

Share on your Social
Media



ASP DOTNET Tutorial

Published On: July 31, 2024

ASP DOTNET Tutorial

Microsoft created the web framework known as ASP.NET. It is employed in the development of web services, web apps, and websites. This ASP.NET tutorial includes topics such as features, server control, projects, MVC, and so on.

[ASP DOTNET Tutorial PDF](#)

Introduction to ASP.NET

Microsoft created and commercialized the ASP.NET web application framework, which enables programmers to create dynamic websites. It makes it simple to create web apps using a feature-rich programming language like C# or VB.NET.

This ASP.NET tutorial covers the following:

- Overview of ASP.NET
- Web Forms
- ASP.Net MVC
- Web Pages
- Web API
- Web Hooks
- Ajax
- ASP.NET Sample Page

[ASP DOTNET Interview Questions](#)

Featured Articles



Want to know
more about
becoming an
expert in IT?

[Click Here to Get Started](#)

100%
Placement
Assurance

AUTHORISED
CERTIFICATION
PARTNER

IBI

Quick Enquiry

Related Courses at SLA

Related Posts



C and C++ Tutorial

Published On: August 1, 2024

C and C++ Tutorial C is a high-level, procedural, general-purpose programming language. Whereas C++, a...

Overview of ASP.NET

Web Forms, ASP.NET MVC, and ASP.NET Web Pages are the three web application frameworks that ASP.NET provides. Any of the three frameworks can be used to develop excellent web apps because they are all reliable and sophisticated.

You can use all the capabilities and advantages of ASP.NET everywhere, regardless of the framework you select. Every framework focuses on a distinct approach to development.

ASP.NET Life Cycle

One might categorize the ASP.NET life cycle into two groups:

- Application Life Cycle
- Page Life Cycle

Application Lifecycle

The application life cycle consists of the following steps:

- The user requests access to a page, which is an application resource. The browser sends this request to the web server.
- The following events take place when a unified pipeline receives the initial request:
 - It creates an instance of the `ApplicationManager` class.
 - To offer details about the resources, an object of the class `HostingEnvironment` is created.
 - The application's top level objects are compiled.
- Objects for responses are made. Application objects are built and initialized, including `HttpContext`, `HttpRequest`, and `HttpResponse`.
- The request is associated with an instance of the `HttpApplication` object that has been constructed.
- The `HttpApplication` class handles the request.



ASP DOTNET Tutorial

Published On: July 31, 2024

ASP DOTNET Tutorial Microsoft created the web framework known as ASP.NET. It is employed in...



Artificial Intelligence Tutorial

Published On: July 30, 2024

Artificial Intelligence Tutorial Artificial intelligence (AI) is significant since it enhances many facets of society...



Appium Testing Tutorial

Published On: July 30, 2024

Appium Testing Tutorial Designed to make the UI automation of many app platforms easier, Appium...

This class raises many events in order to process the request.

Page Lifecycle

Using all of the page's controls, the Page class builds a hierarchical tree. The stages in the page life cycle are:

- Initialization
- Instantiation of the controls on the page
- Restoration and maintenance of the state
- Execution of the event handler codes
- Page rendering

Below are the page lifecycle events in ASP.NET:

PreInit: It uses the IsPostBack attribute to check if the page is a postback.

Init: The control property is initialized and the control tree is constructed here.

InitComplete: Tracking of the view state is possible via the InitComplete event.

LoadViewState: Data about the current view state can be loaded into controls using the LoadViewState event.

LoadPostData: All input fields that are defined with the <form> element have their data evaluated during this phase.

PreLoad: Before the post-back data is loaded into the controls, PreLoad takes place.

Load: The page raises the Load event initially, and all child controls are raised recursively after that.

LoadComplete: This indicates that page validation, control event handlers, and loading are all finished.

PreRender: This event takes place immediately before the output is produced.

PreRenderComplete: This event guarantees that

the pre-rendering stage is finished because the PreRender event is fired recursively for all child controls.

SaveStateComplete: The page's state of control is preserved.

Unload: The final stage of the page life cycle is known as UnLoad. Recursively, it raises the UnLoad event for each control before raising it for the page itself.

