



Dot Net Interview Questions and Answers

Introduction

DOT NET interview questions typically focus on C# programming, the .NET Framework, ASP.NET, and concepts like garbage collection, assemblies, and exception handling. Candidates may also be tested on OOP principles, LINQ, and the web application lifecycle. These questions help evaluate both foundational knowledge and practical experience in .NET development. Struggling with Dot Net Concept? Check out our deep-dive [Dot Net beginners tutorial](#) for a step-by-step walkthrough.

List of Dot Net Interview Questions for Freshers

1. What languages can be used in the Dot Net framework?
2. Define Dot Net Core.
3. What are the major components of the DotNet Framework?
4. Define JIT.
5. Differentiate int and int32.

6. What is role-based security in .Net?
7. List out the order of events in a page's life cycle.
8. Define Caching.
9. Define MIME in .Net.
10. Define Universal Windows Platform (UWP) apps in .Net Core.

Find here our [Dot Net Course Syllabus](#) to get started.

Dot Net Interview Questions and Answers for Freshers

1. What languages can be used in the Dot Net framework?

In the Dot Net Framework, we can use various programming languages like C#, Cobol, VB, Perl, and so on for developing web applications and web services. We can develop desktop applications, websites, and support services in the .Net framework.

2. Define Dot Net Core.

The Dot Net Core is the latest version of the Dot Net framework and it is a general-purpose, open-source platform. It is cross-platform and can be used to develop different kinds of applications like mobile, IoT, web, machine learning, games, and microservices.

3. What are the major components of the DotNet Framework?

The major components of the Dot Net framework are as follows:

- CLR (Common Language Runtime) as an executive engine
- FCL (Framework Class Library) has pre-defined methods
- BCL (Base Class Library) has a huge collection of libraries and functions.
- CTS (Common Type System), which specifies a standard
- CLS (Common Language Specification) defines rules and regulations.

4. Define JIT.

JIT is the just-in-time compiler that converts the intermediate code into native language code during the execution of Dot Net programs.

5. Differentiate int and int32.

Int32 is a type provided by the .NET Framework class, while int is an alias name for int32 in the C# programming language. There is no other difference between int and int32.

**Take your Knowledge
Test Report**

Check your Score



6. What is role-based security in .Net?

Role-based security is used to implement security features in .Net based on the rules allocated to the users of the organization. The authorization of users assigned to their roles in the organization. Likewise, Windows has role-based access for users, guests, and administrators.

7. List out the order of events in a page's life cycle.

The order of events for a successful page rendering contains the following eight events:

- Page_PreInit
- Page_Init
- Page_InitComplete
- Page_PreLoad
- Page_Load
- Page_LoadComplete
- Page_PreRender
- Render

8. Define Caching.

Caching is used to store the data temporarily in memory for easy access by an application instead of searching for it in its original location. It improves the speed and efficiency of an application. There are three types of caching: page caching, data caching, and fragment caching.

9. Define MIME in .Net.

MIME is the abbreviation for Multipurpose Internet Mail Extensions. It is an extension of the e-mail protocol that lets users utilize the protocol for exchanging files across emails easily. Servers insert the MIME header at the beginning of the web transmission to define the MIME transactions.

10. Define Universal Windows Platform (UWP) apps in .Net Core.

Universal Windows Platform (UWP) is one of the methods used for creating client applications for Windows. It will make use of the WinRT API to provide a powerful user interface (UI) and advanced asynchronous features for devices with internet connections.

Want to explore more? Join us to learn the best [Dot Net course in Chennai](#).

List of Dot Net Interview Questions for Experienced

1. Explain the purpose of webHostBuilder()
2. What are the benefits of AOT compilation?
3. Differentiate between value type and reference type.
4. What are the various types of assembly?
5. Explain the garbage collector.
6. Explain MDI and SDI
7. What are the differences between .Net Core and Mono?
8. Differentiate between custom control and user control.
9. Write a code to send an email from an ASP.NET application.
10. What are the differences between ASP and ASP.NET?

Dot Net Technical Interview Questions and Answers for Experienced

1. Explain the purpose of webHostBuilder()

WebHostBuilder() is a function used for HTTP pipeline creation with webHostBuilder.use() method that is in webHostBuilder.Build() using the builder pattern. Microsoft.AspNetCore.Hosting namespaces provide this.

The purpose of the build() method is to build necessary services and a Microsoft.AspNetCore.Hosting.IWebHost is for hosting a web application.

2. What are the benefits of AOT compilation?

AOT compilation, also known as explicit compilation, has the following benefits:

- Application size is smaller as the compiler itself isn't shipped and unused features can be deleted.
- Template the parse errors that were identified previously during the execution time.
- Security is high and it doesn't require dynamically evaluating templates.
- Rendering of a component is faster and more efficient as it has pre-compiled templates.
- For AOT compilation, the tools are needed to accomplish it automatically in the development process.

