain the various way of doing DI in MVC?**

*This is an easy topic to understand but very difficult to explain to interviewer in simple words. So, I would suggest to go through the below explanation, understand it and create your simple one liner.*

 There are three ways of doing DI: - Scoped, Transient and Singleton.

**Transient**: - Creates new instances, every single time service is injected.

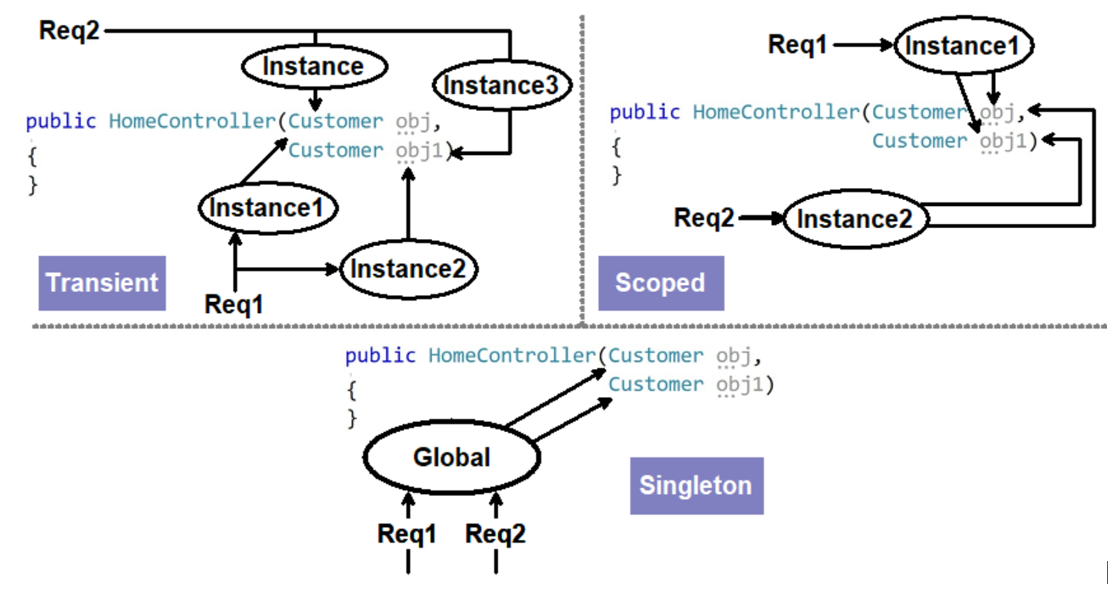
**Scoped**: - Creates new instances, once per request made to server.

**Singleton**: - Instantiates one global object for all requests coming to server from any user.

The below image shows the same visually. In transient for every dependency injection new instance will be injected. For example, in the below image for “obj” and “obj1” fresh new instances will be injected for every request.

In case of scoped for every request same instance will be injected for every dependency injection. So “obj” and “obj1” will have same instance injected.

In case of singleton one big global instance created. For all request , for all Dependency injection same instance will be injected.



**What are the various ways of doing session management in ASP.Net Core ?**

1. Session variables
2. Tempdata
3. ViewData / ViewBag

**ConfigureServices vs  Configure method ?**

 In ConfigureServices we add the objects to be dependency injected while in Configure method we specify middlewares

**Explain Razor pages ?**

Razor is a view engine in which you can write both C# and HTML code. C# code is rendered at the server side.

**How can we pass model data to the view ?**

You can pass by using viewdata or using strongly typed views.

**What is a strongly typed view ?**

Strongly typed views are Razor views where we get intellisense for the model used in the view.

**Explain the concept of viewModel ?**

View Model represents the data to be displayed on the view.

**Explain kestrel web server ?**

Kestrel is an open source default web server used by ASP.NET core application. Kestrel ships with ASP.NET Core setup and works as in-process web server to handle web request. Kestrel is cross platform it works on Linux , windows and Mac.

Every request of ASP.NET core goes through kestrel first.

**Why kestrel web server when we have IIS ?**

 ASP.NET core is meant to run cross platform. So, IIS as default server will not work , because IIS works only on Windows. Web server like NGINX , Apache and so on have their own way to running the application startup and ASP.NET core can not satisfy each one of them. So kestrel acts like a in-process web server take a request send to MVC core application, get response and sends it back to the main web server.

**Explain concept of reverse proxy ?**

 Reverse proxy is a concept where the server acts like a mediator. Kestrel works on reverse proxy concept it takes the request from the main web server like IIS / Apache and forwards its to MVC application and vice-versa.

**What are cookies ?**

Cookies are small text file where browser can store user related information.

**What is the need session management ?**

HTTP protocol is stateless to maintain states we need to use session management.

**What are the various ways of doing Session management in ASP.NET ?**

 There three ways of doing session management: -

1. ViewData / ViewBag
2. Session variables
3. Tempdata

**What exactly is a session ?**

 Session is user interaction which happens with a website over a period of time.Its interaction which happens right when browser opens and browser closes.