[ASP DOTNET Syllabus PDF](#)

Web Forms

You may create dynamic websites with ASP.NET Web Forms by utilizing an event-driven, drag-and-drop approach that is familiar to you.

- With a design surface and hundreds of controls and components, you can create complex, feature-rich UI-driven websites with data access quickly.
- Included with Visual Studio, ASP.NET Web Forms is a component of the ASP.NET web application platform.
- It is one of the four programming models that may be used to create ASP.NET online apps, along with ASP.NET MVC, ASP.NET web pages, and ASP.NET Single Page apps.

What are web forms?

- Built on Microsoft ASP.NET technology, server-side code generates Web page content dynamically and sends it to the browser or client device.
- Compatible with all mobile devices and browsers. Together with ASP.NET MVC, ASP.NET web pages, and ASP.NET single-page apps, it is one of the four programming models that can be used to construct ASP.NET online applications.

- Every language that the .NET common language runtime supports is compatible with it, including Microsoft Visual Basic and Microsoft Visual C#.
- Based on the Framework for Microsoft.NET. All the advantages of the framework, such as inheritance, type safety, and a managed environment, are thus provided.
- Adaptable since you may provide them with third-party and user-created controls.

What do web forms provide?

- Separating application functionality from HTML and other UI code.
- Strong data binding with excellent support for tools.
- Support for browser-based client-side scripting.
- A comprehensive set of server controls for standard operations like data access.
- Support for a wide range of additional features, such as state management, debugging, routing, security, performance, and internationalization.

Why ASP.NET Web Forms?

Writing web apps has its own set of difficulties that are not usually encountered while writing client-side programs. A few of the difficulties are:

- Implementing a rich Web user interface
- Separation of client and server
- Stateless execution
- Unknown client capabilities
- Complications with data access
- Complications with scalability

The ASP.NET framework and ASP.NET Web Forms handle these issues in the following ways:

- Intuitive, consistent object model
- Event-driven programming model
- Intuitive state management

- Browser-independent applications
- NET Framework common language runtime support
- NET Framework scalable server performance

Features of ASP.Net Web Forms

Below are the features of Web Forms:

Server Controls: When an ASP.NET Web page is requested, server controls are objects on the page that run and provide HTML to the browser.

Master Pages: You may design a unified style for all of the pages in your application with ASP.NET master pages.

Working with Data: There are numerous ways to save, retrieve, and display data with ASP.NET.

Membership: ASP.NET Identity keeps your users' login information in a database that it creates.

Client Script and Client Frameworks: By integrating client-script functionality into ASP.NET Web Form pages, you can improve the server-based features of ASP.NET.

Routing: With URL routing, you can set up an application to receive request URLs that are not mapped to actual files.

State Management: ASP.NET Web Forms offers a number of tools to help you maintain data at the application-wide and per-page levels.

Security: Recognizing the risks to an application is crucial to creating one that is more secure.

Performance: You can adjust how quickly page and server control processing occur with ASP.NET Web Forms.

Internationalization: With ASP.NET Web Forms, you can design web pages that can pull in content and other data according to the user's selected

language or the browser's preferred language setting.

Debugging and Error Handling: ASP.NET has tools to assist you in identifying potential issues with your Web Forms application.

Deployment and Hosting: You can get assistance with the process of deploying and hosting your Web Forms application by using tools from Visual Studio, ASP.NET, Azure, and IIS.

ASP DOT NET Developer Salary

ASP.NET MVC

With ASP.NET MVC, you can create dynamic websites with ease and in a pleasant, agile manner. Its powerful, pattern-based approach allows for a clear separation of concerns and provides you with complete control over the HTML.

Numerous capabilities of ASP.NET MVC allow for quick, TDD-friendly development to create complex apps that make use of the most recent web standards.

An application can be developed or designed using the Model-View-Controller (MVC) pattern, which divides an application into three primary parts:

- Model
- View
- Controller

Model: The model is an application component that carries out the logic for the application's data domain. In a database like the SQL Server database, it is used to obtain and store model states. Separating business logic from the application's data is another purpose for it.

View: View is a part of the user interface of the application. The application's web pages are made with it. An edit view of a Products table that shows

text boxes, check boxes, and drop-down menus depending on the state of a Product object at that moment would be an example.

