



**SOS IN COMPUTER SCIENCE & APPLICATION
JIWAJI UNIVERSITY**

Class : MBA (E-Commerce) II Semester

Subject : .Net Technologies

Paper Code: (204)

**Topic: States of ASP .NET (View state, control state,
session state, application state)**

By
Pushpendra Chaturvedi

States of ASP .NET-

Hyper Text Transfer Protocol (HTTP) is a stateless protocol. When the client disconnects from the server, the ASP.NET engine discards the page objects. This way, each web application can scale up to serve numerous requests simultaneously without running out of server memory.

However, there needs to be some technique to store the information between requests and to retrieve it when required. This information i.e., the current value of all the controls and variables for the current user in the current session is called the State.

ASP.NET manages four types of states:

- View State
- Control State
- Session State
- Application State

VIEW STATE –

The view state is the state of the page and all its controls. It is automatically maintained across posts by ASP.NET framework. When a page is sent back to the client, the changes in the properties of the page and its controls are determined, and stored in the value of a hidden input field named `_VIEWSTATE`. When the page is again posted back, the `_VIEWSTATE` field is sent to the server with the HTTP request. The view state could be enabled or disabled for:

- ❑ The entire application by setting the `EnableViewState` property in the `<pages>` section of `web.config` file.
- ❑ A page by setting the `EnableViewState` attribute of the Page directive, as `<%@ Page Language="C#" EnableViewState="false" %>`
- ❑ A control by setting the `Control.EnableViewState` property.

It is implemented using a view state object defined by the `StateBag` class which defines a collection of view state items. The state bag is a data structure containing attribute–value pairs, stored as strings associated with objects.

VIEW STATE –

The StateBag class has the following properties:

Properties	Description
Item(name)	The value of the view state item with the specified name. This is the default property of the StateBag class.
Count	The number of items in the view state collection.
Keys	Collection of keys for all the items in the collection.
Values	Collection of values for all the items in the collection.

VIEW STATE –

The StateBag class has the following methods:

Methods	Description
Add(name, value)	Adds an item to the view state collection and existing item is updated.
Clear	Removes all the items from the collection.
Equals(Object)	Determines whether the specified object is equal to the current object.
Finalize	Allows it to free resources and perform other cleanup operations.
GetEnumerator	Returns an enumerator that iterates over all the key/value pairs of the StateItem objects stored in the StateBag object.
GetType	Gets the type of the current instance.
Remove(name)	Removes the specified item.

SESSION STATE –

When a user connects to an ASP.NET website, a new session object is created. When session state is turned on, a new session state object is created for each new request. This session state object becomes part of the context and it is available through the page. Session state is generally used for storing application data such as inventory, supplier list, customer record, or shopping cart. It can also keep information about the user and his preferences, and keep the track of pending operations.

Sessions are identified and tracked with a 120-bit SessionID, which is passed from client to server and back as cookie or a modified URL. The SessionID is globally unique and random.

The session state object is created from the HttpSessionState class, which defines a collection of session state items.

SESSION STATE –

The HttpSessionState class has the following properties:

Properties	Description
SessionID	The unique session identifier.
Item(name)	The value of the session state item with the specified name. This is the default property of the HttpSessionState class.
Count	The number of items in the session state collection.
Timeout	Gets and sets the amount of time, in minutes, allowed between requests before the session-state provider terminates the session.

SESSION STATE –

The HttpSessionState class has the following properties:

Methods	Description
Add(name, value)	Adds an item to the session state collection.
Clear	Removes all the items from session state collection.
Remove(name)	Removes the specified item from the session state collection.
RemoveAll	Removes all keys and values from the session-state collection.
RemoveAt	Deletes an item at a specified index from the session-state collection.

APPLICATION STATE –

The ASP.NET application is the collection of all web pages, code and other files within a single virtual directory on a web server. When information is stored in application state, it is available to all the users.

To provide for the use of application state, ASP.NET creates an application state object for each application from the `HttpApplicationState` class and stores this object in server memory. This object is represented by class file `global.asax`.

Application State is mostly used to store hit counters and other statistical data, global application data like tax rate, discount rate etc. and to keep the track of users visiting the site.

The `HttpApplicationState` class has the following properties:

Properties	Description
Item(name)	The value of the application state item with the specified name. This is the default property of the <code>HttpApplicationState</code> class.
Count	The number of items in the application state collection.

APPLICATION STATE –

The `HttpApplicationState` class has the following methods:

Methods	Description
<code>Add(name, value)</code>	Adds an item to the application state collection.
<code>Clear</code>	Removes all the items from the application state collection.
<code>Remove(name)</code>	Removes the specified item from the application state collection.
<code>RemoveAll</code>	Removes all objects from an <code>HttpApplicationState</code> collection.
<code>RemoveAt</code>	Removes an <code>HttpApplicationState</code> object from a collection by index.

APPLICATION STATE –

The HttpSessionState class has the following methods:

Methods	Description
Lock()	Locks the application state collection so only the current user can access it.
Unlock()	Unlocks the application state collection so all the users can access it.

Application state data is generally maintained by writing handlers for the events:

- Application_Start
- Application_End
- Application_Error
- Session_Start
- Session_End