Owin

August 18, 2026

Owin

Kind: Service

Source: `Pams/Logic/Pams.Business/Pams.Business.csproj` (line 1)

Part of: Pams

NuGet package dependency

Owin is a NuGet package dependency declared by `Pams.Business.csproj` in the Pams business layer. It supports OWIN-based middleware and HTTP pipeline integration when the application is hosted with OWIN-compatible components.

Diagram

```
sequenceDiagram
    participant Host as Application Host
    participant Startup as OWIN Startup
    participant Owin as Owin Package
    participant Business as Pams.Business

    Host->>Startup: Start application
    Startup->>Owin: Configure middleware pipeline
    Owin->>Business: Forward request handling
    Business-->>Owin: Return response
    Owin-->>Host: Send HTTP response
```

Usage

```
type PamsClientOptions = {
  baseUrl: string;
};

export async function callPams(
  options: PamsClientOptions,
  path: string,
```

```
  init: RequestInit = {},
) {
  const response = await fetch(`${options.baseUrl}${path}`, {
    ...init,
    headers: {
      Accept: "application/json",
      ...init.headers,
    },
  });

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

  return response.json();
}

// Calls an endpoint handled by the OWIN-hosted Pams application.
const result = await callPams(
  { baseUrl: process.env.PAMS_BASE_URL ?? "" },
  "/api/example",
);
```

AI Coding Instructions

- Keep the Owin package reference in `Pams/Logic/Pams.Business/Pams.Business.csproj` aligned with the project's legacy package format.
- Configure OWIN middleware during application startup rather than from business-domain classes.
- Ensure middleware forwards requests to the Pams business layer and returns responses through the OWIN pipeline.
- Do not import the .NET Owin package from TypeScript or JavaScript; JavaScript clients should call the hosted HTTP application.

Used by

4 references from 4 files. Each is a place in this repository where the symbol is actually used — go read one rather than trusting an example.

Injected or called by (4)

- `Pams.API` — `Pams/API/Pams.API/Pams.API.csproj`:1
- `Pams.Core` — `Pams/Core/Pams.Core/Pams.Core.csproj`:1
- `Pams.Security` — `Pams/Core/Pams.Security/Pams.Security.csproj`:1

- `Pams.Business` — `Pams/Logic/Pams.Business/Pams.Business.csproj`:1