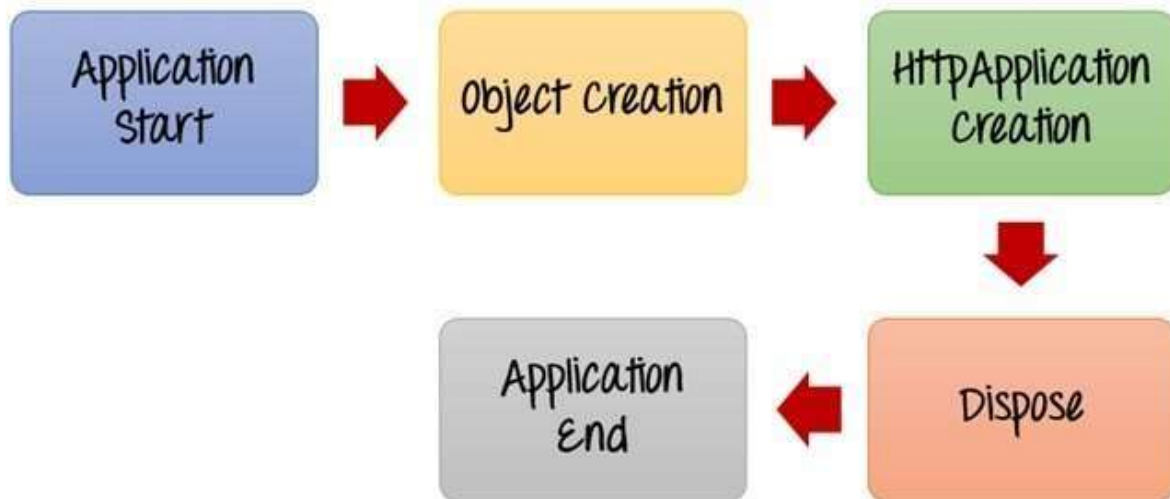


ASP.Net Lifecycle

When an ASP.Net application is launched, there are series of steps which are carried out. These series of steps make up the lifecycle of the application.

Let's look at the various stages of a typical page lifecycle of an ASP.Net Web Application.



ASP.Net Lifecycle

- 1) **Application Start** - The life cycle of an ASP.NET application starts when a request is made by a user. This request is to the Web server for the ASP.Net Application. This happens when the first user normally goes to the home page for the application for the first time. During this time, there is a method called `Application_start` which is executed by the web server. Usually, in this method, all global variables are set to their default values.
- 2) **Object creation** - The next stage is the creation of the `HttpContext`, `HttpRequest` & `HttpResponse` by the web server. The `HttpContext` is just the container for the `HttpRequest` and `HttpResponse` objects. The `HttpRequest` object contains information about the current request, including cookies and browser information. The `HttpResponse` object contains the response that is sent to the client.
- 3) **HttpApplication creation** - This object is created by the web server. It is this object that is used to process each subsequent request sent to the