**Explain "HTTP is a stateless protocol"​ ?**

 HTTP does not remember states between request and response. User sends a request , server processes it and sends response. Now if the user comes again in the same session server treats its as a new request.

**What are various way of doing session management ?**

 There are three primary ways of doing session management Session variables , viewdata/view bag and Tempdata.

**Are sessions enabled by default ?**

 To keep ASP.NET light weight session variables are not enabled by default.

**How to enable sessions in MVC core ?**

 To enable session variables in Startup.cs we need to "AddSession"​ and "UseSession"​.

**Are sessions variables shared(global) between users ?**

 Session variables are created for a particular user. Data of session variable is not shared between users.

**Do session variables use cookies ?**

Yes,session variables use cookies.

**What is a cookie ?**

Cookies are small text files information which is stored in the end users computer.

**Explain idle time out in sessions ?**

Idle timeout is the time when user does not do any activity. So if user is inactive you would like to ensure session variables expire.

**What does a Context means in HTTP ?**

HttpContext holds information like session , request data and response information. It has all the data needed for a particular request and response.

**When should we use viewdata ?**

To pass data from controller to view.

**How to pass data from controller to view ?**

By using viewdata.

**In same request can viewdata persist across actions ?**

No it will not persiste. Viewdata scope only from action to view.

**ViewBag vs ViewData ?**

ViewBag is syntatic sugar over viewdata.

**How does ViewBag work internally ?**

ViewBag uses Dynamic variables.

**Explain viewModels ?**

They represent data for a view.

**ViewBag vs ViewModel Whats the best practice ?**

For passing data from action to view use viewmodel as ViewBag are dynamic and we can have mistakes.

**Explain tempdata ?**

Tempdata persists for the current and we can control do we want the state in the next request or not.

**Can tempdata persist across action redirects ?**

Yes

**How is tempdata different from viewdata ?**

Tempdata can persist from action to action ( i.e. action redirects) while viewdata is only from action to the view.

**If tempdata is read is it available for next request ?**

No

**How to persist tempdata ?**

We can persist tempdata by calling the keep method.

**What does Keep do in tempdata ?**

Keep method persist the data for the next request.

**Explain Peek in tempdata ?**

Peek = Keep + Read. It will read as well as keep the data.

**How is tempdata different from session variables ?**

Session variables persist right from browser open to browser close.While tempdata persist across action redirect and also we can decide using keep can we perist state for new request.

**If i restart the server does tempdata,session stay ?**

No, tempdata and session do not persist in server restarts.

**Is tempdata private to a user ?**

Yes , Tempdata is private to a user.

**ViewData vs ViewBag vs Tempdata vs Session variables**

 Session variables scope is from browser open to close.

Viewdata scope is only from action to view.

Viewdata does not persists in action or controller redirects.

Viewbag is a syntatic sugar over ViewData.

Tempdata maintains data in single request irrespective we have redirects or not.

Once tempdata is read state is not maintained across request.

If we call Keep method tempdata state is maintained across requests even if tempdata is read.

**What is WebAPI ?**

Web API controller delivers data and services easily by using HTTP REST services.

**What is the advantage of WebAPI ?**

 HTTP is a most used protocol so by exposing your service over HTTP any nature of clients can consume it , making it a true client server model. So your client can now be a mobile app or app made in JS frameworks like react/Angular or must be a simple browser , every one can integrate with WebAPI seamlessly.

**Explain REST and Architectural constraints of REST?**

 REST stands for representational state transfer. REST is a architectural style/principle where client and server talks in representations and states.

Some important principle of REST is Client Server,Statelessness,Unique URI,Manipulation happens through representation,

WebAPI helps to expose services / data using HTTP protocol.

**Can we use UDP/TCPIP protocol with Web API?**

No

**How WebAPI different from MVC controller ?**

In MVC Controller we can return RAZOR views but in WebAPI we can not return Views. WebAPI follows REST principles and does things like content negotiations.

**What is content negotiations in Web API ?**

Content negotiations looks at client accept type and depending on the client accept type it provides data format. So if in the accept client sends XML it will send XML , if it sends JSON it will send JSON.

**WebAPI vs WCF ?**

WCF was meant for SOA , XML is compulsory format and support any protocol.

WebAPI was meant for REST, No compulsion on Format and supports only HTTP protocol.

**WCF REST vs WebAPI REST ?**

WCF REST was creates to ensure that old legacy WCF can be REST enabled , While WebAPI was built from scratch to serve REST architecture.For new project use WebAPI. If you want to make legacy WCF compatible with REST you will used WCF REST.

**How to return HTTP status codes ?**

return StatusCode(StatusCodes.Status200OK,"Test"​);

**For error which status code is returned ?**

500 internal server error.

**How did you secure your web API ?**

By using JWT token.

**How do current JS frameworks work with webAPI ?**

Client side is built usin JS frameworks like Angular , React and they communicate to MVC using HTTP.

Sssss

Can we remove catchOr can we add multiple catch block

Final keyword

Dot net core method

Ref and out

Filters

Entity framework

Extantion method

method used in dot net core like startup.cs

Singleton

Raw number and ranking

How can we create model by use entityframework