

Setup

This guide explains how to use the `StremioDotNet` package to build and configure Stremio addons.

Example Code

Below is an example of how to set up a `StremioDotNet` addon in an ASP.NET Core application:

```
using StremioDotNet.Builders;
using StremioDotNet.Extensions;
using StremioDotNet.Structs.Manifest;

var builder = WebApplication.CreateBuilder(args);

builder.Services.AddEndpointsApiExplorer();
builder.Services.AddSwaggerGen();
builder.Services.AddControllers();
builder.Services.AddStremio(); // Add Stremio

var app = builder.Build();

if (app.Environment.IsDevelopment())
{
    app.UseSwagger();
    app.UseSwaggerUI();
}

app.UseRouting(); // Required, order matters!
app.UseHttpsRedirection();

AddonBuilder ConfigureAddon() => new AddonBuilder(
    "com.example",
    "1.0.0",
    "Example Addon",
    "Example Description",
```

```

[AddonBuilder.Resources.Stream, AddonBuilder.Resources.Meta, AddonBuilder.Resources.Catalog],
["movie", "series"],
["tt", "ee"]
).SetCatalogs([
    new Catalog
    {
        Type = "movie",
        Id = "stremioDotNet",
        Name = "StremioDotNet Movies"
    }
]); // Configure Stremio

if (app.Environment.IsProduction())
{
    app.UseStremioAddon(() => ConfigureAddon().PublishToCentral("https://stremio.nathan.rip")); // Submit to
Stremio's repository in production
}
else
{
    app.UseStremioAddon(ConfigureAddon); // Local development
}

app.MapControllers();
app.Run();

```

Key Components

1. **AddStremio** Method

Definition:

```
public static void AddStremio(this IServiceCollection services)
```

- **Namespace:** `StremioDotNet.Extensions`
- **Purpose:** Registers the required services for a Stremio addon, including:
 - CORS configuration.
 - An HTTP client.

- IMDb ID resolution middleware.

Usage:

```
builder.Services.AddStremio();
```

Parameters:

- `services`: The `IServiceCollection` to which the services are added.

2. AddonBuilder Class

Definition:

```
public class AddonBuilder
```

- **Namespace:** `StremioDotNet.Builders`

Purpose: The `AddonBuilder` class simplifies the creation and configuration of a Stremio Addon Manifest by providing:

- Methods to set behavior hints.
- Catalog definitions.
- Other necessary configurations.

Example:

```
AddonBuilder ConfigureAddon() => new AddonBuilder(
    "com.example",
    "1.0.0",
    "Example Addon",
    "Example Description",
    [AddonBuilder.Resources.Stream, AddonBuilder.Resources.Meta, AddonBuilder.Resources.Catalog],
    ["movie", "series"],
    ["tt", "ee"]
).SetCatalogs([
    new Catalog
    {
        Type = "movie",
        Id = "stremioDotNet",
        Name = "StremioDotNet Movies"
```

```
}  
});
```

3. UseStremioAddon Method

Definition:

```
public static void UseStremioAddon(  
    this IApplicationBuilder app,  
    Func<AddonBuilder> createAddonBuilder)
```

- **Namespace:** `StremioDotNet.Extensions`
- **Purpose:** Configures the application to use a Stremio addon by:
 - Setting up middleware.
 - Mapping a GET endpoint (`/manifest.json`) to serve the addon manifest.

Usage:

```
app.UseStremioAddon(() => ConfigureAddon().PublishToCentral("https://stremio.nathan.rip"));
```

Parameters:

- `app`: The `IApplicationBuilder` instance used to configure the HTTP request pipeline.
- `createAddonBuilder`: A factory function that creates an `AddonBuilder` instance for manifest configuration.

Exceptions:

- `ArgumentNullException`: Thrown if `createAddonBuilder` is null.

Note: `PublishToCentral` makes a request to Stremio's repository of addons, allowing your addon to appear on the community addons page.

Summary

By using the `AddStremio`, `AddonBuilder`, and `UseStremioAddon` methods, you can quickly set up a fully functional Stremio addon within your ASP.NET Core application. For production environments, use `PublishToCentral` to submit your addon to Stremio's repository and make it visible in the community addons page.

Revision #3

Created 25 December 2024 23:59:16 by Nathan

Updated 26 December 2024 00:34:54 by Nathan