**HTTPGet and HTTPPost Action verbs**

**GET Request –**

**GET** is generally used to get data from the web server.

A GET request is generally issued:  
**1.** When you click on a hyperlink  
**2.** When Response. Redirect() statement is executed  
**3.** When you type URL in the address bar and hit enter  
  
**POST Request –**

**Post** is generally used to submit data to the server. A POST request is generally issued,  
**1.** When you click on a submit button  
**2.** When AUTOPOST back is set true and when a selection in the Dropdown List is changed  
  
**Difference between GET and POST method**  
**1.** GET method appends data to the URL, whereas with the POST method data can either be appended to the URL or in the message body.  
**2.** As GET request rely on query strings to send data to the server, there is a length restriction, whereas POST requests have no restrictions on data length.  
**3.** While it is possible to change the state of data in database using GET request, they should only be used to retrieve data.

**ViewData and ViewBag in mvc**

Both **ViewData**and **ViewBag**are used to pass data from a controller to a view. ViewData is a dictionary of objects that are stored and retrieved using strings as keys. The syntax of ViewData is very similar to that of ViewState, SessionState and ApplicationState.

// Storing data in ViewData  
ViewData["YourData"] = "SomeData";  
  
// Retrieving data from ViewData  
string strData = ViewData["YourData"].ToString();  
  
ViewData does not provide compile time error checking. For example, if you mis-spell the key names you wouldn't get any compile time error. You get to know about the error only at runtime.  
  
**ViewBag**uses the dynamic feature that was introduced in to C# 4.0. It allows an object to have properties dynamically added to it. Using ViewBag the above code can be rewritten as below.  
// Storing data in ViewBag  
ViewBag.YourData = "SomeData";  
  
// Retrieving data from ViewBag  
string strData = ViewBag.YourData;  
  
Just like ViewData, ViewBag does not provide compile time error checking. For example, if you mis-spell the property name, you wouldn't get any compile time error. You get to know about the error only at runtime.  
  
Internally ViewBag properties are stored as name/value pairs in the ViewData dictionary.

**Good practice:**

Always use strong types view model. Because it throws an error when you mis spell property name or method ,in compile time not in run time.