

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

ASP.NET Interview Questions

1. What is ASP.NET?

Answer:

ASP.NET is a web framework developed by Microsoft for building web applications. It is a part of the .NET framework and supports languages like C#, VB.NET, and F#. ASP.NET allows developers to build dynamic, data-driven websites and applications efficiently.

2. What are the different types of ASP.NET applications?

Answer:

1. **Web Forms:** Event-driven programming model for rapid application development.
 2. **MVC (Model-View-Controller):** Provides a separation of concerns and more control over HTML.
 3. **Web API:** Framework for building HTTP services for browsers and mobile devices.
 4. **Blazor:** Enables building interactive web UIs using C# instead of JavaScript.
-

3. What are the main components of ASP.NET?

Answer:

1. **Server Controls:** Predefined controls like TextBox, Button, etc.
 2. **HTTP Handlers:** Process incoming HTTP requests.
 3. **HTTP Modules:** Inspect and modify HTTP requests and responses.
 4. **Page Lifecycle:** Key stages include Initialization, Load, Validation, Rendering, and Unload.
-

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

4. What is the ASP.NET Page Life Cycle?

Answer:

The life cycle stages of an ASP.NET page include:

1. **Page Request**
 2. **Start**
 3. **Initialization**
 4. **Load**
 5. **PostBack Event Handling**
 6. **Rendering**
 7. **Unload**
-

5. What is the difference between ASP.NET Web Forms and ASP.NET MVC?

Answer:

Feature	Web Forms	MVC
Architecture	Event-driven	Model-View-Controller
Control	Server-side	Full control of HTML
Testability	Limited	High
Separation of Concerns	Less	High

.NET Core MVC Questions

6. What is .NET Core MVC?

Answer:

.NET Core MVC is a lightweight, open-source framework for building cross-platform web applications. It follows the **Model-View-Controller (MVC)** design pattern and provides flexibility for developers to create robust web solutions.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

7. What is the role of Middleware in ASP.NET Core?

Answer:

Middleware are components that handle HTTP requests and responses in ASP.NET Core. Examples include **authentication**, **logging**, **routing**, and serving static files.

8. What is Dependency Injection (DI) in .NET Core?

Answer:

Dependency Injection is a design pattern used to inject dependencies into a class, rather than creating them inside the class. In .NET Core, DI is built-in, and services are registered in the **Startup.cs** file via the **IServiceCollection**.

9. What is the difference between IActionResult and ActionResult in ASP.NET Core MVC?

Answer:

- **IActionResult**: It is a general interface for any result type.
 - **ActionResult**: It is a class that provides built-in result types like **ViewResult**, **JsonResult**, etc. Both are used to return responses from a controller action.
-

10. What are Tag Helpers in ASP.NET Core MVC?

Answer:

Tag Helpers are server-side code that helps create and render HTML elements in Razor views. For example:

html

Copy code

```
<form asp-action="Submit" asp-controller="Home">  
</form>
```

C# Interview Questions

11. What is C#?

Answer:

C# is an object-oriented, type-safe, and managed programming language developed by Microsoft as part of the .NET ecosystem. It is widely used for developing desktop, web, and mobile applications.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

12. What is the difference between ref and out parameters in C#?

Answer:

- **ref:** Requires the variable to be initialized before passing it.
 - **out:** Does not require the variable to be initialized; it must be assigned a value within the method.
-

13. What are the access modifiers in C#?

Answer:

1. **Public:** Accessible from anywhere.
 2. **Private:** Accessible only within the containing class.
 3. **Protected:** Accessible within the class and its derived classes.
 4. **Internal:** Accessible within the same assembly.
 5. **Protected Internal:** Accessible within the same assembly or a derived class.
-

14. What is the difference between abstract class and interface?

Answer:

Feature	Abstract Class	Interface
Inheritance	Single inheritance	Multiple inheritance
Implementation	Can have implemented methods	Cannot have implementation
Fields	Can have fields	Cannot have fields

15. Explain the difference between async and await in C#.

Answer:

- **async:** Marks a method as asynchronous.
 - **await:** Pauses the execution of an async method until the awaited task is completed.
-

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

16. What is the difference between Array and ArrayList?

Answer:

- **Array:** Fixed size and type-safe.
 - **ArrayList:** Dynamic size and can store any type of data (non-generic).
-

17. What is the difference between String and StringBuilder in C#?

Answer:

- **String:** Immutable; every modification creates a new object.
 - **StringBuilder:** Mutable; allows efficient modifications without creating new objects.
-

18. What is LINQ in C#?

Answer:

LINQ (Language Integrated Query) is a set of methods for querying data in a type-safe and readable manner. Example:

csharp

Copy code

```
var result = list.Where(x => x.Age > 18).ToList();
```

19. What is Garbage Collection in C#?

Answer:

Garbage Collection (GC) is an automated memory management process in C#. It automatically frees up memory by removing unused objects from the heap.

