

REVIEW LESSON

MTA Course: Web Development Fundamentals

Lesson name: Web Development Fundamentals 1.2 (One 50-minute class period)

Topic: Understand ASP.NET intrinsic objects

File name: WebDevFund_RL_1.2

Lesson Objective:

1.2: Understand ASP.NET intrinsic objects. *This objective may include but is not limited to:* Request, Server, Application, Session, Response, HttpContext.

Preparation Details

Prerequisite student experiences and knowledge

This MTA Certification Exam Review lesson is written for students who have learned about Web design and Web application programming. Students who do not have the prerequisite knowledge and experiences cited in the objective will find additional learning opportunities using resources such as those listed in the Microsoft® resources and Web links at the end of the lesson.

Instructor preparation activities

For this lesson, you will need a computer with Microsoft Office 2007® attached to a liquid crystal display (LCD) projector to display and review the attached Microsoft PowerPoint® document. You will also need Web development software such as Microsoft Expression Web or Microsoft Visual Web Developer.

Resources, software, and additional files needed for this lesson:

- Office (preferably 2007)
- LCD projector
- Microsoft Visual Basic 2008®, Express Edition
- Microsoft Visual C# 2008®, Express Edition

- Microsoft Visual Web Developer 2008, Express Edition
- WebDevFund_PPT_1.2
- WebDevFund_SA_1.2

Teaching Guide

Essential Vocabulary:

application—A computer program that defines the methods, properties, and events to be used in this website. Sample programming languages include Visual Basic®, C#, and C++.

HttpApplication—A property of the HttpContext class which defines the methods, properties, and events that are common to all application objects in an ASP.NET application. An ASP.NET application is the sum of all files, pages, handlers, modules, and code within the scope of a virtual directory and its subdirectories on a single Web server. This class is the base class for applications that are defined by the user in the Global.asax file.

application state—Enables sharing of global information across multiple sessions and requests within an ASP.NET application.

HttpApplicationState—A property of the HttpContext class which enables sharing of global information across multiple sessions and requests within an ASP.NET application.

HttpContext—This .NET Framework class encapsulates all HTTP-specific information about an individual HTTP request, and contains many intrinsic ASP.NET objects, including: Response, Request, Application, Session, and Server. Although these objects are properties of the System.Web.HttpContext class, because the objects are automatically created by ASP.NET when a new request for a Web resource is received and a new context is created, you can use them directly without having to instantiate new objects.

request—A property of the HttpContext class, this is an intrinsic object that allows you to retrieve posted data (i.e., Request.Form is used for the POST command). Enables ASP.NET to read the Hypertext Transfer Protocol (HTTP) values sent by a client during a Web request.

HttpRequest—A property of the HttpContext class which enables ASP.NET to read the HTTP values sent by a client during a Web request.

response—Allows a message or data to be sent back to the client in a format displayable by a browser.

HttpResponse—A property of the HttpContext class which Encapsulates HTTP-response information from an ASP.NET operation

server—This is the computer that receives requests from the client. It also stores data and provides helper methods for processing Web requests.

HttpServer—A property of the HttpContext class which gets the HttpServerUtility object that provides methods used in processing Web requests.

session state—Manages data sent from one page to be used by another page later.

HttpSessionState—A property of the HttpContext class which gets the HttpSessionState object for the current HTTP request. Provides session-state management to enable you to store information associated with a unique browser session across multiple requests. You can use this object to store a collection of values referenced by a key name or by numerical index. Access to session values and functionality is available to any page accessed within the session's lifetime.

state of the client—Mostly the contents of form input fields collectively referred to as the *page state*; is easily accessible through the server-side collections that store posted values.

View State—The concept and object used by an ASP.NET Web page to persist changes to the state of a Web Form across postbacks. The properties and values of any ASP.NET page control or other page-level object may be included in the ViewState object which is transferred, deserialized, and interpreted when a page is loaded to determine the state of the controls and page objects before the page was posted, enabling them to be re-created in the same state as they were before the page was posted. ViewState is often used to persist page-level variables which would otherwise be disposed of when the page is unloaded to perform a postback.

Control State—Control state is designed for storing a control's essential data (such as a pager control's page number) that must be available on postback to enable the control to function. Control State is similar to View State but functionally independent, and is always available regardless of whether the View State has been disabled or not.

Lesson Sequence

Activating prior knowledge/lesson staging (10 minutes)

1. View the site link: <http://www.asp.net/get-started/> and have the students search the sites for “Who is using ASP.NET?” As a group, identify what parts of the pages are specifically using ASP.NET, such as the shopping cart, and talk about how this works. Use one of the sites to review the event model on the PowerPoint slides and talk about the session state and why it is important to use ASP.NET instead of strictly Extensible Hypertext Markup Language (XHTML)?

Lesson activities (30 minutes)

1. Using the PowerPoint presentation WebDevFund_PPT_1.2, walk the students through the concepts for this lesson (approximately 10 minutes).
2. Using the Web development application for your class, create a simple Web page for a user login page using the ASP.NET Framework (20 minutes). Have the students follow along and complete the same assignment. Discuss how each of the parts relates to the vocabulary for this lesson.
3. Distribute Student Activity document WebDevFund_SA_1.2.
4. Test the final product to ensure that it works as expected in a Web browser.

Assessment/lesson reflection (10 minutes)

1. Ask the students to reflect on the activity and provide areas that need additional explanation.
2. Wrap up the discussion and provide homework/enrichment opportunities.

Microsoft resources and Web links

Ramp-Up:

<http://msdn.microsoft.com/en-us/rampup/dd861531.aspx/>

The official Microsoft ASP.NET site:

<http://www.asp.net/>

Additional activities (homework or enrichment):

1. Have the students watch the following videos on creating a Web form user interface and application state:
<http://www.asp.net/learn/3.5-videos/video-13.aspx>
<http://www.asp.net/learn/3.5-videos/video-11.aspx>
2. Tell them to review any of the other videos that address parts of the Web Developer Fundamentals.

Suggested best practices:

- Peer review and class presentations encourage collaboration and quality products.
- Use the prompting questions from Bloom's taxonomy, such as:
 - How can you use...?
 - Why do you think...?
 - Can you list the parts...?

- Can you make changes to solve...?
- Can you predict the outcome if ...?
- Can you identify the different parts...?

(More are available if you research Bloom's Taxonomy.)