**ASP.NET (UI) Developers**

·        **Describe how a browser-based Form POST becomes a Server-Side event like Button1\_OnClick. :)**

·         I am not certain but I think when we need to submit some information or manipulate data at server side we use POST method which transmit client side information to server side in Request object and send the information to server and server perform task according to information given by client so in this case**Button1\_OnClick**is transferred to server and it’s handler called the relevant method.

·         **What is a PostBack? :)**

·         Postback is way to send client side information to server after loading it.

·         **What is ViewState? How is it encoded? Is it encrypted? Who uses ViewState?**:)

·         ViewState stores information so it could be accessible after postback and information could be persisted  after any number of postbacks but only within the scope of page. It encoded into Base64 format. It can be encrypted. Controls uses viewstate to store control state and data to be preserved after postback.

·         **What is the <machinekey> element and what two ASP.NET technologies is it used for? :)**

·         machinekey defines unique key which could be use by session and cache so if site is cluster based or single sign on based then it’s very useful.

·         **What three Session State providers are available in ASP.NET 1.1? What are the pros and cons of each?**:)

·         In-proc, out-proc, sqlserver

·         **What is Web Gardening? How would using it affect a design?**:)

·         If server has multiple processors and requests traffic is dividing accordingly that is called web gardening. If server is using external resources (like sql server, web services, logging) then it could be very useful.

·         **Given one ASP.NET application, how many application objects does it have on a single proc box? A dual? A dual with Web Gardening enabled? How would this affect a design? :)**

·         HTTPApplication object with single key based would be only 1 per worker process so in case of single processor box ONLY 1, in case of dual processor box without webgardening there would be only 1 single worker process so again single object but in case of web gardening two worker process means 2 objects. So need to have machine key otherwise … :D

·         **Are threads reused in ASP.NET between reqeusts? Does every HttpRequest get its own thread? Should you use Thread Local storage with ASP.NET? :)**

·         ASP.NET is MTA based so multiple threads are possible but I remember once I tried to call methods through callback based async delegates which makes it run on separate thread so when I tried that I lost httpcontext L in that thread. So each thread gets its own thread and usage is depends upon need.

·         **Is the [ThreadStatic] attribute useful in ASP.NET? Are there side effects? Good or bad? ;)**

·         I never used it but I think I got my just above mentioned problem’s solution that we can share httpContext through this attribute in multiple threads ;)

·         **Give an example of how using an HttpHandler could simplify an existing design that serves Check Images from an .aspx page. :)**

·         We can simply make httphandler for such specific requirements as .aspx are not specialized for images based web page which we could make and make things efficient and manipulate response according to our need. If page is based upon images only and needs to show loads of images thumbnail and such features we can make specific httpHandler for such pages to customize response and have more command on request.

·         **What kinds of events can an HttpModule subscribe to? What influence can they have on an implementation? What can be done without recompiling the ASP.NET Application? :)**

·         HTTPModule subscribe to Request life cycle kinds of events like beginRequest,EndRequest,Session,… They are useful to manipulate Request and generate Response according to state and lifecycle. They are separate DLL which are referred by asp.net application and if we change HTTPModule and recompile it we don’t need to recompile our whole application. So they are plugged like plugins ;)

·         **Describe ways to present an arbitrary endpoint (URL) and route requests to that endpoint to ASP.NET. ;)**

·         I am sure either I interpreted this question in right way but here is the answer according to my understanding about this question. URL could be define in couple of ways relative path and absolute path and request could be route in multiple ways using Redirect, Transfer, URL Rewrite…

·         **Explain how cookies work. Give an example of Cookie abuse. :)**

·         Cookie stored at client side and readable by any user if it’s not encrypted, if you stored some important information about client data at server any other site could read cookies and get that information.

·         **Explain the importance of HttpRequest.ValidateInput()? :)**

·         It checks integrity of data in Request object (Form, Querystring, Cookies), you can prevented by XSS types of attacks and validate request data to see so there is no malicious input ;)

·         **What kind of data is passed via HTTP Headers? :)**

·         HTTP Header contains information about browser and some of page like browser name, version, page size, mime type, cookies

·         **Juxtapose the HTTP verbs GET and POST. What is HEAD? :);)**

·         Get is used when we don’t need to do manipulation at server side and have nothing really secret information to send in querystring as it send information through querystring which is limited so input can’t be very long like article :D but it’s pretty fast as compare to POST as post sends information in hidden fields and preferred in case of manipulation at server side. Using Head we just send page header information not the actual contents/body.