20. What is the difference between Task and Thread in C#?

Answer:

- **Thread:** Represents an independent path of execution.
 - **Task:** High-level abstraction over threads that supports asynchronous programming.
-

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

Additional ASP.NET Questions

21. What is ViewState in ASP.NET?

Answer:

ViewState is a mechanism to preserve page and control values between postbacks. It uses a hidden field to store the state information in Base64-encoded format.

22. What are HTTP Handlers in ASP.NET?

Answer:

HTTP Handlers are responsible for processing incoming HTTP requests. For example, .ashx files are handled by custom HTTP handlers.

23. What is the difference between `Server.Transfer` and `Response.Redirect`?

Answer:

- **Server.Transfer:** Transfers the request on the server without a round trip to the client.
 - **Response.Redirect:** Performs a client-side redirection, causing a new HTTP request.
-

24. What is the role of the `web.config` file in ASP.NET?

Answer:

The `web.config` file is used to configure application settings, connection strings, session state, error handling, and security configurations.

25. What is the use of the `Global.asax` file?

Answer:

The `Global.asax` file defines application-level events like `Application_Start`, `Application_End`, `Session_Start`, and `Session_End`.

26. What is the purpose of the `HttpContext` object?

Answer:

The `HttpContext` object provides access to all HTTP-specific information about the request, such as cookies, session state, and server variables.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

27. What is a Web API, and how is it different from WCF?

Answer:

- **Web API:** Used for building RESTful services over HTTP.
 - **WCF:** Supports multiple protocols like HTTP, TCP, and MSMQ, making it more versatile.
-

28. What is Output Caching in ASP.NET?

Answer:

Output caching stores the generated output of a page or control and reuses it for subsequent requests to improve performance.

29. What is the difference between synchronous and asynchronous requests in ASP.NET?

Answer:

- **Synchronous:** Blocks the thread until the operation is complete.
 - **Asynchronous:** Allows other operations to run while waiting for a response.
-

Additional .NET Core MVC Questions

30. What is Razor in ASP.NET Core MVC?

Answer:

Razor is a markup syntax that combines HTML and C# for creating dynamic web pages. It uses the .cshtml file extension.

31. What are Filters in ASP.NET Core MVC?

Answer:

Filters allow code to be executed before or after certain stages of the request processing pipeline. Examples include **Authorization**, **Resource**, **Action**, and **Exception** filters.

32. What is Routing in ASP.NET Core?

Answer:

Routing maps incoming requests to specific controller actions. It is defined in the Program.cs or Startup.cs file using endpoints.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

33. What is Model Binding in ASP.NET Core?

Answer:

Model Binding automatically maps HTTP request data to action method parameters or model objects.

34. What are View Components in ASP.NET Core?

Answer:

View Components are reusable components that render a chunk of HTML. They are similar to partial views but with added logic.

35. What is the difference between Partial View and Layout in ASP.NET Core?

Answer:

- **Partial View:** A reusable view fragment.
 - **Layout:** A template used to define the overall structure of the views.
-

36. What is SignalR in ASP.NET Core?

Answer:

SignalR is a library for building real-time web applications, enabling server-side code to send asynchronous notifications to clients.

37. What is Kestrel in ASP.NET Core?

Answer:

Kestrel is the cross-platform, lightweight web server used by ASP.NET Core applications.

38. What is the role of the appsettings.json file?

Answer:

The appsettings.json file is used to store configuration settings, such as connection strings and application-specific settings, in a structured JSON format.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

39. What is Middleware Pipeline in ASP.NET Core?

Answer:

The Middleware Pipeline defines the sequence of middleware components that process HTTP requests and responses.

Additional C# Questions

40. What is the difference between abstract and virtual methods in C#?

Answer:

- **Abstract:** Must be implemented in derived classes.
 - **Virtual:** Can be overridden but provides a default implementation.
-

41. What is polymorphism in C#?

Answer:

Polymorphism allows methods to have multiple forms, such as method overriding (runtime polymorphism) and method overloading (compile-time polymorphism).

42. What is the difference between struct and class in C#?

Answer:

- **Struct:** Value type, stored on the stack.
 - **Class:** Reference type, stored on the heap.
-

43. What are delegates in C#?

Answer:

Delegates are type-safe pointers to methods, allowing methods to be passed as arguments.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

44. What are Generics in C#?

Answer:

Generics enable type-safe data structures and methods without specifying the data type during compile-time. Example:

csharp

Copy code

```
List<int> numbers = new List<int>();
```

45. What are extension methods in C#?

Answer:

Extension methods add new functionality to existing types without modifying their source code.

