Introduction to HTTP

Hypertext Transfer Protocol (HTTP) is what the web is built on. All information is sent or requested using HTTP. Each request has a particular verb (GET and POST being the most common) that indicates what is desired. This is more obvious to MVC developers than WebForms developers because it is exposed more directly to the developer.

<http://en.wikipedia.org/wiki/HTTP_Verbs>

Pay specific attention to the following sections on that page: Request Methods, Safe Methods.

One important thing to remember is that the web is *stateless*. In most programs data (in the form of variables) exist for the duration of the program, unless explicitly destroyed. Variables created during a server’s handling of a web request are not typically persisted until future requests are made. This statelessness must be considered during application development.

The response to the user agent (typically a browser) indicates success or failure of a request. Failure can be due to many reasons and you need to be familiar with HTTP response status codes.

<http://en.wikipedia.org/wiki/List_of_HTTP_status_codes>

With this basic understanding, it is time to introduce an important pattern Post-Redirect-Get. This pattern allows for a smoother user experience when retrieving and refreshing web pages in the browser. It is a pattern that should be used to smooth the user experience when MVC is the web application platform.

<http://en.wikipedia.org/wiki/Post/Redirect/Get>

Now, understanding that HTTP is designed as a stateless protocol, all server side development platforms allow for long-term state management. There are issues to consider regarding memory consumption, persistence, etc. There are several different details, but often there is a use of a HTTP cookie to identify the user agent for a HTTP session. This session identifier is used to link a request to prior data stored in the server data store (memory, database, etc.). It is preferable that the only information that session is used to store is information that would be consistent for a user across all tabs of a web browser so that there is no contamination of data or expectations of a user that are broken due to session issues. Additional details of how to implement session state will be covered in other training documents.