

## Probable Viva questions and answers

### **AJAX – Asynchronous Javascript and XML**

#### **What is the core object of AJAX technology?**

**XMLHttpRequest** is the core object in AJAX technology regardless of any implementation. **XMLHttpRequest** object is used to exchange data with a server seamlessly. Basically JavaScript uses this Object to exchange XML as well as text data between client and server. An AJAX implementation uses this object and communicates with server but it doesn't require the complete page to be refreshed.

#### **Difference between Synchronous and Asynchronous Postback?**

In Synchronous postback, complete web page is sent to server and in return rendering the output (i.e. complete page), whereas in case of Asynchronous postback, partial page goes to the server and renders only partial (required) part of the page.

#### **What are the basic controls in ASP.NET AJAX?**

Following controls can be considered as core AJAX controls in ASP.NET.

- **ScriptManager**
- **ScriptManagerProxy**
- **UpdatePanel**
- **UpdateProgress**
- **Timer**

Later, more controls are added to ASP.NET AJAX library, e.g., Script Loader, Client Data Context, Client Data Access, jQuery Integration, etc.

#### **What is a ScriptManager in ASP.NET AJAX?**

In order to use AJAX functionality on a web page, we add a **ScriptManager** control to the page in most of the scenarios, because **ScriptManager** control registers AJAX library scripts to that particular web page. We can have only one **ScriptManager** per page.

#### **Difference between ScriptManager and ScriptManagerProxy**

##### **Script Manager :-**

1. You can only have one script manager per page
2. A server control that makes script resources available to the browser, including the Microsoft AJAX Library and the functionality that enables partial-page rendering.

### Script manager proxy :-

1. You can have multiple proxies.
2. A server control that enables nested components to add script and service references if the page already contains a ScriptManager control.

### What is "AsyncPostBackTimeout" Property?

1. AsyncPostBackTimeout Property specifies the time, in seconds, before an asynchronous postback timeout occurs if no response is received from the server.
2. The default value of this property is 90 (seconds).
3. We can also set the user defined error message using "asyncpostbackerrormessage" property (as shown in the code above) for the time out.
4. For example:

```
<asp:ScriptManager ID="scriptManager1" runat="server" AsyncPostBackErrorMessage="
We can not serve your request at this moment. Please try
later." AsyncPostBackTimeout="120">
</asp:ScriptManager>
```

### How to use Script Manager's "AsyncPostBackTimeout" at runtime?

1. If you have a Script Manager on the Master page and need to exceed AsyncPostBackTimeout only on specific page, then
2. Add this code line to the "Page\_Load" event of your page.
3. This will set the AsyncPostBackTimeout to 2 minutes.
4. For example:

```
ScriptManager _scriptMan = ScriptManager.GetCurrent(this);
_scriptMan.AsyncPostBackTimeout = 120;
```

OR

```
ScriptManager.GetCurrent(this).AsyncPostBackTimeout = 120;
```

### Can we nest the **UpdatePanel** controls?

Yes, we can nest the **UpdatePanel** control.

### Importance of UpdatePanel control

The UpdatePanel control is probably the most important control in the ASP.NET AJAX package. It will AJAX'ify controls contained within it, allowing partial rendering of the area.

The `<asp:UpdatePanel>` tag has two childtags - the `ContentTemplate` and the `Triggers` tags.

The `ContentTemplate` tag is required, since it holds the content of the panel. The content can be anything that you would normally put on your page, from literal text to web controls.

The `Triggers` tag allows you to define certain triggers which will make the panel update it's content.

### #Timer Control

To refresh `UpdatePanel` controls at a timed interval.

### Using a Timer Control Inside an UpdatePanel Control

When the `Timer` control is included inside an `UpdatePanel` control, the `Timer` control automatically works as a trigger for the `UpdatePanel` control.

### Using a Timer Control Outside an UpdatePanel Control

When the `Timer` control is outside an `UpdatePanel` control, you must explicitly define the `Timer` control as a trigger for the `UpdatePanel` control to be updated.

```
<asp:ScriptManager runat="server" id="ScriptManager1" />
<asp:Timer ID="Timer1" runat="server" Interval="120000"
  OnTick="Timer1_Tick">
</asp:Timer>
<asp:UpdatePanel ID="UpdatePanel1" runat="server">
  <Triggers>
    <asp:AsyncPostBackTrigger ControlID="Timer1"
      EventName="Tick" />
  </Triggers>
  <ContentTemplate>
    <asp:Label ID="Label1" runat="server" ></asp:Label>
  </ContentTemplate>
</asp:UpdatePanel>
```

### #Limitations of AJAX

- If javascript is disabled AJAX will not work.
- The page cannot be bookmarked

## WCF – Windows Communication Foundation

### Features of WCF

Windows Communication Foundation (WCF) is a secure, reliable, and scalable messaging platform for the .NET Framework 3.0,

- Service Orientation
- Interoperability

- Multiple Message Patterns
- Service Metadata
- Data Contracts
- Security
- Multiple Transports and Encodings
- Reliable and Queued Messages
- Durable Messages
- Transactions
- AJAX and REST Support
- Extensibility

## What are Contracts in WCF?

The main use of contracts is to allow the client and services agree as to the types of operations and structures they will use during the communication process.

