

RSCG nr 16 : PartiallyApplied

Info

Nuget : <https://www.nuget.org/packages/PartiallyApplied/>

You can find more details at : <https://github.com/JasonBock/PartiallyApplied/blob/main/docs/Quickstart.md>

Author : Jason Bock

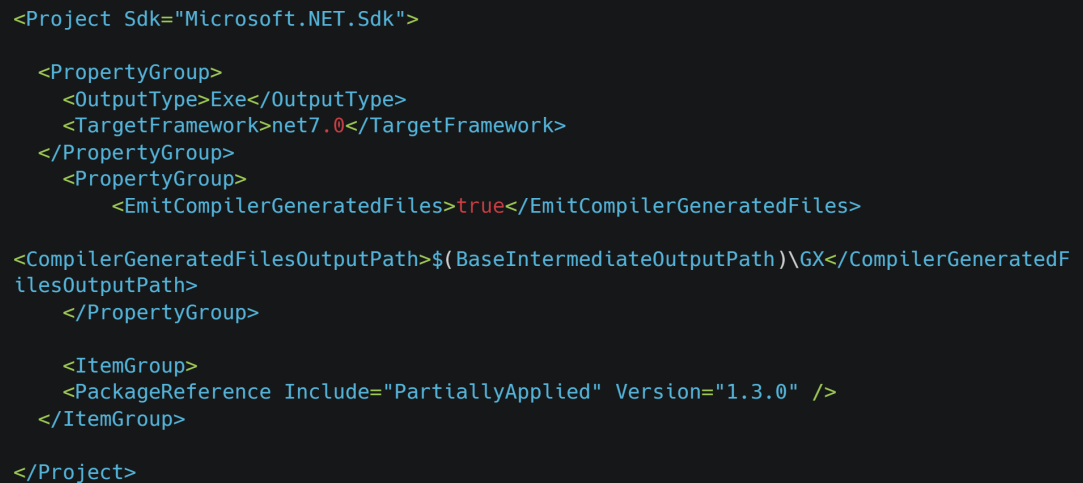
Source : <https://github.com/JasonBock/PartiallyApplied>

About

If you need to curry functions, you can use this package

How to use

Add reference to the PartiallyApplied in the csproj

A screenshot of a code editor with a dark background and light-colored text. The code is XML for a .csproj file. It includes a Project element with Sdk="Microsoft.NET.Sdk". Inside, there are two PropertyGroup elements. The first sets OutputType to Exe and TargetFramework to net7.0. The second sets EmitCompilerGeneratedFiles to true. Below these is a CompilerGeneratedFilesOutputPath element with a path ending in \GX. Finally, there is an ItemGroup element containing a PackageReference for PartiallyApplied version 1.3.0.

```
<Project Sdk="Microsoft.NET.Sdk">

  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net7.0</TargetFramework>
  </PropertyGroup>

  <PropertyGroup>
    <EmitCompilerGeneratedFiles>true</EmitCompilerGeneratedFiles>


    <CompilerGeneratedFilesOutputPath>$(BaseIntermediateOutputPath)\GX</CompilerGeneratedFilesOutputPath>
  </PropertyGroup>

  <ItemGroup>
    <PackageReference Include="PartiallyApplied" Version="1.3.0" />
  </ItemGroup>

</Project>
```

This was for me the starting code

I have coded the file Program.cs



```
•using System;

namespace PartFunc;
class Program
{
    static void Main(string[] args)
    {
        var disc10Percent = Partially.Apply(Accounting.Discount, 1/10f);
        Console.WriteLine(disc10Percent(disc10Percent(100)));
    }
}
```

I have **coded** the file Accounting.cs




```
•namespace PartFunc;

public class Accounting
{
    public static float Discount( float discount, float price)
    {
        var val= price * (1- discount);
        return val;
    }
}
```

And here are the *generated* files

The file *generated* is Partially.g.cs



```
using System;

#nullable enable
public static partial class Partially
{
    public static Func<float, float> Apply(Func<float, float, float> method, float
discount) =>
        new((price) => method(discount, price));
}
```

You can download the code and this page as pdf from https://ignatandrei.github.io/RSCG_Examples/v2/docs/PartiallyApplied

You can see the whole list at https://ignatandrei.github.io/RSCG_Examples/v2/docs/List-of-RSCG