Page No.
Date

Dot Net Interview

- TutorialRide.com

Q. What is .Net framework?

A. is a s/w development platform for building various app^s.

It contains inbuilt func. in form of class, library & APIs that are used to develop app^s for web, windows, phone.

Q2 Major components of .Net framework?

→ Common Language Runtime (CLR)

↳ it manages the execution of .Net programs. It works as a layer b/w OS & app^s written in .Net Languages. It makes the development process easier by praiding services like Memory Management, Thread execution, Type Safety etc.

→ .Net framework class Library (FCL)

↳ It manages includes collection of classes, namespaces, interfaces & value types that are used for .Net applications to access common functionality.

Q.3 What is .Net Assembly?

A. An assembly is a smallest unit of deployment in a Microsoft .Net framework application. It can be implemented as .exe & .dll file. It provides all required execution information to CLR. The .Net framework maintains multiple versions of app. in sys. via assembly.

→ Assembly has 4 diff. components:

↳ Assembly manifest

↳ contains info. about versions of assembly

↳ Type meta data

↳ it contains binary info. of the program.

↳ MSIL source code

↳ Microsoft Intermediate Lang. is a CPU independent set of inst.s that can be efficiently converted to native code.

↳ Resources

↳ All other related files like icon, images etc.

Q.4 Diff. types of Assemblies?

A. ↳ Private assembly

↳ is used by a single application. It's kept in a local folder in which the client app. has been installed.

↳ Public or Shared Assembly

↳ is allowed to be shared by multiple app's. A SA must reside in Global Assembly cache (GA with a strong name assigned to it.

Q.5 Describe GAC in the .Net framework.

A. GAC stands for global assembly cache. It's an area of memory reserved to store assemblies in .Net app that are running on a machine. It stores those assemblies which will be shared by many app's.

Q.6 What is a Satelite assembly?

A. provides an assembly the multi-lingual support. They include localized resources for an app.

Using satellite assembly, you can include resources for diff. lang. in diff. assembly.

Q7. Explain diff. b/w dataset and data reader.

A. Dataset is a disconnect arch. We dont require conn. while working with Dataset.

DataReader is a connection oriented service, the data is ava. as long as a connection exists.

Dataset is an in-memory representation of a collection of Db objects. It's like a small Db because it stores the schema and data in the app. memory area.

Data Reader provides fwd-only & read-only access to data from data source. It's like a fwd only record set.

Dataset can be serialized & represented in XML, so it can easily pass around to other tiers.

Data Reader can't be serialized.

Page No.
Date

- Dataset is updatable, we can make changes to the data & send those changes back to the data source.

Dataset is best choice when we want manipulation of Data.

→ Data Reader best choice when we require no manipulation.

Q8. Web.Config & Machine.Config

- Machine.Config

↳ is the master config file on your sys. & is default conf. for all app's running on a machine. This is automatically installed when you install Visual Studio .Net. We can have only one Machine.config file on a server.

- Web.Config

↳ is an app. level configuration file which is automatically created when you create an ASP.Net web app. It stores config. data in XML. Generally one website = one WC but can have multiple to manage settings at various level.

Q9. What do you understand by side-side execution of assembly?

A. Allows assemblie to co-exists on same PC.

Q10. What is MVC

A. Model, view, Controller. MVC is a s/w arch-pattern which separates the representation and user interaction.

Model

↳ rep. the real world object & provides data to the view. It's nothing but a set of classes that describes business logic

View

↳ is responsible for look & feel. It represents UI components like HTML, CSS, jQuery etc.

Controller

↳ is resp. for taking end user req.s via View & loading appropiate Model & View

Q11. Explain the situation you will use a Web Service & Remoting in projects.

A. → Both .Net Web Service & .Net Remothing are solutions for dist. sol.

- .Net Web Services

↳ allows communication b/w app's over the WWW. It's a simple programming model with broad cross-platform reach. It supports heterogeneous environment which means client & remote object can build in any platform.

Web services should be used if application demands communication over a public n/w & require to work across multiple platforms.

- .Net Remoting

↳ allows objects to interact with eachother across application domains. It supports homogenous env. that requires the client be build using .NET.

It allows high speed communication b/w a client application & components in a binary format.

.Net Remoting good choice when we want to establish communication b/w proprietary .NET components, usually over an internet n/w.

Q12. What is a garbage collection? Is it possible to force garbage collection to run?

A. Garbage collection is a .Net feature that manages the

memory for applications by allocating & releasing memory automatically.

When the garbage collector performs a collection, it identifies objects in the managed heap that are no longer being used by the application & performs the necessary operations to reclaim their memory.

System.GC.Collect() → to force GC.

Q13. Why do we need ~~to~~ serialization in .NET?

A. It's the process of converting an object into a stream of bytes to persist the object's state into the memory, a database, or a file. The reverse of serialization is deserialization.

Serialization facilitates the transmission of an obj. over a n/w.

Remoting app^s heavily depends on serialization & deserialization.

Q14. Managed & Un Managed code?

A. Managed Code

↳ is a code whose execution is managed by

Page No.
Date

Common Language Run-time. This code runs inside the CLR.
Hence, it's necessary to install the .Net framework in order to execute the managed code.

- Un-manage code

↳ is any code that doesn't depends on CLR for execution and is not under the control of CLR. It means it is developed by other lang. independent of .NET framework.
It uses its own runtime environment for compiling & execution.