Introduction to ASP.NET

What is ASP.NET? How it works?
Architecture of ASP.NET

Mohammad Jawad
Table of Contents

- Introduction to ASP.NET
  - What is ASP.Net
  - History of ASP.NET
- ASP.NET Architecture Overview
  - Separation of Presentation from Logic
- ASP.NET Base Components
  - Web Controls
  - Web Forms
  - Web.Config
  - Global.asax
What is ASP.Net

ASP.NET is a server side scripting technology that enables scripts (embedded in web pages) to be executed by an Internet server.

- ASP.NET is a Microsoft Technology
- ASP stands for Active Server Pages
- ASP.NET is a program that runs inside IIS
- IIS (Internet Information Services) is Microsoft's Internet server
- IIS comes as a free component with Windows servers
- IIS is also a part of Windows 2000, XP, Vista, 7, 8 and 10
History of ASP.Net

- At the beginning of Internet (up to 1997)
  - CGI, ISAPI – C, C++
  - Based on VB Script, COM, ADO
- ASP.NET 1.0 / 1.1 (2002-2005)
  - The First .NET based Web Development API
- ASP.NET 2.0 (2005-2007) – based on .NET 2.0
- ASP.NET 3.5 (2007-2009) – LINQ to SQL
- ASP.NET 4.0 (2010)
- ASP.NET 4.5 (2012)
.NET Framework (pronounced dot net) is a:

- software framework developed by Microsoft that runs primarily on Microsoft Windows.
- It includes a large class library known as Framework Class Library (FCL) and provides language interoperability across several programming languages.
- Programs written for .NET Framework execute in a software environment known as Common Language Runtime (CLR).
<table>
<thead>
<tr>
<th>Version number</th>
<th>CLR version</th>
<th>Release date</th>
<th>Development tool</th>
<th>Included in Windows</th>
<th>Included in Windows Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>2002-02-13</td>
<td>Visual Studio .NET[^22]</td>
<td>XP[^a]</td>
<td>N/A</td>
</tr>
<tr>
<td>3.0</td>
<td>2.0</td>
<td>2006-11-06</td>
<td>Expression Blend[^26][^b]</td>
<td>Vista</td>
<td>2008 SP2, 2008 R2 SP1</td>
</tr>
<tr>
<td>3.5</td>
<td>2.0</td>
<td>2007-11-19</td>
<td>Visual Studio 2008[^27]</td>
<td>7, 8[c], 8.1[c], 10[c]</td>
<td>2008 R2 SP1</td>
</tr>
<tr>
<td>4.0</td>
<td>4</td>
<td>2010-04-12</td>
<td>Visual Studio 2010[^28]</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>4.5.2</td>
<td>4</td>
<td>2014-05-05</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>4.6</td>
<td>4</td>
<td>2015-07-20</td>
<td>Visual Studio 2015[^31]</td>
<td>10</td>
<td>N/A</td>
</tr>
<tr>
<td>4.6.1</td>
<td>4</td>
<td>2015-11-30[^32]</td>
<td>Visual Studio 2015 Update 1</td>
<td>10 v1511</td>
<td>N/A</td>
</tr>
<tr>
<td>4.6.2</td>
<td>4</td>
<td>2016-08-02[^33]</td>
<td>Visual Studio 2015 Update 1</td>
<td>10 v1607</td>
<td>N/A</td>
</tr>
</tbody>
</table>
ASP.NET Execution

ASP.NET applications are executed via a sequence of HTTP requests and HTTP responses

- Client Web browser request ASPX pages
- The Web server executes the ASPX page and produce XHTML + CSS + JavaScript
## ASP.NET Architecture

<table>
<thead>
<tr>
<th>XML-based configuration</th>
<th>Html Controls</th>
<th>AJAX</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Web controls</td>
<td>User controls</td>
</tr>
<tr>
<td></td>
<td>HttpHandlers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASP.NET pages</td>
<td>ASP.NET Web services</td>
</tr>
<tr>
<td></td>
<td>HttpSession</td>
<td>Authentication</td>
</tr>
<tr>
<td></td>
<td>HttpApplication</td>
<td>Cache</td>
</tr>
</tbody>
</table>

- **ASP.NET runtime (aspnet_wp.dll / w3wp.dll)**
- **Internet Information Server (IIS)**
- **ISAPI Filters (aspnet_isapi.dll)**
- **Windows Server**
ASP.NET: How it Works?

- Traditional Web pages (static HTML)
  - Consist of static HTML, images, styles, etc.
  - Execute code on the client side
  - Simple operations

- ASP.NET Web Forms
  - Execute code on the server side
  - Database access
  - Dynamic pages
  - Higher security level
Separate Visualization from Business Logic

- Traditional Web development keep HTML and programming code in one file (PHP, ASP, …)
  - Hard to read, understand and maintain
  - Hard to test and debug
- ASP.NET splits the Web pages into two parts:
  - .aspx file containing HTML for visualization
  - "Code behind" files (.cs for C#) containing presentation logic for particular page
Separate Visualization from Business Logic
Separate Visualization from Business Logic (2)

- Class generated from the .aspx file does not derive directly from Page class.
- Derives from class defined in the "code behind", where it is easy to add methods, event handlers, etc.
- Using "code behind" separates the presentation logic from UI visualization.
ASP.NET Base Components

- Web Forms – deliver ASP.NET user interface
- Web Control – the smallest part we can use in our Web application (e.g. text box)
- "Code behind" – contains the server-side code
- Web.config – contains ASP.NET application configuration
- Global.asax – class containing application level event handlers
ASP.NET Web Controls

- ASP.NET Web controls are the smallest component part
- Deliver fast and easy component-oriented development process
- HTML abstraction, but finally everything is HTML

```html
<form runat="server" ID="frmMain">
  <asp:button runat="server" ID="btn" Text="Click me!" OnClick="btn_Click" />
</form>
```
Web.config

- Main settings and configuration file for ASP.NET
- Text based XML document
- Defines:
  - Connection strings to any DB used by app
  - The default language for child pages
  - Whether debugging is allowed

```xml
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
  </system.web>
</configuration>
```
Global.asax

- Also known as ASP.NET application file
- Located in the Web application root folder
- Exposes application and session level events
  - Application_Start
  - Application_End
  - Session_Start
  - Session_End
ASP.NET Execution Model

First call to particular page

Client

Server

Internet Explorer

Parser

Compiler

Assembly Cache

Memory

Assembly IL

Execute HTTP Runtime

Assembly IL
Any other call after the first