## How many Contracts are in WCF?

Service Contract

Operation Contract

Data Contract

Message Contract

Fault Contract

## What are the properties of WCF?

Each endpoint consists of three properties:

- An **A**ddress that indicates where the endpoint can be found.
- A **B**inding that specifies how a client can communicate with the endpoint.
- A **C**ontract that identifies the operations available.

An Address is a unique Uniform Resource Locator (URI) that identifies the location of the service. It defines the network address for sending and receiving the messages.

Binding describes how client will communicate with service.

A binding has several characteristics, including the following:

- **Transport** - Defines the base protocol to be used like HTTP, Named Pipes, TCP, and MSMQ are some type of protocols.

- **Encoding (Optional)** - Three types of encoding are available-Text, Binary, or Message Transmission Optimization Mechanism (MTOM). MTOM is an interoperable message format that allows the effective transmission of attachments or large messages (greater than 64K).
- **Protocol(Optional)** - Defines information to be used in the binding such as Security, transaction or reliable messaging capability.

List of Protocols:- **BasicHttpBinding, WSHttpBinding, WSDualHttpBinding, NetTcpBinding** etc.

The contract outlines what functionality the endpoint exposes to the client.

Eg:

**<endpoint**

```

address="http://localhost:8090/MyService/MathService.svc"
contract="IMathService"

```

```

binding="wsHttpBinding"/>

```

**# Different hosting procedures of WCF**

- ▶ **IIS** - IIS stands for Internet Information Service. It offers a myriad of advantages using the HTTP protocol by a service. Here, it is not required to have the host code for activating the service code; instead, the service code gets activated automatically.
- ▶ **Windows Activation Service** - This is popularly known as WAS and comes with IIS 7.0. Both HTTP and non-HTTP based communication is possible here by using TCP or Namedpipe protocols.
- ▶ **Self-hosting** - This is a mechanism by which a WCF service gets self-hosted as a console application. This mechanism offers amazing flexibility in terms of choosing the desired protocols and setting own addressing scheme.
- ▶ **Windows Service** - Hosting a WCF service with this mechanism is advantageous, as the services then remain activated and accessible to the client due to no runtime activation.

**#Difference between Message Contract and Data Contract**

Message Contracts:

Message contracts describe the structure of SOAP messages sent to and from a service and enable you to inspect and control most of the details in the SOAP header and body.

Data Contracts:

A data contract is a formal agreement between a service and a client that abstractly describes the data to be exchanged. Data contract can be explicit or implicit. Simple type such as int, string etc has an implicit data contract. User defined object are explicit or Complex type, for which you have to define a Data contract using [DataContract] and [DataMember] attribute.

---

## Web Services

1. What are web services?
2. What are the features of web services?
  - It is available over the Internet or private (intranet) networks.
  - It uses a standardized XML messaging system.
  - It is not tied to any one operating system or programming language.
  - It is self-describing via a common XML grammar.
  - It is discoverable via a simple find mechanism.
3. What the components of a Web Service?
4. How Does a Web Service Work?
5. What is the purpose of XML in a web service?
6. What is the purpose of SOAP in a web service?
7. What is the purpose of WSDL in a web service?
8. What are the benefits of Web Services?
9. What do you mean by Interoperability of Web Services?

Web services allow various applications to talk to each other and share data and services among themselves. Other applications can also use the web services. For example, a VB or .NET application can talk to Java web services and vice versa. Web services are used to make the application platform and technology independent.

10. What do you mean by loosely coupled architecture of Web services?
11. Do Web services supports Remote Procedure Calls(RPCs)?
12. What are the behavioral characteristics of web services?
13. What are the benefits of having XML based WEB services?

Using XML eliminates any networking, operating system, or platform binding. So Web Services based applications are highly interoperable application at their core level.

14. What is the benefit of a Web services being loosely coupled?
15. What is Synchronicity?
16. What are the core Roles in Web Service architecture?

17. What is the purpose of Service Provider in Web Service architecture?
18. What is the purpose of Service Requestor in Web Service architecture?
19. What is the purpose of Service Registry in Web Service architecture?
20. What are the core layers in Web Service Protocol Stack?
21. What is the purpose of Service Transport layer in Web Service Protocol Stack?
22. What is the purpose of XML Messaging layer in Web Service Protocol Stack?
23. What is the purpose of Service Description layer in Web Service Protocol Stack?
24. What is the purpose of Service Discovery layer in Web Service Protocol Stack?
25. What HTTP stands for?
26. What is HTTP?
27. What BEEP stands for?
28. What is BEEP?
29. What is XML-RPC?
30. How request is sent in XML-RPC?
31. How response is sent in XML-RPC?
32. What are the features of XML-RPC?
33. What SOAP stands for?
34. What is SOAP?
35. What are the features of SOAP?
36. Is SOAP platform independent?
37. What WSDL stands for?
38. What is WSDL?
39. What are the features of WSDL?
40. What UDDI stands for?
41. What is UDDI?
42. What are the features of UDDI?
43. What are the primary security issues with web services?
44. Which component of Web service describes interfaces to web services?
45. Which language UDDI uses?
46. Is XML-RPC is platform-dependent?
47. If a client sends an XML request to a server, can we ensure that the communication remains confidential?
48. If a client connects to a web service, how do we identify the user?
49. Is the user authorized to use the service?
50. What do you mean by Web services manageability?
51. How to handle Network security threats in Web services?