

asp net examples example source code organized by topic

Published: September 19, 2026 | Index ID: #-

Introduction

ASP.NET is a powerful, open source framework that enables developers to build dynamic, data driven web applications with ease. Whether you're creating a simple informational site, a complex enterprise solution, or a real time communication platform, ASP.NET offers a rich set of tools and libraries to accelerate development. One of the best ways to learn and master this framework is through hands on examples. By studying **asp net examples example source code organized by topic**, developers can quickly grasp concepts, experiment with new features, and integrate best practices into their projects.

This article provides an exhaustive, organized guide to ASP.NET examples. We'll walk through the most common application types—Web Forms, MVC, Web API, Razor Pages, SignalR, Blazor—and dive into topics such as authentication, data access, unit testing, and deployment. Each section includes concise code snippets, explanations, and links to full source files so you can copy, run, and modify them right away.

Why Structured Example Collections Matter

- Fast learning curve: Seeing a problem solved in a clean, context specific example saves hours of trial and error.
- Reusability: Organized examples can be dropped into new projects, saving time on boilerplate code.
- Best practices: Well structured source code demonstrates modern patterns—dependency injection, async/await, separation of concerns.
- Community validation: Examples that are widely used and reviewed by the community tend to be more robust.
- SEO and discoverability: Clear topic headings and keyword rich descriptions help search engines index useful content.

Getting Started: Setting Up Your Development Environment

Before diving into examples, make sure you have the following installed:

1. Visual Studio 2022 (or newer) with the ASP.NET and web development workload.
2. .NET 8 SDK (the latest LTS release).
3. Git for version control.
4. SQL Server Express or LocalDB for data access examples.

Once you have the environment ready, you can clone the sample repository from GitHub:

```
git clone https://github.com/yourusername/aspnet-examples.git
cd aspnet-examples
dotnet restore
```

The repository is structured by topic, with each folder containing a self contained project and a README that explains the purpose and usage.

ASP.NET Core MVC Examples

1. Classic CRUD Application

Our first example demonstrates a classic Create Read Update Delete flow using Entity Framework Core and a SQLite database. The project is called MvcCrudSample and includes:

- Models: Product.cs with data annotations.
- Controllers: ProductsController.cs with async actions.
- Views: Razor pages for listing, editing, and deleting.
- Data context: AppDbContext.cs with migrations.

```
// ProductsController.cs
[Route("[controller]")]
public class ProductsController : Controller
{
    private readonly AppDbContext _context;
    public ProductsController(AppDbContext context) => _context = context;

    [HttpGet]
    public async Task Index() =>
        View(await _context.Products.ToListAsync());

    [HttpGet("{id}")]
    public async Task Details(int id) =>
        View(await _context.Products.FindAsync(id));

    [HttpPost]
    public async Task Create([Bind("Name,Price")] Product product)
    {
        if (ModelState.IsValid)
        {
            _context.Add(product);
            await _context.SaveChangesAsync();
            return RedirectToAction(nameof(Index));
        }
        return View(product);
    }
}
```

Run the application with `dotnet run` and navigate to `/products` to see the CRUD UI in action.

2. Advanced Routing & Attribute Routing

ASP.NET Core MVC allows you to fine tune URLs with attribute routing. The MvcRoutingSample project showcases custom route templates, constraints, and fallback routes. Here's a snippet:

```
[Route("products")]
public class ProductController : Controller
{

```

```

[Route("list")]
public IActionResult List() => View();

[Route("{id:int:min(1)}")]
public IActionResult Details(int id) => View();
}

```

Notice the `int:min(1)` constraint that ensures only positive integers match the route.

3. API Integration with Swagger

In the `MvcApiSample` project, we expose the same CRUD operations as a RESTful API and automatically generate interactive documentation using Swashbuckle. The `Startup.cs` configuration includes:

```

services.AddSwaggerGen(c =>
{
    c.SwaggerDoc("v1", new OpenApiInfo
    {
        Title = "Product API",
        Version = "v1",
        Description = "A simple ASP.NET Core Web API"
    });
});

```

Run the app and visit `/swagger` to interact with the API.

ASP.NET Core Razor Pages Examples

1. Simple Razor Page

The `RazorSimpleSample` project demonstrates how Razor Pages simplify page centric development. A single `Index.cshtml` file and its code behind `Index.cshtml.cs` handle all logic:

```

// Index.cshtml.cs
public class IndexModel : PageModel
{
    public string Message { get; set; }
    public void OnGet() => Message = "Hello, Razor Pages!";
}

```

Run the app and navigate to `/` to see the greeting.

