



A Study on Page Life-Cycle of ASP.Net

Shikha Verma

Abstract— ASP.NET is a powerful platform for building Web applications. With any platform, it is important to understand the working of the application. The ASP.NET page life cycle has a series of page elements which are loaded and corresponding events are fired. In this paper the events are discussed in detail.

Keywords— Page lifecycle, init, pre-render, load, load complete, save state, view state, master pages.

I. INTRODUCTION

ASP.Net is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build up robust web application for PC, as well as mobile devices. ASP.Net works on top of the HTTP protocol and uses the HTTP commands and policies to set a browser-to-server two-way communication and cooperation. ASP.Net is a part of Microsoft .Net platform. ASP.Net applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework

Page and control event occurs in a certain order, which is called page Lifecycle. The precise order and the number of events in this lifecycle is discussed in this paper. At each stage of the page life cycle, the page raises some events, which could be coded. An event handler is basically a function or subroutine, bound to the event, using declarative attributes like Onclick or handle. When a page request is sent to the Web server, the page is run through a series of events during its creation and disposal.

Following are the page life cycle events:-

PREINIT

- Preinit is the first event that occurs during the page lifecycle
- It is used to check the ISPOSTBACK property of the page to determine whether the page to determine whether the page is being processed for the first time
- You can use this event to set the master page dynamically
- The page theme is set and initialized. The private method InitializeThemes method of the page is called
- It also creates or recreates dynamic controls
- It sets the theme property and the after any skin setting have been applied
- This event also calls the Get Control adapter which allows a developer to change the markup produced by server controls. This step gets any control adapter defined in the App_Browser folder for the control.
- In addition it gets and sets specific profile property values.
-

INIT

- This event is raised after the initialization of all the controls and after any skin settings have been applied
- It is used to read or initialize control properties
- One cannot access other server controls yet as there is no guarantee that it has been initialized yet .View state information cannot be used here.
- It is called during the initialize stage to initialize control properties

INIT COMPLETE

- This event is raised at the end of the page's initialization stage
- Use this event to make changes to view state that you want to make sure are persisted after the next post back

PRELOAD

- Raised after the page loads view state for itself and all controls and after it processes post back data that is included with the request instance

LOAD

- The onload event method is called to set the properties of controls and establish database connections

- The page class calls the onload event method on the webpage and then recursively does the same for each child control until the page and all controls are loaded.
- It is used to perform most of the processing steps for the page and controls that occur for each page request.