Example:

csharp

Copy code

```
public static string ToUpperCase(this string str) {  
    return str.ToUpper();  
}
```

46. What are the types of assemblies in C#?

Answer:

1. **Private Assembly:** Used by a single application.
 2. **Shared Assembly:** Stored in the Global Assembly Cache (GAC).
 3. **Satellite Assembly:** Used for localization.
-

47. What is async/await in C#?

Answer:

async/await is used for asynchronous programming. It allows methods to run asynchronously without blocking the main thread.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

48. What is Reflection in C#?

Answer:

Reflection provides the ability to inspect and interact with metadata and assemblies during runtime.

49. What is a `NullReferenceException` in C#?

Answer:

This exception occurs when attempting to use an object that has not been instantiated.

50. What are tuples in C#?

Answer:

Tuples are data structures that allow storing multiple items of different types. Example:

csharp

Copy code

```
var person = (Name: "John", Age: 30);
```

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

ASP.NET Web API Questions

1. What is Web API?

Answer:

Web API is a framework for building RESTful services over HTTP. It is lightweight and ideal for developing services for devices like mobile phones, tablets, or browsers.

2. What are the main features of Web API?

Answer:

- Supports CRUD operations using HTTP methods (GET, POST, PUT, DELETE).
 - Open source and supports OData.
 - Can be self-hosted or run in IIS.
-

3. What is the difference between WCF and Web API?

Answer:

Feature	WCF	Web API
Protocol Support	Supports multiple protocols	HTTP only
Hosting	IIS, Windows Services	IIS, Self-hosting

4. How does Web API support routing?

Answer:

Web API uses **attribute-based routing** or **convention-based routing** to map URLs to actions.

5. What is MediaTypeFormatter in Web API?

Answer:

MediaTypeFormatter is responsible for serializing and deserializing data into a format like JSON or XML.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

6. What are Filters in Web API?

Answer:

Filters allow code execution at specific stages of request processing. Types include:

- **Authorization Filters**
 - **Action Filters**
 - **Exception Filters**
-

7. What is the use of the HttpResponseMessage class?

Answer:

It represents an HTTP response message, allowing you to customize the response status, headers, and body.

8. What are Action Results in Web API?

Answer:

Action Results represent the response from a controller action. Examples:

- **Ok()** for 200 status.
 - **NotFound()** for 404 status.
 - **BadRequest()** for 400 status.
-

9. What is Dependency Injection in Web API?

Answer:

Dependency Injection allows you to inject services into controllers, making them more testable and maintainable.

10. What is CORS in Web API?

Answer:

CORS (Cross-Origin Resource Sharing) enables secure communication between a Web API and applications from different domains.

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

11. How can you secure a Web API?

Answer:

- Use **HTTPS**.
 - Implement **authentication** (e.g., JWT, OAuth).
 - Use **Authorization filters** to control access.
-

12. What is the difference between IHttpActionResult and HttpResponseMessage?

Answer:

- **IHttpActionResult**: Higher-level abstraction for action results.
 - **HttpResponseMessage**: Lower-level approach for crafting custom responses.
-

13. How is Web API different from MVC?

Answer:

Feature	MVC	Web API
Use Case	Web Applications	RESTful Services
Content Type	Mostly HTML	JSON, XML, etc.

14. What is content negotiation in Web API?

Answer:

Content negotiation determines the response format (e.g., JSON, XML) based on client preferences.

15. How can you implement versioning in Web API?

Answer:

Versioning can be implemented using:

- URI (e.g., /v1/endpoint).
 - Query String (e.g., ?version=1).
 - Custom Headers.
-

ASP.NET, .NET Core MVC, .NET Web API & C# Interview Questions

16. What is OData in Web API?

Answer:

OData (Open Data Protocol) is a standard for building and consuming RESTful APIs with query capabilities like filtering, sorting, and pagination.

17. What is the role of HttpClient in Web API?

Answer:

HttpClient is used to send HTTP requests and receive responses in client applications.

18. What is Swagger in Web API?

Answer:

Swagger (now known as OpenAPI) provides tools for documenting APIs, generating client SDKs, and testing endpoints interactively.

19. What is the role ofFromBody and FromUri attributes?

Answer:

- **FromBody:** Maps data from the request body to parameters.
 - **FromUri:** Maps data from the query string or route to parameters.
-

20. How do you handle exceptions in Web API?

Answer:

- Use **Exception Filters**.
- Implement **try-catch** blocks.
- Log errors using tools like **Serilog** or **NLog**.