2. Form Handling with Validation

The `RazorFormSample` project includes a contact form with client side validation. The `Contact.cshtml` page uses `<form>` tags and the `ContactModel` class for data binding:

```

public class ContactModel : PageModel
{
    [BindProperty]

```

```

[Required]
public string Email { get; set; }

[BindProperty]
[StringLength(500)]
public string Message { get; set; }

public void OnPost()
{
    if (ModelState.IsValid)
    {
        // Process the message
    }
}
}

```

Client side validation is automatically rendered by Razor, thanks to unobtrusive validation libraries.

ASP.NET Core Web API Examples

1. Minimal APIs

ASP.NET Core 6 introduced Minimal APIs, a lightweight approach to building HTTP services. The MinimalApiSample project includes a simple product API with minimal code:

```

var builder = WebApplication.CreateBuilder(args);
var app = builder.Build();

app.MapGet("/products", async () =>
{
    // Return a list of products
});

app.MapPost("/products", async (Product product) =>
{
    // Add a new product
});

app.Run();

```

Run with `dotnet run` and test the endpoints using tools like Postman or cURL.

2. Authentication & Authorization

The WebApiAuthSample project demonstrates JWT bearer authentication. The Startup.cs includes:

```

services.AddAuthentication(JwtBearerDefaults.AuthenticationScheme)
    .AddJwtBearer(options =>

```

```

{
    options.TokenValidationParameters = new TokenValidationParameters
    {
        ValidateIssuer = true,
        ValidateAudience = true,
        ValidateLifetime = true,
        ValidateIssuerSigningKey = true,
        ValidIssuer = "https://localhost",
        ValidAudience = "https://localhost",
        IssuerSigningKey = new SymmetricSecurityKey(Encoding.UTF8.GetBytes("SuperSecretKey"))
    };
});

```

All API routes are protected by the [Authorize] attribute, ensuring only authenticated users can access them.

ASP.NET SignalR Examples

1. Real Time Chat Application

The SignalRChatSample project implements a simple chat hub using SignalR. The ChatHub.cs class exposes methods that clients can invoke:

```

public class ChatHub : Hub
{
    public async Task SendMessage(string user, string message)
    {
        await Clients.All.SendAsync("ReceiveMessage", user, message);
    }
}

```

The client side uses JavaScript to connect and listen for ReceiveMessage events.

2. Group Management

In the SignalRGroupsSample project, we extend the chat example to support user groups. Clients can join or leave groups, and messages are broadcast only to group members.

```

public async Task JoinGroup(string groupName)
{
    await Groups.AddToGroupAsync(Context.ConnectionId, groupName);
}

public async Task LeaveGroup(string groupName)
{
    await Groups.RemoveFromGroupAsync(Context.ConnectionId, groupName);
}

```

This pattern is useful for implementing features like live dashboards or multiplayer games.

ASP.NET Blazor Examples

1. Server Side Blazor

The BlazorServerSample project showcases a server side Blazor app with a counter component. The Counter.razor file contains:

```
@page "/counter"
```

```
Current count: @currentCount
```

Server side Blazor runs on the server and uses SignalR to keep the UI in sync.

2. WebAssembly Blazor

The BlazorWasmSample project demonstrates a client side Blazor WebAssembly app that consumes a REST API. It includes:

- API client service using HttpClient.
- State management with ProtectedLocalStorage.
- Lazy loading of components.

Run with `dotnet run` and navigate to / to see the SPA in action.

ASP.NET Core Data Access Examples

1. Entity Framework Core with Code First Migrations

The EfCoreSample project illustrates how to set up EF Core, define entities, and apply migrations. The OnModelCreating method configures relationships:

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.Entity()
        .HasMany(o => o.Items)
        .WithOne()
        .HasForeignKey(i => i.OrderId);
}
```

Run `dotnet ef migrations add InitialCreate` followed by `dotnet ef database update` to create the database.

2. Dapper Micro ORM

For high performance scenarios, the DapperSample project shows how to use Dapper for lightweight data access. The ProductRepository.cs class contains:

```
public async Task< GetAsync()
{
    using var connection = new SqlConnection(_connectionString);
    return await connection.QueryAsync("SELECT * FROM Products");
}
```

Dapper maps query results to POCOs with minimal overhead.

ASP.NET Core Security Examples

Baca Juga (Artikel Terkait)

- [applied mechanics for engineering technology keith m walker](http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-keith-m-walker.pdf)

URL: <http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-keith-m-walker.pdf>

- [applied mechanics for engineering technology answers](http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-answers.pdf)

URL: <http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-answers.pdf>

- [applied mechanics for engineering technology 8th edition solution manual](http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-8th-edition-solution-manual.pdf)

URL: <http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-8th-edition-solution-manual.pdf>

- [applied mechanics for engineering technology 8th edition](http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-8th-edition.pdf)

URL: <http://www.atemplatefree.com/applied-mechanics-for-engineering-technology-8th-edition.pdf>

Tags: [#examples](#) [#example](#) [#source](#) [#code](#) [#organized](#) [#topic](#)

