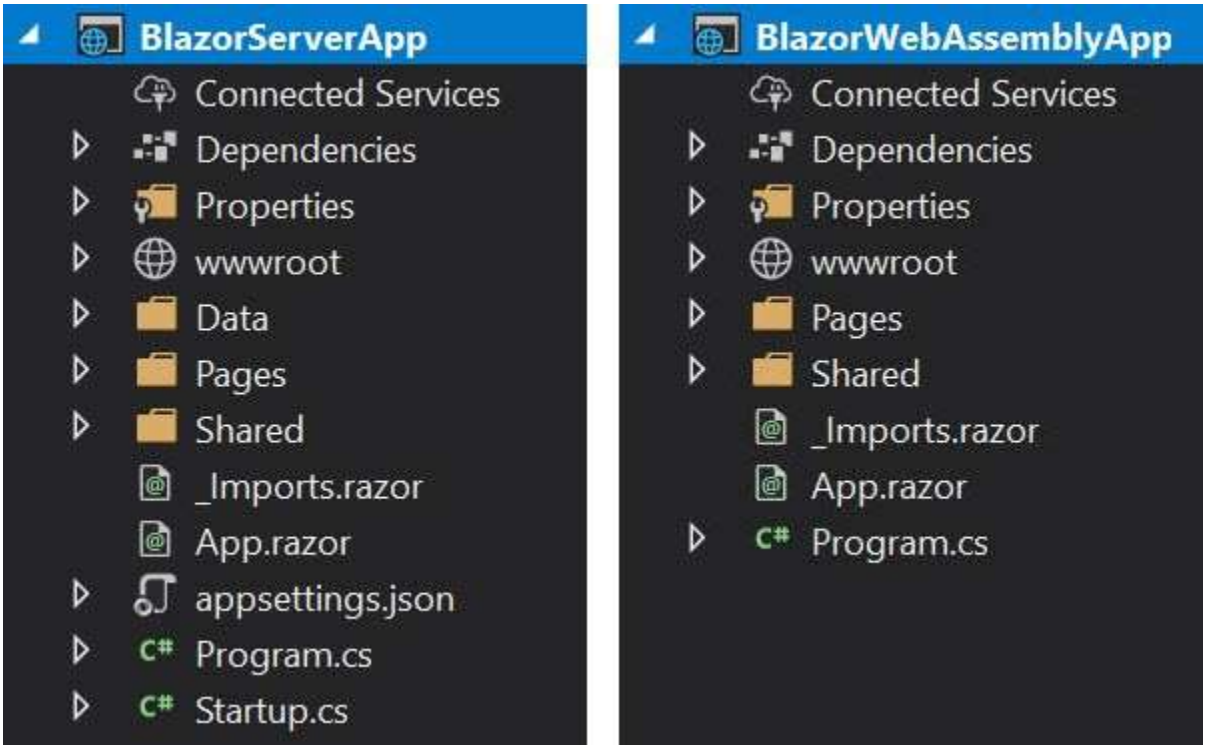
**Blazor project structure in the new unified model introduced in .NET 8 and continued in .NET 9 / .NET 10 Blazor Web App structure** used for production applications.



### 1. Default Project Structure (Unified Model)

```
BlazorWebApp/
│
├── Components/
│   ├── App.razor
│   ├── Routes.razor
│
├── Pages/
│   ├── Home.razor
│   ├── Counter.razor
│   └── Weather.razor
│
├── Layouts/
│   └── MainLayout.razor
│
├── wwwroot/
│   ├── css/
│   ├── js/
│   └── images/
│
├── Properties/
│   └── launchSettings.json
│
├── Program.cs
├── _Imports.razor
└── BlazorWebApp.csproj
```

---

### 3. Core Folders Explained

**Components/**

- Contains **core application components**
- Includes app bootstrapping files

Typical files:

- App.razor – Root component
- Routes.razor – Central routing configuration

Purpose:

- App initialization
- Routing pipeline
- Render mode boundaries

---

**Pages/**

- Contains **routable UI components**
- Each page uses @page

Example:

@page "/products"

@rendermode InteractiveAuto

Purpose:

- Represents screens/pages
- Entry points for navigation

---

**Layouts/**

- Defines **shared UI layout**
- Controls structure across pages
- Uses @Body placeholder

---

**Wwwroot/**

- **Contains static assets**
    - **CSS files**
    - **JavaScript files**
    - **Images**
- **Equivalent to wwwroot in ASP.NET Core MVC.**
- **In WebAssembly, everything here is downloaded to the browser.**

---

**4. Key Root Files**

**Program.cs**

Central startup configuration:

- Service registration
- Render mode configuration
- Middleware pipeline
- Configures:
    - Services (DI)
    - Authentication / Authorization
    - HttpClient
    - Blazor hosting model
- Example responsibilities:
- builder.Services.AddRazorComponents()
- AddServerSideBlazor() or AddWebAssembly()

---

**App.razor**

Root component responsible for:

- Application routing
- Layout assignment

Now often delegates routing to Routes.razor.

---

**Routes.razor (New in .NET 8+)**

- Centralizes routing logic
- Separates routing from app initialization

Purpose:

- Cleaner architecture
- Easier route management

---

**_Imports.razor**

Global Razor imports:

- Common namespaces
- Shared components

---

**.csproj**

Defines:

- Target framework (net8.0, net9.0, net10.0)
- Blazor features
- Package references

---

**5. Render Modes (Key Concept in .NET 8/10)**

Blazor Web App supports **per-component render modes**:

| **Render Mode** | **Behavior** |
|---|---|
| InteractiveServer | Server + SignalR |
| InteractiveWebAssembly | Runs in browser |
| InteractiveAuto | Server first, WASM later |

Example:

@rendermode InteractiveWebAssembly

This is **the biggest architectural change** in modern Blazor.

---

**6. Recommended Enterprise Additions**

├── Services/ → Business logic & API calls

├── Models/ → DTOs / ViewModels

├── State/ → App state containers

├── Auth/ → Authentication / Authorization

├── Infrastructure/ → External integrations

---

**7. Execution Flow (Simplified)**

1. App starts in **Program.cs**
2. Root component (App.razor) loads
3. Routing handled by Routes.razor
4. Layout applied
5. Page rendered with chosen render mode
6. Components become interactive

---