

Microsoft.Extensions.Options.ConfigurationExtensions

August 18, 2026

Microsoft.Extensions.Options.ConfigurationExtensions

Kind: Service

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

Part of: [Pams](#)

NuGet package dependency

`Microsoft.Extensions.Options.ConfigurationExtensions` is a NuGet dependency in `Pams.Web.csproj` that binds application configuration values to .NET options classes. The web application uses it to read configuration providers such as `appsettings.json` and expose typed settings through dependency injection.

Diagram

```
sequenceDiagram
    participant Client as TypeScript Client
    participant Api as PAMS Web API
    participant Config as IConfiguration
    participant Options as Typed Options

    Client->>Api: Request endpoint
    Api->>Options: Read injected settings
    Options->>Config: Bind configuration section
    Config-->>Options: Configuration values
    Options-->>Api: Typed settings
    Api-->>Client: Response
```

Usage

```
type ApplicationSettings = {
    apiBaseUrl: string;
};

async function loadApplicationSettings(): Promise<ApplicationSettings> {
    const response = await fetch("/api/settings");

    if (!response.ok) {
        throw new Error("Unable to load application settings.");
    }
}
```

```
    return response.json();
  }

  const settings = await loadApplicationSettings();

  const usersResponse = await fetch(`${settings.apiUrl}/users`);
  const users = await usersResponse.json();

  console.log(users);
```

AI Coding Instructions

- Keep configuration binding in the ASP.NET Core application; TypeScript clients should consume settings through API endpoints when required.
- Add configuration keys to the relevant `appsettings.json` file and bind them to a typed options class.
- Register options binding during application startup before injecting the options type into services or controllers.
- Avoid placing secrets in client-side configuration responses or source-controlled settings files.

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.Web` — `Pams/Web/Pams.Web/Pams.Web.csproj` :1