

Pams.WebApp

August 18, 2026

Pams.WebApp

Kind: Module

Source: Pams/Web/Pams.WebApp/Pams.WebApp.csproj (line 1)

Part of: [Pams](#)

.NET Library project (C#)

Pams.WebApp is a C# .NET project in the Pams web layer. It contains web-facing application code and connects browser or HTTP callers to Pams services. TypeScript clients interact with the hosted application through its configured routes.

Diagram

```
graph TD
    Client[Client[Browser or TypeScript Client]] --> WebApp[WebApp[Pams.WebApp]]
    WebApp --> Services[Services[Pams Services]]
    WebApp --> Dependencies[Dependencies[Configured Dependencies]]
```

Usage

```
const webAppUrl = import.meta.env.VITE_PAMS_WEBAPP_URL;

async function requestPamsWebApp() {
  const response = await fetch(webAppUrl, {
    headers: {
      Accept: "application/json",
    },
  });

  if (!response.ok) {
    throw new Error(`Pams.WebApp request failed: ${response.status}`);
  }

  return response.json();
}
```

```
}  
  
const result = await requestPamsWebApp();  
console.log(result);
```

AI Coding Instructions

- Keep web-facing logic in `Pams.WebApp` and move domain behavior into the related Pams service layer.
- Follow existing project references and dependency-registration patterns in the solution.
- Do not assume route paths or response formats; inspect controllers, minimal API mappings, and configuration before adding client calls.
- Keep browser clients dependent on hosted HTTP routes rather than direct C# project references.

Relationships

- DEPENDS_ON → `Microsoft.AspNetCore.All`
- DEPENDS_ON → `Microsoft.VisualStudio.Web.CodeGeneration.Design`
- DEPENDS_ON → `NETStandard.Library`