

System.Security

August 17, 2026

System.Security

Kind: Service

Source: Pams/Data/Pams.Data/Pams.Data.csproj (line 1)

Part of: Pams

NuGet package dependency

`System.Security` is a NuGet package dependency declared by the `Pams.Data` project. It provides .NET security-related types that may be required by data-layer code and is resolved during the project restore and build process.

Diagram

```
sequenceDiagram
    participant Build as Build Process
    participant Project as Pams.Data.csproj
    participant NuGet as NuGet Restore
    participant Package as System.Security

    Build->>Project: Read package references
    Project->>NuGet: Request System.Security
    NuGet->>Package: Resolve package dependency
    Package-->>NuGet: Provide package assets
    NuGet-->>Build: Restore dependency for Pams.Data
```

Usage

```
type SecurityDependencyStatus = {
    packageName: string;
    restored: boolean;
};

function verifySecurityDependency(
    dependency: SecurityDependencyStatus,
```

```

): void {
    if (dependency.packageName != "System.Security") {
        throw new Error("Unexpected security package dependency.");
    }

    if (!dependency.restored) {
        throw new Error("Restore NuGet packages before building Pams.Data.");
    }
}

verifySecurityDependency({
    packageName: "System.Security",
    restored: true,
});

```

AI Coding Instructions

- Treat `System.Security` as a project-level NuGet dependency, not a TypeScript runtime package.
- Keep package reference changes in `Pams.Data.csproj` so restore and build tooling can resolve the dependency.
- Verify package restore succeeds before diagnosing missing security-related .NET types.
- Avoid adding client-side code that assumes `System.Security` can be imported from JavaScript or TypeScript.

Used by

1 reference from 1 file. 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 (1)

- `Pams.Data` — `Pams/Data/Pams.Data/Pams.Data.csproj` :1