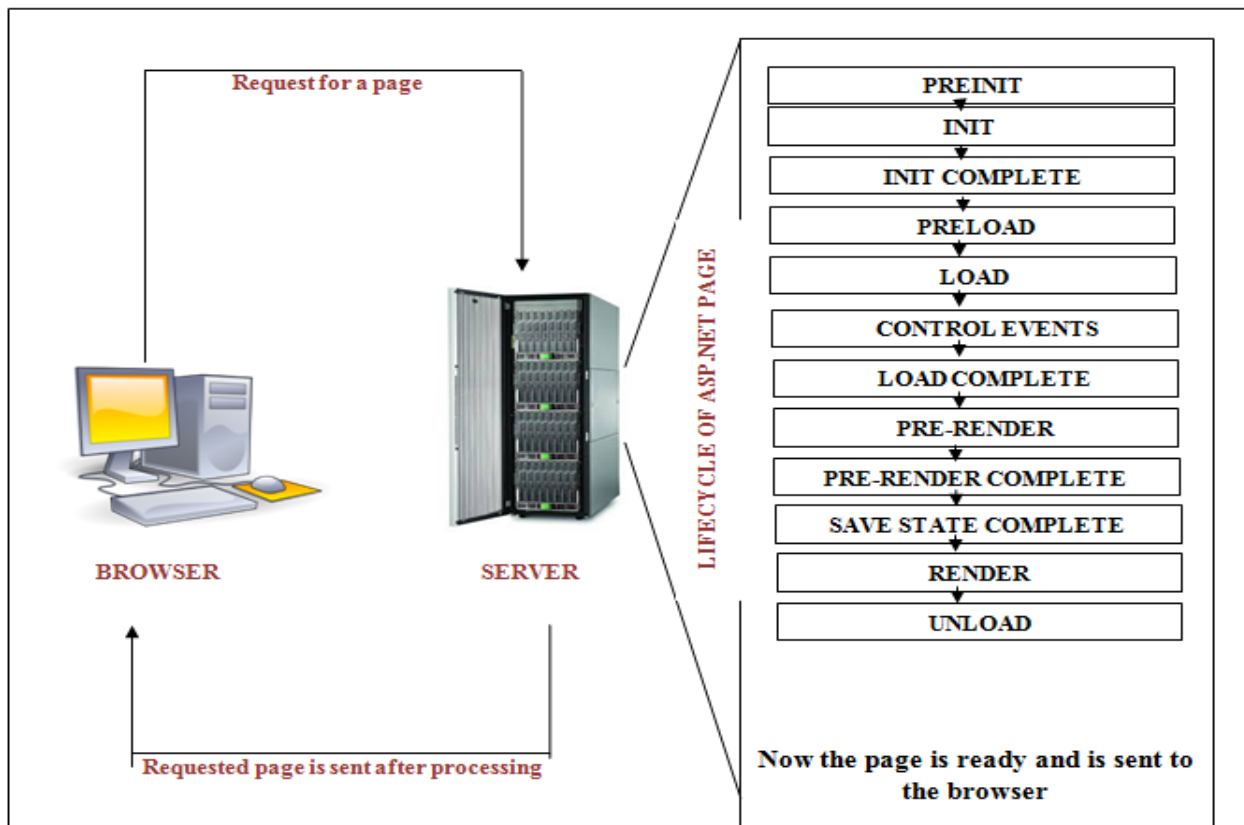


Fig 1:- ASP.NET Page Lifecycle

CONTROL EVENTS

- Use these events to handle specific, control events such as buttons control's click event or textbox control's text changed event
- Controls have their events invoked at this stage such as responding to a button click, the selected index changing on a list box and so on. If your page has validation controls, check the `IsValid` status of the controls and the page before continuing with your event handler

LOAD COMPLETE

- It occurs after all load events or raised at the end of the event handling stage.
- One can use this event for any tasks that requires all controls to be loaded
- Use this event for tasks that require that all other controls on the page to be loaded

PRE-RENDER

- Typically where page logic is written to deal with connection information
- This occurs before the page is about to render
- This event is triggered just before the final version of the page is rendered
- Raised after the page object has created all controls that are required in order to render the page, including child controls of the composite controls
- Use the event to make final changes to the content of the page or its controls before the rendering stage begins

PRE RENDER COMPLETE

- Raised after each data bound control whose `DATASOURCEID` property is set calls its data bind method.
- It indicates that all the content for the page has been pre-rendered.

SAVESTATE COMPLETE

- Personalization data is saved.
- This event is raised after view state and control state have been saved for the page and for all the controls.
- Any changes to the page or control at this control at this point affect rendering, but the changes will not be retrieved on the next postback

TABLE 1:- Availability of VIEW STATE in the lifecycle of ASP.NET

STAGE	LIFECYCLE EVENT	VIEWSTATE
INIT	PREINIT	
	INIT	
	INIT COMPLETE	
LOAD	PRELOAD	VIEW STATE AVAILABLE
	LOAD	
	LOAD COMPLETE	
	CONTROL EVENTS	
RENDERING	PRE-RENDER	
	PRE-RENDER COMPLETE	
	SAVE STATE COMPLETE	
	RENDER	
UNLOAD	UNLOAD	

RENDER

- This is not event, instead at this stage of processing; the page object calls this method on each control
- All asp.net web server controls have a RENDER method that writes out the controls markup to send to the browser.
- Calls the RenderControl method which recursively calls the RenderControl method of child control to render its html markup text which is sent to client browser.
- Creates an instance of HTML textwriter class that encapsulates the instance into it.

UNLOAD

- It is also known as the cleanup phase
- In this event page controls releases resources.
- This event is first called for all the controls. When all controls are done then it is called for the page.
- In controls, use this event to do final cleanup for specific controls, such as closing control specific database connections.
- In this case Response. Write will throw an exception if you attempt to call this.
- For the page itself, use this event to do final cleanup work such as closing open files and database
- This event is fired after the page is removed from the memory but hasn't yet disposed

II CONCLUSIONS

The requesting of an ASP.NET page triggers a sequence of events that encompass the page life cycle. The Web browser sends a post request to the Web server. The Web server recognizes the ASP.NET file extension for the requested page and sends the request to the HTTP Page Handler class. When a page is requested, it is loaded into the server memory, processed and sent to the browser. Then it is unloaded from the memory. At each of this steps, methods and events are available, which could be overridden according to the need of the application. In other words, you can write your own code to override the default code. Understanding the page cycle helps in writing codes for making some specific thing happen at any stage of the page life cycle. It also helps in writing custom controls and initializing them at right time, populate their properties with view-state data and run control behavior code.

REFERENCES

- [1] Asp.Net 3.5:Covers C# & Vb 2008 Codes, Black Book, Platinum ... -Kogent Solutions I
- [2] ASP.NET AJAX Programming Tricks -Matthew David Ellis