3. Differentiate between value type and reference type.

The major differences between value type and reference type are as follows:

- A value type holds the *actual data* directly within the memory location, while a reference type holds *a pointer that contains the address* of another memory location that has actual data.
- A value type stores its contents in the *stack memory*, while a reference type stores its contents in the *heap memory*.
- It will *copy the value directly* when assigning a value type variable to another variable, while it *doesn't copy the value* when assigning a reference type variable to another, but it creates a second copy of the reference.

- Examples of value types are *predefined data types, enums, and structures*. Examples of reference types are *classes, objects, indices, interfaces, and arrays*.

4. What are the various types of assembly?

Assemblies are divided into two major types: private assemblies and shared or public assemblies.

- **Private Assembly:** It is accessible only through the application. They can't access the private assembly unless they copy it. Users need to copy this private assembly separately to all the application folders wherever they want to use that assembly.
- **Public or Shared Assembly:** Multiple apps can share the shared assembly, also known as the public assembly. It doesn't require copying it separately into all application folders. Only one copy is enough at the system level and we can use the same copy for multiple applications. It will be installed in the Global Assembly Cache (GAC).

5. Explain the garbage collector.

A garbage collector is used to free up the unused code objects in the memory and the memory heap is partitioned into 3 generations.

- Generation 0 holds short-lived objects, Generation 1 stores medium-lived objects, and Generation 2 is for long-lived objects.
- Garbage collection involves searching the managed heap's generations for things that the application is no longer using.
- It performs the required operations to reclaim the memory. It must perform a collection to free up some memory space.

Take your Knowledge
Test Report

Check your Score



6. Explain MDI and SDI

MDI stands for *Multiple Document Interface*, which allows the developer to open multiple windows. It will have one parent window with many child windows. The components are shared from the parent window, like the Menubar and the toolbar.

SDI stands for *Single Document Interface*, which opens each document in a separate window, and each window has its components like a toolbar, menubar, etc and it is not constrained to the parent window.

7. What are the differences between .Net Core and Mono?

- Dot Net Core is a subset implementation of the .Net Framework developed by Microsoft. Mono is the complete implementation of the .Net Framework for Linux, iOS, and Android, developed by Xamarin.
- Dot Net Core allows developers to build web applications and console applications. Mono allows the developers to build various application types available in the .NET Framework, including mobile apps, GUI-enabled desktop apps, etc.
- Dot Net Core doesn't have the in-built capacity to be compiled into WebAssembly-compatible packages. Mono has an in-built capacity to be compiled into WebAssembly-compatible packages.
- Dot Net Core is not for gaming app development. The developer can only build a text-based adventure or a relatively basic browser-based game using .Net Core. Mono is intended for game development, and the games will be developed using the Unity gaming engine that supports Mono.

8. Differentiate between custom control and user control.

- Custom Control inherits from the System.Windows.Controls.Control Class, while User Control inherits from the system.Windows.Controls.UserControls class.
- Custom Control is designed for single-application scenarios, while User Control is designed for multi-application scenarios.
- Custom Control introduces redundancy and maintenance problems if used in more than one application, while User Control will be distributed without facing any problems associated with redundancy and maintenance.

- Custom Control contains loosely coupled control concerning code and UI (User Interface), while User Control is a tightly coupled control concerning code and UI.

9. Write a code to send an email from an ASP.NET application.

The following code is used to send emails from the ASP.Net application:

```
MailMessage mailMess = new MailMessage();  
  
mailMess.From = "person1@gmail.com";  
  
mailMess.To = "person2@gmail.com";  
  
mailMess.Subject = "Sample Email";  
  
mailMess.Body = "Verify the sample mail.";  
  
SmtpMail.SmtpServer = "localhost";  
  
SmtpMail.Send (mailMess);
```

MailMessage and SmtpMail are predefined classes that are in System.Web.Mail namespace.

10. What are the differences between ASP and ASP.NET?

- ASP stands for Active Server Page, which uses VBScript, and the code of ASP is interpreted during execution time. Whereas ASP.NET uses .Net languages such as VB.Net or C# that are compiled into MSIL.
- ASP is the server-side technology of Microsoft that is used to create web pages. ASP.NET is also developed by Microsoft for developing dynamic web applications.
- ASP is partially based on OOPs, but ASP.Net is completely based on Object-Oriented programming.
- ASP has no in-built support for XML, whereas ASP.Net has full XML support for easy data exchange.
- ASP uses ADO technology, and ASP.NET utilizes ADO.Net technology for connecting to and working with databases.

Conclusion

Explore a wide range of opportunities in DotNet development and prepare with this article that contains DotNet interview questions and answers that will be helpful to ace the interview confidently. Mastering .NET requires blending C# expertise with Core architecture and cloud integration. Elevate your coding career with our [software training institute in Chennai](#) with the .NET Boot Camp.