·         **Name and describe at least a half dozen HTTP Status Codes and what they express to the requesting client. :)**

·         Very famous I just remember like 404 error page (page not found) 200 for OK, 500 Server not found

·         **How does if-not-modified-since work? How can it be programmatically implemented with ASP.NET?  
Explain <@OutputCache%> and the usage of VaryByParam, VaryByHeader. :)**

·         No idea about the if-not-modified thingy but can tell about OutputCache which is directive used for caching contents and tells server that Header information could be change by VaryByHeader Param and Parameter could be change in query string using VaryByParam so it get different versions for different header (like browsers) and parameters respectively.

·         **How does VaryByCustom work? :)**

·         Can output different pages from cache on the basis of change of state of any control or value or object within that page

·         **How would one implement ASP.NET HTML output caching, caching outgoing versions of pages generated via all values of q= except where q=5 (as in**[**http://localhost/page.aspx?q=5**](http://localhost/page.aspx?q=5)**)? :)**

·         <@OutputCache VaryByParam=”q” %>

ASP NET Life cycle.

|  |  |
| --- | --- |
| **Stage** | **Description** |
| Page request | The page request occurs before the page life cycle begins. When the page is requested by a user, ASP.NET determines whether the page needs to be parsed and compiled (therefore beginning the life of a page), or whether a cached version of the page can be sent in response without running the page. |
| Start | In the start stage, page properties such as [Request](http://msdn.microsoft.com/en-us/library/system.web.ui.page.request.aspx) and [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) are set. At this stage, the page also determines whether the request is a postback or a new request and sets the [IsPostBack](http://msdn.microsoft.com/en-us/library/system.web.ui.page.ispostback.aspx) property. The page also sets the [UICulture](http://msdn.microsoft.com/en-us/library/system.web.ui.page.uiculture.aspx) property. |
| Initialization | During page initialization, controls on the page are available and each control's [UniqueID](http://msdn.microsoft.com/en-us/library/system.web.ui.control.uniqueid.aspx) property is set. A master page and themes are also applied to the page if applicable. If the current request is a postback, the postback data has not yet been loaded and control property values have not been restored to the values from view state. |
| Load | During load, if the current request is a postback, control properties are loaded with information recovered from view state and control state. |
| Postback event handling | If the request is a postback, control event handlers are called. After that, the [Validate](http://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.basevalidator.validate.aspx) method of all validator controls is called, which sets the [IsValid](http://msdn.microsoft.com/en-us/library/system.web.ui.ivalidator.isvalid.aspx)property of individual validator controls and of the page. |
| Rendering | Before rendering, view state is saved for the page and all controls. During the rendering stage, the page calls the [Render](http://msdn.microsoft.com/en-us/library/system.web.ui.control.render.aspx) method for each control, providing a text writer that writes its output to the [OutputStream](http://msdn.microsoft.com/en-us/library/system.web.httpresponse.outputstream.aspx) object of the page's [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) property. |
| Unload | The [Unload](http://msdn.microsoft.com/en-us/library/system.web.ui.control.unload.aspx) event is raised after the page has been fully rendered, sent to the client, and is ready to be discarded. At this point, page properties such as [Response](http://msdn.microsoft.com/en-us/library/system.web.ui.page.response.aspx) and [Request](http://msdn.microsoft.com/en-us/library/system.web.ui.page.request.aspx) are unloaded and cleanup is performed. |

Authentication

**Windows authentication** enables you to identify users without creating a custom page. Credentials are stored in the Web server s local user database or an Active Directory domain. Once identified you can use the user s credentials to gain access to resources that are protected by Windows authorization.  
  
**Forms authentication**enables you to identify users with a custom database such as an ASP.NET membership database. Alternatively you can implement your own custom database. Once authenticated you can reference the roles the user is in to restrict access to portions of your Web site.  
  
**Passport authentication** relies on a centralized service provided by Microsoft. Passport authentication identifies a user with using his or her e-mail address and a password and a single Passport account can be used with many different Web sites. Passport authentication is primarily used for public Web sites with thousands of users.  
  
**Anonymous authentication** does not require the user to provide credentials.

how do you differentiate managed code and unmanaged code?