Controller: The part in charge of managing user interaction is the controller. In order to render the web page, it works with the model and chooses the view. The view in an MVC application does little more than show data; user input and requests are handled by the controller.

Advantages of ASP.NET MVC Framework

Master pages and membership-based authentication are two aspects that are implemented in this lightweight framework design approach. The System.Web.Mvc assembly defines it.

The following benefits are offered by this strategy:

- By splitting a program into its model, view, and controller, it controls the complexity of the application.
- Neither server-based forms nor view states are used. Because of this, developers that desire complete control over an application's behavior will find the MVC framework to be perfect.
- It provides better support for test-driven development.
- Web applications and large-scale development teams can benefit from it.
- It gives the developer a great deal of control over how the application behaves.

Web Pages

Combining server code and HTML to create dynamic web content is quick, easy, and lightweight when using ASP.NET Web Pages and the Razor syntax.

You may develop stunning websites that adhere to the most recent web standards by connecting to

databases, adding video, linking to social networking sites, and incorporating many other features.

ASP.NET Razor

We are able to incorporate server code into the web pages using the standard markup syntax called Razor. To generate views, it makes use of its own syntax and keywords.

In the event that a web page contains server code, the server runs the code first before responding to the browser. It enables us to carry out rational operations on the view page. On the view page, we can construct variables, loops, and expressions.

Its easier to learn and the code syntax is streamlined. The extension of this file is.cshtml.

@ Character

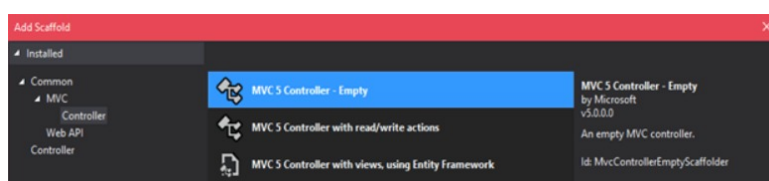
Razor uses this character to translate HTML to C#. The @ symbol changes to razor-specific markup when it is used with razor syntax; otherwise, it changes to plain C#. It was employed to begin a single line of text, a single sentence, or a multi-statement block.

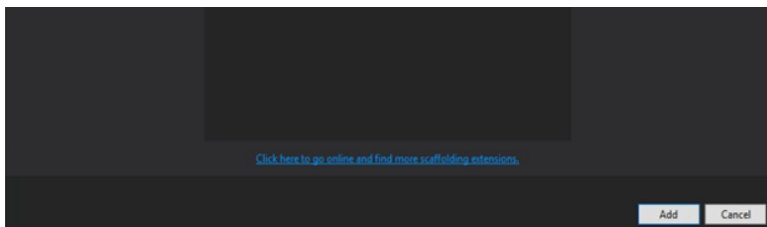
Razor keywords

- functions
- inherits
- model
- section
- helper (Not supported by ASP.NET Core.)

Example

Now let's develop a Razor syntax view. The following dialog box will appear when you right-click on the Controller folder and choose Add->Controller.





ASP.NET Tutorial

Name a controller

ASP.NET Tutorial

By selecting the “add” button, the controller will be created and the following code will be added.

Name: //StudentsController.cs

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Mvc;

namespace RazorViewExample.Controllers

{

    public class StudentsController : Controller

    {

        public ActionResult Index()

        {

            return View();

        }

    }

}
```

Index.cshtml

```
@{

    Layout = null;
```

```
var name = "Bright";