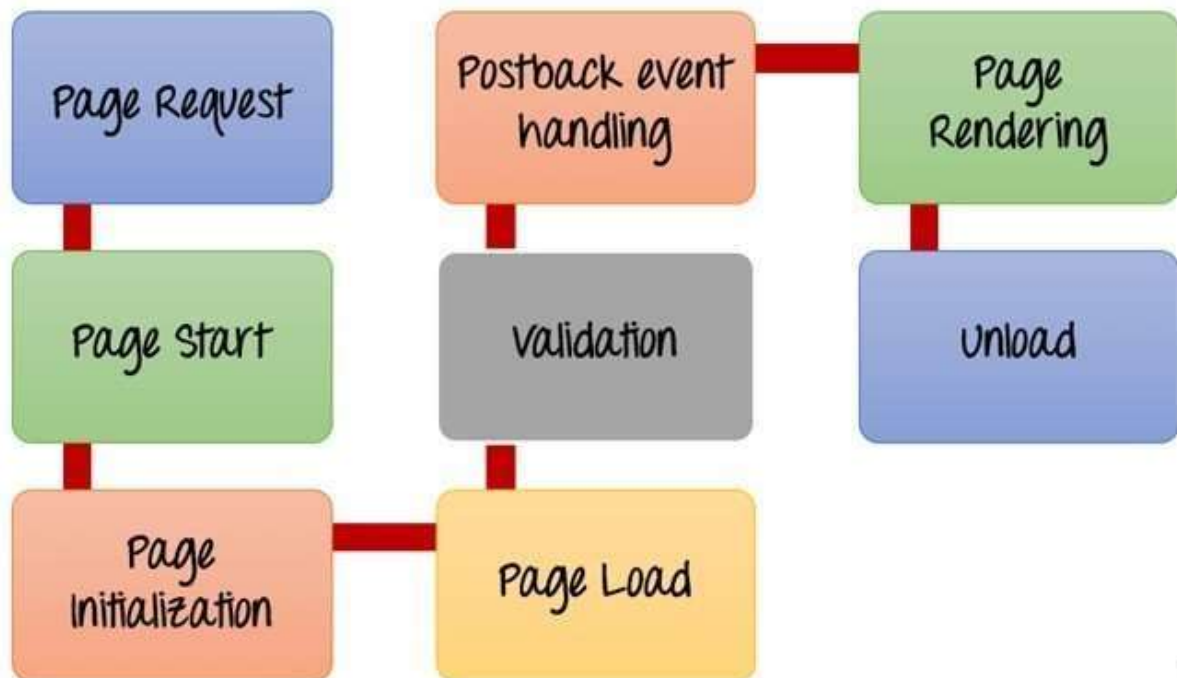
application. For example, let's assume we have 2 web applications. One is a shopping cart application, and the other is a news website. For each application, we would have 2 HttpApplication objects created. Any further requests to each website would be processed by each HttpApplication respectively.

- 4) **Dispose** - This event is called before the application instance is destroyed. During this time, one can use this method to manually release any unmanaged resources.
- 5) **Application End** - This is the final part of the application. In this part, the application is finally unloaded from memory.

ASP.Net Page Lifecycle?

When an ASP.Net page is called, it goes through a particular lifecycle. This is done before the response is sent to the user. There are series of steps which are followed for the processing of an ASP.Net page.

Let's look at the various stages of the lifecycle of an ASP.Net web page.



ASP.Net Page Lifecycle

1. **Page Request**- This is when the page is first requested from the server. When the page is requested, the server checks if it is requested for the first time. If so, then it needs to compile the page, parse the response and send it across to the user. If it is not the first time the page is requested, the cache is checked to see if the page output exists. If so, that response is sent to the user.
2. **Page Start** – During this time, 2 objects, known as the Request and Response object are created. The Request object is used to hold all the information which was sent when the page was requested. The Response object is used to hold the information which is sent back to the user.
3. **Page Initialization** – During this time, all the controls on a web page is initialized. So if you have any label, textbox or any other controls on the web form, they are all initialized.
4. **Page Load** – This is when the page is actually loaded with all the default values. So if a textbox is supposed to have a default value, that value is loaded during the page load time.
5. **Validation** – Sometimes there can be some validation set on the form. For example, there can be a validation which says that a list box should have a certain set of values. If the condition is false, then there should be an error in loading the page.
6. **PostBack event handling** – This event is triggered if the same page is being loaded again. This happens in response to an earlier event. Sometimes there can be a situation that a user clicks on a submit button on the page. In this case, the same page is displayed again. In such a case, the Postback event handler is called.
7. **Page Rendering** – This happens just before all the response information is sent to the user. All the information on the form is saved, and the result is sent to the user as a complete web page.

Unload – Once the page output is sent to the user, there is no need to keep the ASP.net web form objects in memory. So the unloading process involves removing all unwanted objects.