Managed code :Code that is executed by the CLR. Managed code provides information (i.e., metadata) to allow the CLR to locate methods encoded in assembly modules, store and retrievesecurity information, handle exceptions, and walk the program stack. Managed code can access both managed data and unmanaged data. Managed data—Memory that is allocated and released by the CLR using Garbage Collection. Managed data can only be accessed by managed code  
  
Unmanaged Code:Unmanaged code is what you use to make before Visual Studio .NET 2002 was released. Visual Basic 6, Visual C++ ,  It is compiled directly to machine code that ran on the machine where you compiled it—and on other machines as long as they had the same chip, or nearly the same. It didn't get services such as security or memory management from an invisibleruntime; it got them from the operating system. And importantly, it got them from the operating system explicitly, by asking for them, usually by calling an API provided in the Windows SDK. More recent unmanaged applications got operating system services through COM calls.

What is diffrence between debug.trace and trace.write?  
Where is the information of debug stored?

**Debug and Trace both are used to display messages. However Debug won't work in Release mode.  
Debug information stores in .pdb file.  
PDB stands for Project DataBase.**

Cookie and session.

Cookies:  
1) Cookies can hold small amount of data in text format.  
2) Cookies created on browser.  
3) Cookies can be disabled by user computer.  
  
Session:  
1) Session can hold large amount of data.  
2) Session created on the Server.  
3) Session can not be disabled by user computer.

**What role does the ScriptManager play?**

The ScriptManager manages all ASP.NET AJAX resources on a page and renders the links for the ASP.NET AJAX client libraries, which lets you use AJAX functionality like PageMethods, UpdatePanels etc. It creates the PageRequestManager and Application objects, which are prominent in raising events during the client life cycle of an ASP.NET AJAX Web page. It also helps you create proxies to call web services asynchronously.

**Can we use multiple ScriptManager on a page?**

No. You can use only one ScriptManager on a page.

**What is the role of a ScriptManagerProxy?**

A page can contain only one ScriptManager control. If you have a Master-Content page scenario in your application and the MasterPage contains a ScriptManager control, then you can use the ScriptManagerProxy control to add scripts to content pages.

Also, if you come across a scenario where only a few pages in your application need to register to a script or a web service, then its best to remove them from the ScriptManager control and add them to individual pages, by using the ScriptManagerProxy control. That is because if you added the scripts using the ScriptManager on the Master Page, then these items will be downloaded on each page that derives from the MasterPage, even if they are not needed, which would lead to a waste of resources.

**Explain the UpdatePanel?**

The UpdatePanel enables you to add AJAX functionality to existing ASP.NET applications. It can be used to update content in a page by using Partial-page rendering. By using Partial-page rendering, you can refresh only a selected part of the page instead of refreshing the whole page with a postback.

**Difference between Server-Side AJAX framework and Client-side AJAX framework?**

ASP.NET AJAX contains both a server-side Ajax framework and a client-side Ajax framework. The server-side framework provides developers with an easy way to implement Ajax functionality, without having to possess much knowledge of JavaScript. The framework includes server controls and components and the drag and drop functionality. This framework is usually preferred when you need to quickly ajaxify an asp.net application. The disadvantage is that you still need a round trip to the server to perform a client-side action.

The Client-Side Framework allows you to build web applications with rich user-interactivity as that of a desktop application. It contains a set of JavaScript libraries, which is independent from ASP.NET. The library is getting rich in functionality with every new build released.

**How can you debug ASP.NET AJAX applications?**

Explain about two tools useful for debugging: Fiddler for IE and Firebug for Mozilla.

**Can we call Server-Side code (C# or VB.NET code) from javascript?**

Yes. You can do so using [PageMethods in ASP.NET AJAX](http://www.dotnetcurry.com/ShowArticle.aspx?ID=109) or using webservices.

**Can you nest UpdatePanel within each other?**

Yes, you can do that. You would want to nest update panels to basically have more control over the Page Refresh.

**How can you to add JavaScript to a page when performing an asynchronous postback?**

Use the ScriptManager class. This class contains several methods like the RegisterStartupScript(), RegisterClientScriptBlock(), RegisterClientScriptInclude(), RegisterArrayDeclaration(),RegisterClientScriptResource(), RegisterExpandoAttribute(), RegisterOnSubmitStatement() which helps to add javascript while performing an asynchronous postback.

**Explain differences between the page execution lifecycle of an ASP.NET page and an ASP.NET AJAX page?**

In an asynchronous model, all the server side events occur, as they do in a synchronous model. The Microsoft AJAX Library also raises client side events. However when the page is rendered, asynchronous postback renders only the contents of the update panel, where as in a synchronous postback, the entire page is recreated and sent back to the browser.