}

<!DOCTYPE html>

<html>

<head>

    <meta name="viewport" content="width=device-width" />

    <title>Index</title>

</head>

<body>

    <h2>My name is @name </h2>

</body>

</html>
```

Output



ASP.NET Tutorial

Web API

A framework for creating web APIs on top of the .NET Framework is called ASP.NET Web API.

- HTTP is used for more than just web page delivery.
- Another effective framework for creating APIs that expose data and services is HTTP.
- HTTP is simple, adaptable, and widely used.
- Because HTTP libraries are available for almost any platform you can imagine,
- HTTP services are accessible to a wide variety of clients, such as browsers, mobile devices, and conventional desktop applications.

ASP dot net Training

Web Hooks

A lightweight HTTP structure called WebHooks offers a straightforward pub/sub architecture for connecting Web APIs and SaaS services.

- Registered subscribers receive notifications via HTTP POST requests whenever an event occurs in a service.
- The information about the event included in the POST request enables the recipient to respond appropriately.

Sending and receiving WebHooks as part of your ASP.NET application is made simpler with Microsoft ASP.NET WebHooks:

- It offers a standard mechanism for accepting and handling WebHooks from any number of WebHook providers on the receiving end.

Dropbox, GitHub, PayPal, Salesforce, Trello, WordPress, and Zendesk are supported out of the box, but it is simple to add support for more.

- It supports sending event notifications to the appropriate group of subscribers in addition to handling and storing subscriptions on the sending end.

With this, you can create a custom set of events that subscribers can sign up for, and you can tell them when anything happens.

Web Hooks Processing Pipeline

For incoming WebHooks, the Microsoft ASP.NET WebHooks processing pipeline looks like this:

Here, receivers and handlers are the two essential ideas:

- Receivers are in charge of managing the specific WebHook flavor that a sender sends and implementing security measures to guarantee that the sender of the WebHook request is who they say they are.
- Handlers are the places where user code executes to process a specific WebHook.

ASP.NET Ajax

The acronym for Asynchronous JavaScript and XML is AJAX. This cross-platform technology accelerates response times.

- The AJAX server manages the addition of script to the page, which the browser runs and processes.
- These AJAX server controls can, however, be connected with methods and event handlers that are handled on the server side, much like conventional ASP.NET server controls.
- 'AJAX Extensions' are a set of controls found in the Visual Studio IDE's control toolbox.

Script Manager

The most crucial control is the ScriptManager control, which is necessary for other controls to function and must be present on the page.

Syntax:

```
<asp:ScriptManager ID="ScriptManager1"
runat="server">

</asp:ScriptManager>
```

Update Panel

Derived from the Control class, the UpdatePanel control is a container control. It has no interface of its own; instead, it serves as a container for the child controls that are housed therein. The UpdatePanel steps in to start the post asynchronously and update only that section of the page when a control inside it causes a post back.

ASP DOTNET Tutorial

Update Progress

While updating one or more update panel controls, the UpdateProgress control acts as a kind of browser feedback.

Example: During a user's login process or while they're working on a database-related task and waiting for a server answer, it shows a visible indication that work is ongoing, such as "Loading page..."

Syntax

```

<asp:UpdateProgress ID="UpdateProgress1" runat="server"
DynamicLayout="true" AssociatedUpdatePanelID="UpdatePanel1"
>

    <ProgressTemplate>

        Loading...

    </ProgressTemplate>

</asp:UpdateProgress>

```

Timer Control

The timing control initiates the automated post-back. There are two ways to go about this:

- Configuring the UpdatePanel control's triggers property:

```
<Triggers>
```

```

    <asp:AsyncPostBackTrigger ControlID="btnpanel2"
    EventName="Click" />

```

```
</Triggers>
```

- Installing a timer control as a child control trigger straight within the UpdatePanel. A single timer can initiate several UpdatePanels.

```

<asp:UpdatePanel ID="UpdatePanel1" runat="server"
UpdateMode="Always">

    <ContentTemplate>

        <asp:Timer ID="Timer1" runat="server" Interval="1000">

            </asp:Timer>

            <asp:Label ID="Label1" runat="server" Height="101px"
            style="width:304px" >

                </asp:Label>

            </ContentTemplate>

    </asp:UpdatePanel>

```

ASP.NET Sample Page

A server-side file stored with the.aspx extension can also be an ASP.NET page. Its modular design allows it to be split into the following main sections:

- Page Directives
- Code Section
- Page Layout

Page Directives

The page directives set up the operational environment of the page.

- The ASP.NET page parser and compiler use page-specific characteristics defined by the @Page directive.
- Page directives define the assumptions that must be made about the page and how it should be processed.
- Namespace imports, assembly loading, and the registration of new controls with unique tag names and namespace prefixes are all supported.

Code Selection

The handlers for the page and control events, as well as other necessary functions, are provided in the code section.

- ASP.NET uses an object model. These entities initiate actions in response to user interface events, such as button clicks or cursor movements.
- The event handler functions code the type of answer that these events require in return.
- All of the event handlers have functions that are attached to the controls.
- All of these event handler routines and other developer-utilized functionalities are provided in the code section or code behind file.
- It is possible to precompile the page code and distribute it as a binary assembly.

Page Layout

The page layout offers a page interface. It includes HTML tags, text, inline JavaScript, and server controls.

A sample ASP.NET page demonstrating page directives, a code section, and page layout in C# may be found in the code excerpt below:

```
<!-- directives -->

<% @Page Language="C#" %>

<!-- code section -->

<script runat="server">

    private void convertoupper(object sender, EventArgs e)

    {

        string str = mytext.Value;

        changed_text.InnerHtml = str.ToUpper();

    }

</script>

<!-- Layout -->

<html>

<head>

    <title> Change to Upper Case </title>

</head>

<body>

    <h3> Conversion to Upper Case </h3>

    <form runat="server">

        <input runat="server" id="mytext" type="text" />

        <input runat="server" id="button1" type="submit"
value="Enter..." OnServerClick="convertoupper"/>

        <hr />

        <h3> Results: </h3>

        <span runat="server" id="changed_text" />

    </form>

</body>

</html>
```

To the root directory of the web server, copy this file. It is often c:\inetput\wwwroot. When you run the file

through a browser, the following outcome is produced:

ASP DOTNET Tutorial

In Visual Studio IDE

Let's utilize the Visual Studio IDE to create the same example. You can drag the controls into the design view to avoid inputting the code:

ASP DOTNET Tutorial

The content file is generated automatically. To complete the setup, simply add the Button1_Click routine, which looks like this:

protected void Button1_Click(object sender, EventArgs e)

```
{  
  
    string buf = TextBox1.Text;  
  
    changed_text.InnerHtml = buf.ToUpper();  
  
}
```

The following is the content file code:

```
<%@ Page Language="C#" AutoEventWireup="true"  
CodeBehind="Default.aspx.cs"  
  
    Inherits="firstexample._Default" %>  
  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml" >  
  
    <head runat="server">  
  
        <title>  
  
            Untitled Page  
  
        </title>  
  
    </head>  
  
    <body>  
  
        <form id="form1" runat="server">  
  
            <div>  
  
                <asp:TextBox ID="TextBox1" runat="server"  
style="width:224px">  
  
                </asp:TextBox>  
  
                <br />  
  
                <br />  
  
                <asp:Button ID="Button1" runat="server" Text="Enter..."  
style="width:85px" onclick="Button1_Click" />  
  
                <hr />  
  
                <h3> Results: </h3>
```

```
<span runat="server" id="changed_text" />

</div>

</form>

</body>

</html>
```

Output:

ASP DOTNET Tutorial

Conclusion

We hope this ASP.NET tutorial will be helpful to gain a basic understanding of how ASP.NET works. Get started on your learning journey by enrolling in our [**ASP.NET training in Chennai.**](#)

Share on your Social
Media



Navigation

About Us

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.

Landmark: Karnataka Bank Building

Phone: [+91 86818 84318](tel:+918681884318)

Email: enquiry@softlogicsys.in

Map: [Google Maps Link](#)

OMR

No. EI-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai – 600 130.

Landmark: Adj. to AGS Cinemas

Phone: [+91 89256 88858](tel:+918925688858)

Email: info@softlogicsys.in

Map: [Google Maps Link](#)

Courses

Python

Software Testing

Full Stack Developer

Java

Power BI

Clinical SAS

Data Science

Embedded

Cloud Computing

Hardware and Networking

VBA Macros

Mobile App Development

DevOps

[Blog Posts](#)

[Careers](#)

[Contact](#)

[Placement Training](#)

[Corporate Training](#)

[Hire With Us](#)

[Job Seekers](#)

[SLA's Recently Placed Students](#)

[Reviews](#)

[Sitemap](#)

Important Links

[Disclaimer](#)

[Privacy Policy](#)

[Terms and Conditions](#)

Social Media Links



Review Sources

[Google](#)

[Trustpilot](#)

[Glassdoor](#)

[Mouthshut](#)

[Sulekha](#)

[Justdial](#)

[Ambitionbox](#)

[Indeed](#)

[Software Suggest](#)

